

Guía del usuario

AWS CodeDeploy



Versión de API 2014-10-06

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CodeDeploy: Guía del usuario

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 CodeDeploy?	1
Ventajas de AWS CodeDeploy	2
Descripción general de las plataformas informáticas CodeDeploy	3
Descripción general de los tipos de implementación CodeDeploy	11
Información general de la implementación local	13
Información general de la implementación azul/verde	14
Esperamos tener noticias tuyas	19
Componentes principales	19
Aplicación	19
Plataforma de informática	20
Configuración de implementación	21
Grupo de implementación	21
Tipo de implementación	22
Tipo de implementación	23
Revisión	23
Rol de servicio	24
Revisión de destino	24
Otros componentes	24
Implementaciones	25
Implementaciones en una plataforma de informática de AWS Lambda	25
Implementación en una plataforma de informática de Amazon ECS	29
Implementación azul/verde en una plataforma de informática de EC2/en las instalaciones	42
Archivos de especificación de la aplicación	49
AppSpec archivos en una plataforma informática Amazon ECS	50
AppSpec archivos en una plataforma AWS Lambda informática	50
AppSpec archivos en una plataforma informática local EC2/	50
Cómo usa el CodeDeploy agente el AppSpec archivo	51
Introducción	52
Paso 1: Configurar	52
Inscríbese en un Cuenta de AWS	52
Creación de un usuario con acceso administrativo	53
Concesión de acceso programático	54
Paso 2: Crear un rol de servicio	56
Creación de un rol de servicio (consola)	58

Creación de un rol de servicio (CLI)	61
Obtención del ARN del rol de servicio (consola)	64
Obtención del ARN del rol de servicio (CLI)	64
Paso 3: Limitar los permisos del CodeDeploy usuario	65
Paso 4: Crear un perfil de instancia de IAM	69
Creación de un perfil de instancia de IAM para las instancias de Amazon EC2 (CLI)	69
Creación de un perfil de instancia de IAM para las instancias de Amazon EC2 (consola)	73
Integraciones de productos y servicios	77
Integración con otros AWS servicios	77
Amazon EC2 Auto Scaling	85
Elastic Load Balancing	94
Integración con productos y servicios de socios	99
GitHub	105
Ejemplos de integración de la comunidad	109
Publicaciones de blog	110
Tutoriales	111
Tutorial: Implemente en una instancia WordPress que no sea de Windows	111
Paso 1: Lanzar una instancia de Amazon EC2	112
Paso 2: Configurar el contenido de origen	115
Paso 3: Cargar la aplicación en Amazon S3	120
Paso 4: Implementar la aplicación	125
Paso 5: Actualizar y volver a implementar la aplicación	132
Paso 6: limpiar	136
Tutorial: Implementación de una aplicación Hello World en una instancia de Windows Server ..	139
Paso 1: Lanzar una instancia de Amazon EC2	140
Paso 2: Configurar el contenido de origen	143
Paso 3: Cargar la aplicación en Amazon S3	146
Paso 4: Implementar la aplicación	151
Paso 5: Actualizar y volver a implementar la aplicación	156
Paso 6: limpiar	159
Tutorial: Implementación de una aplicación en una instancia en las instalaciones	163
Requisitos previos	164
Paso 1: Configurar la instancia en las instalaciones	164
Paso 2: Crear una revisión de aplicación de ejemplo	164
Paso 3: Empaquetado y carga de la revisión de aplicación en Amazon S3	170
Paso 4: Implementar la revisión de la aplicación	170

Paso 5: Comprobar la implementación	170
Paso 6: Limpiar los recursos	171
Tutorial: Implementación en un grupo de escalado automático	173
Requisitos previos	173
Paso 1: Crear y configurar el grupo de escalado automático	174
Paso 2: Implementar la aplicación en el grupo de escalado automático	181
Paso 3: Comprobar los resultados	191
Paso 4: Aumentar el número de instancias de Amazon EC2 en el grupo de escalado automático	193
Paso 5: Comprobar los resultados de nuevo	194
Paso 6: limpiar	197
Tutorial: Implemente una aplicación desde GitHub	199
Requisitos previos	200
Paso 1: Configurar una GitHub cuenta	200
Paso 2: Crea un GitHub repositorio	200
Paso 3: Sube una aplicación de muestra a tu repositorio GitHub	203
Paso 4: Aprovisionar una instancia	208
Paso 5: Crear una aplicación y un grupo de implementación	208
Paso 6: Implementar la aplicación en la instancia	211
Paso 7: Monitorizar y comprobar la implementación	215
Paso 8: Eliminación	216
Tutorial: Implementación de una aplicación en Amazon ECS	218
Requisitos previos	220
Paso 2: Actualizar la aplicación de Amazon ECS	221
Paso 2: Crea el AppSpec archivo	222
Paso 3: Utilice la CodeDeploy consola para implementar la aplicación	224
Paso 4: Limpiar	228
Tutorial: Implementación de un servicio de Amazon ECS con una prueba de validación	229
Requisitos previos	232
Paso 1: Creación de un oyente de prueba	232
Paso 2: Actualizar la aplicación de Amazon ECS	233
Paso 3: Creación de una función de Lambda de enlace de ciclo de vida	233
Paso 4: actualice AppSpec el archivo	236
Paso 5: Utilice la CodeDeploy consola para implementar el servicio Amazon ECS	237
Paso 6: Ver el resultado de la función de enlace Lambda en los registros CloudWatch	239
Paso 7: limpiar	240

Tutorial: Implementación de una función Lambda mediante SAM AWS	241
Requisitos previos	242
Paso 1: Configurar la infraestructura	242
Paso 2: Actualizar la función de Lambda	257
Paso 3: Implementar la función de Lambda actualizada	260
Paso 4: Visualizar los resultados de la implementación	262
Paso 5: Eliminar	265
Trabajando con el CodeDeploy agente	267
Sistemas operativos compatibles con el agente CodeDeploy	268
Sistemas operativos de AMI de Amazon EC2 compatibles	268
Sistemas operativos en las instalaciones compatibles	268
Protocolo de comunicación y puerto para el CodeDeploy agente	268
Historial de versiones del agente CodeDeploy	269
Administrar el CodeDeploy proceso	285
Limpieza de revisiones de aplicación y archivos de registro	285
Archivos instalados por el agente CodeDeploy	286
Gestión de las operaciones CodeDeploy de los agentes	289
Compruebe que el CodeDeploy agente se esté ejecutando	290
Determine la versión del CodeDeploy agente	292
Instala el CodeDeploy agente	293
Actualice el CodeDeploy agente	305
Desinstale el CodeDeploy agente	309
Enviar los registros CodeDeploy de los agentes a CloudWatch	310
Trabajar con instancias de	315
Comparación de las instancias de Amazon EC2 y las instancias en las instalaciones	315
Tareas de instancia para CodeDeploy	317
Etiquetado de instancias para despliegues CodeDeploy	319
Ejemplo 1: un único grupo de etiquetas, una única etiqueta	320
Ejemplo 2: un único grupo de etiquetas, varias etiquetas	321
Ejemplo 3: varios grupos de etiquetas, etiquetas únicas	323
Ejemplo 4: varios grupos de etiquetas, varias etiquetas	325
Uso de instancias de Amazon EC2	329
Cree una instancia de Amazon EC2 para CodeDeploy	329
Crear una instancia de Amazon EC2 (plantilla)AWS CloudFormation	336
Configuración de una instancia de Amazon EC2	347
Trabajo con instancias en las instalaciones	351

Requisitos previos para configurar una instancia en las instalaciones	352
Registro de una instancia en las instalaciones	354
Cómo administrar operaciones de instancias en las instalaciones	390
Visualización de los detalles de la instancia	397
Visualización de los detalles de la instancia (consola)	398
Visualización de los detalles de la instancia (CLI)	399
Estado de la instancia	399
Estado	400
Acerca del número mínimo de instancias en buen estado	401
Acerca del número mínimo de instancias en buen estado por zona de disponibilidad	406
Uso de configuraciones de implementación	409
Configuraciones de implementación en una plataforma de informática EC2/en las instalaciones	409
Configuraciones de implementación predefinidas	409
Configuraciones de implementación en una plataforma de informática de Amazon ECS	415
Configuraciones de implementación predefinidas de Amazon ECS	415
Configuraciones de implementación para implementaciones azul/verde de AWS CloudFormation (Amazon ECS)	416
Configuraciones de implementación en una plataforma de informática de AWS Lambda	417
Configuraciones de implementación predefinidas para Lambda	417
.....	418
Creación de una configuración de implementación	418
Crear una configuración de implementación (consola)	419
Crear una configuración de implementación (AWS CLI)	421
Consulta de los detalles de las configuraciones de implementación	423
Consulta de los detalles de las configuraciones de implementación (consola)	423
Consulta de la configuración de implementación (CLI)	424
Eliminación de una configuración de implementación	424
Trabajo con aplicaciones	425
Crear una aplicación	426
Creación de una aplicación para una implementación local (consola)	428
Creación de una aplicación para una implementación azul/verde (consola)	432
Creación de una aplicación para la implementación de un servicio de Amazon ECS (consola)	437
Creación de una aplicación para la implementación de una función de AWS Lambda (consola)	439

Creación de una aplicación (CLI)	441
Visualización de los detalles de la aplicación	442
Visualización de los detalles de la aplicación (consola)	442
Visualización de los detalles de la aplicación (CLI)	442
Creación de una regla de notificación	443
Cambiar el nombre de una aplicación	446
Eliminar una aplicación	446
Eliminar una aplicación (consola)	447
Eliminar una aplicación (AWS CLI)	447
Trabajo con grupos de implementación	448
Grupos de implementación en las implementaciones en la plataforma de informática Amazon ECS	448
Grupos de despliegue en despliegues de plataformas AWS Lambda informáticas	448
Grupos de implementación en las implementaciones en la plataforma de informática EC2/en las instalaciones	448
.....	449
Creación de un grupo de implementación	450
Creación de un grupo de implementación para una implementación local (consola)	450
Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones (consola)	455
Creación de un grupo de implementación para una implementación de Amazon ECS (consola)	460
Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2	462
Configure un balanceador de carga, grupos objetivo y agentes de escucha para las implementaciones de CodeDeploy Amazon ECS	464
Creación de un grupo de implementación (CLI)	469
Visualización de los detalles de un grupo de implementación	470
Visualización de los detalles de un grupo de implementación (consola)	471
Visualización de los detalles de un grupo de implementación (CLI)	471
Cambio de la configuración de un grupo de implementación	472
Cambie la configuración del grupo de implementación (consola)	472
Cambie la configuración del grupo de implementación (CLI)	473
Configuración de las opciones avanzadas de un grupo de implementación	475
Eliminación de un grupo de implementación	478
Eliminar un grupo de implementación (consola)	479

Eliminar un grupo de implementación (CLI)	479
Gestión de revisiones de aplicaciones	480
Planificación de una revisión	480
Añadir un AppSpec archivo	481
Añadir un AppSpec archivo para una implementación de Amazon ECS	481
Agregar un AppSpec archivo para una implementación de AWS Lambda	485
Agregue un AppSpec archivo para una implementación local de EC2/	487
Elección de un tipo de repositorio	491
Envío de una revisión	494
Impulsa una revisión mediante el AWS CLI	496
Visualización de los detalles de una revisión de aplicación	498
Visualización de los detalles de una revisión de aplicación (consola)	499
Visualización de los detalles de una revisión de aplicación (CLI)	499
Registro de una revisión de una aplicación	501
Registrar una revisión en Amazon S3 con CodeDeploy (CLI)	501
Registrar una revisión GitHub con CodeDeploy (CLI)	502
Trabajo con implementaciones	504
Creación de una implementación	505
Requisitos previos para una implementación	506
Creación de una implementación en una plataforma de informática de Amazon ECS (consola)	509
Creación de una implementación en la plataforma de informática de AWS Lambda (consola)	511
Creación de una implementación en una plataforma de informática de EC2/en las instalaciones (consola)	513
Creación de una implementación en una plataforma de informática de Amazon ECS (CLI)	518
Creación de una implementación en la plataforma de informática de AWS Lambda (CLI) ...	520
Creación de una implementación en una plataforma de informática EC2/en las instalaciones (CLI)	522
Cree una implementación azul/verde de Amazon ECS mediante AWS CloudFormation	526
Visualización de los detalles de la implementación	529
Visualización de los detalles de la implementación (consola)	530
Visualización de los detalles de la implementación (CLI)	531
Visualización de los datos de registro de una implementación	531
Ver los datos de los archivos de registro en la CloudWatch consola de Amazon	532

Visualización de los archivos de registro de una instancia	532
Detención de una implementación	535
Detención de una implementación (consola)	536
Detener una implementación (CLI)	537
Volver a implementar y restaurar una implementación	537
Restauraciones automáticas	537
Restauraciones manuales	538
Flujo de trabajo de la restauración y reimplementación	538
Comportamiento de la restauración con contenido existente	540
Implementar una aplicación en una AWS cuenta diferente	544
Paso 1: Crear un bucket de S3 en cualquiera de las cuentas	544
Paso 2: Conceder permisos de bucket de Amazon S3 al perfil de instancia de IAM de la cuenta de producción	545
Paso 3: Crear recursos y un rol entre cuentas en la cuenta de producción	546
Paso 4: Cargar la revisión de la aplicación al bucket de Amazon S3	547
Paso 5: Asumir el rol entre cuentas e implementar las aplicaciones	548
Validación de un paquete de implementación en una máquina local	548
Requisitos previos	549
Creación de una implementación local	551
Ejemplos	554
Monitorización de las implementaciones	556
Herramientas automatizadas	556
Herramientas manuales	558
Supervisión de las implementaciones con las herramientas de Amazon CloudWatch	559
Supervisión de los despliegues con alarmas CloudWatch	559
Supervisión de despliegues con Amazon Events CloudWatch	561
Supervisar los despliegues con AWS CloudTrail	564
CodeDeploy información en CloudTrail	564
Descripción de las entradas de los archivos de CodeDeploy registro	565
Monitorización de implementaciones con notificaciones de eventos de Amazon SNS	567
Concesión de permisos de Amazon SNS a un rol de servicio	568
Crea un desencadenante para un CodeDeploy evento	569
Edición de un disparador en un grupo de implementación	576
Eliminación de un disparador de un grupo de implementación	578
Formatos de datos JSON para los disparadores	580
Seguridad	582

Protección de datos	583
Privacidad del tráfico entre redes	584
Cifrado en reposo	584
Cifrado en tránsito	585
Administración de claves de cifrado	585
Administración de identidades y accesos	585
Público	586
Autenticación con identidades	586
Administración de acceso mediante políticas	590
¿Cómo AWS CodeDeploy funciona con IAM	592
AWS políticas gestionadas (predefinidas) para CodeDeploy	597
CodeDeploy actualizaciones de las políticas AWS gestionadas	604
Ejemplos de políticas basadas en identidad	606
Resolución de problemas	613
Referencia de permisos de CodeDeploy	615
Prevención de la sustitución confusa entre servicios	624
Respuesta frente a incidencias	626
Auditoría de todas las interacciones con CodeDeploy	626
Administración de incidentes y alertas	626
Validación de conformidad	627
Resiliencia	628
Seguridad de la infraestructura	629
Referencia	630
AppSpec referencia de fichero	630
AppSpec archivos en una plataforma informática Amazon ECS	631
AppSpec archivos en una plataforma AWS Lambda informática	631
AppSpec archivos en una plataforma informática local EC2/	631
AppSpec Estructura de archivos	632
AppSpec Ejemplo de archivo	680
AppSpec Espaciado de archivos	686
Valide su AppSpec archivo y su ubicación	688
Referencia de configuración del agente	688
Temas relacionados de	693
AWS CloudFormation referencia de plantilla	694
Úselo CodeDeploy con Amazon Virtual Private Cloud	697
Disponibilidad	698

Cree puntos finales de VPC para CodeDeploy	701
Configure los permisos del CodeDeploy agente y de IAM	701
Referencia del kit de recursos	702
Nombres de bucket del kit de recursos por región	702
Contenido del kit de recursos	705
Mostrar una lista de los archivos del kit de recursos	707
Descargar los archivos del kit de recursos	708
Cuotas	711
Resolución de problemas	719
Solución de problemas generales	719
Lista de comprobación de solución de problemas generales	720
CodeDeploy los recursos de despliegue solo se admiten en algunas regiones AWS	721
Los procedimientos de esta guía no coinciden con los de la CodeDeploy consola	722
Los roles de IAM necesarios no están disponibles	722
El uso de algunos editores de texto para crear AppSpec archivos y scripts de shell puede provocar errores en las implementaciones	723
El uso de Finder en macOS para empaquetar una revisión de la aplicación puede producir un error en la implementación	724
Resolución de problemas de implementación de EC2/en las instalaciones	724
CodeDeploy error CommandPoller de falta de credenciales en el complemento	725
La implementación produce un error con el mensaje "Validation of PKCS7 signed message failed"	726
La implementación o la reimplementación de los mismos archivos en la misma instancia producen el error "The deployment failed because a specified file already exists at this location"	726
Las rutas de archivo largas provocan el error "No existe tal archivo o directorio"	729
Los procesos de larga duración pueden producir un error en la implementación	730
Solución de problemas de un evento AllowTraffic de ciclo de vida fallido sin que se haya registrado ningún error en los registros de implementación	731
Solución de problemas relacionados con un error o un evento del ciclo de vida de ApplicationStop la BeforeBlockTraffic implementación AfterBlockTraffic	732
Solución de problemas de un evento fallido en el ciclo de vida de una DownloadBundle implementación con UnknownError: no abierto para lectura	733
Solución de problemas cuando se omiten todos los eventos del ciclo de vida	734
PowerShell Los scripts de Windows no utilizan la versión de 64 bits de Windows de forma predeterminada PowerShell	736

Resolución problemas de implementación de Amazon ECS	737
Se agota el tiempo de espera mientras se espera a que se establezca la tarea de reemplazo	738
Se agota el tiempo de espera mientras se espera a que continúe una notificación	739
El rol de IAM no tiene permisos suficientes	739
Se agotó el tiempo de espera de la implementación mientras se esperaba una llamada de estado	740
La implementación falló porque una o más de las funciones de validación de eventos del ciclo de vida fallaron	741
No se pudo actualizar el ELB debido al siguiente error: el grupo objetivo del conjunto de tareas principal debe estar detrás del oyente	741
Mi implementación a veces falla cuando uso el escalado automático	742
Solo ALB admite el enrutamiento de tráfico gradual; en su lugar, utilice el enrutamiento de AllAtOnce tráfico cuando cree o actualice un grupo de implementación	743
A pesar de que mi implementación se realizó correctamente, el conjunto de tareas de reemplazo no pasa las comprobaciones de estado de Elastic Load Balancing y mi aplicación no funciona	744
¿Puedo asociar varios equilibradores de carga a un grupo de implementación?	745
¿Puedo realizar despliegues CodeDeploy azules o verdes sin un balanceador de carga? ...	745
¿Cómo puedo actualizar mi servicio de Amazon ECS con nueva información durante una implementación?	745
Solución de problemas de implementación de AWS Lambda	746
AWS Lambda las implementaciones fallan después de detener manualmente una implementación de Lambda que no tiene reversiones configuradas	746
Solución de problemas de grupos de implementación	747
Cuando se etiqueta una instancia como parte de un grupo de implementaciones, no se implementa automáticamente la aplicación en la nueva instancia	747
Solución de problemas de instancias	747
Las etiquetas deben estar configuradas correctamente	748
AWS CodeDeploy el agente debe estar instalado y ejecutándose en las instancias	748
Las implementaciones no producen error hasta una hora después cuando se termina una instancia durante una implementación	748
Análisis de archivos de registro para investigar los errores de implementación en instancias	749
Cree un nuevo archivo de CodeDeploy registro si se ha eliminado accidentalmente	749

Solución de errores de despliegue «InvalidSignatureException — Firma caducada: [hora] es anterior a [hora]»	749
Solucione problemas con los GitHub tokens	750
El token de GitHub OAuth no es válido	750
Se ha superado el número máximo de tokens de GitHub OAuth	750
Solución de problemas de Amazon EC2 Auto Scaling	751
Solución de problemas generales de Amazon EC2 Auto Scaling	751
«CodeDeployRole no le da permiso para realizar operaciones en el siguiente AWS servicio: AmazonAutoScaling» error	753
Las instancias de un grupo de Amazon EC2 Auto Scaling se aprovisionan y se terminan constantemente antes de que pueda implementarse una revisión	753
La terminación o el reinicio de una instancia de Amazon EC2 Auto Scaling puede producir un error en las implementaciones	754
Evite asociar varios grupos de implementación a un único grupo de Amazon EC2 Auto Scaling	755
Las instancias de EC2 de un grupo de Amazon EC2 Auto Scaling no se pueden lanzar y aparece el error "Heartbeat Timeout"	756
Los enlaces de ciclo de vida de Amazon EC2 Auto Scaling que no coincidan pueden provocar que las implementaciones automáticas en los grupos de Amazon EC2 Auto Scaling se detengan o fallen.	758
Error “La implementación falló porque no se encontró ninguna instancia para su grupo de implementación”	760
Códigos de error	768
Temas relacionados de	774
Recursos	775
Guías de referencia y recursos de soporte	775
Muestras	775
Blogs	775
AWS kits y herramientas de desarrollo de software	776
Historial de documentos	777
Actualizaciones anteriores	795
AWS Glosario	820
.....	dcccxxi

¿Qué es CodeDeploy?

CodeDeploy es un servicio de implementación que automatiza las implementaciones de aplicaciones en instancias de Amazon EC2, instancias locales, funciones Lambda sin servidor o servicios de Amazon ECS.

Puede implementar una variedad de contenidos de aplicación prácticamente ilimitada, entre las que se incluyen:

- Código
- Funciones sin servidor AWS Lambda
- Archivos de configuración y web
- Executables
- Paquetes
- Scripts
- Archivos multimedia

CodeDeploy puede implementar contenido de aplicaciones que se ejecute en un servidor y se almacene en buckets, GitHub repositorios o repositorios de Bitbucket de Amazon S3. CodeDeploy también puede implementar una función Lambda sin servidor. No necesita realizar cambios en el código existente para poder usarlo. CodeDeploy

CodeDeploy le facilita:

- Lanzar rápidamente nuevas características.
- Actualiza las versiones de las AWS Lambda funciones.
- Evitar tiempos de inactividad durante la implementación de las aplicaciones.
- Gestionar el complejo proceso de actualización de las aplicaciones, evitando muchos de los riesgos asociados a las implementaciones manuales, que son susceptibles a errores.

El servicio se adapta a su infraestructura, por lo que puede implementar fácilmente en una sola instancia o en miles de ellas.

CodeDeploy funciona con varios sistemas para la gestión de la configuración, el control de fuentes, [la integración continua](#), [la entrega](#) continua y el despliegue continuo. Para obtener más información, consulte [Integraciones de productos](#).

La CodeDeploy consola también proporciona una forma de buscar rápidamente sus recursos, como repositorios, proyectos de compilación, aplicaciones de implementación y canalizaciones. Elija Ir a recurso o pulse la tecla / y, a continuación, escriba el nombre del recurso. Se muestran todas las coincidencias en la lista. En las búsquedas, no se distingue entre mayúsculas y minúsculas. Solo puede ver los recursos para los que tiene permiso. Para obtener más información, consulte [Administración de identidades y accesos para AWS CodeDeploy](#).

Temas

- [Ventajas de AWS CodeDeploy](#)
- [Descripción general de las plataformas informáticas CodeDeploy](#)
- [Descripción general de los tipos de implementación CodeDeploy](#)
- [Esperamos tener noticias tuyas](#)
- [CodeDeploy componentes principales](#)
- [CodeDeploy despliegues](#)
- [CodeDeploy archivos de especificación de la aplicación \(AppSpec\)](#)

Ventajas de AWS CodeDeploy

CodeDeploy ofrece los siguientes beneficios:

- Aplicaciones de servidor, sin servidor y de contenedor. CodeDeploy permite implementar tanto aplicaciones tradicionales en servidores como aplicaciones que implementan una versión de AWS Lambda función sin servidor o una aplicación de Amazon ECS.
- Implementaciones automatizadas. CodeDeploy automatiza completamente las implementaciones de sus aplicaciones en sus entornos de desarrollo, pruebas y producción. CodeDeploy escala con su infraestructura para que pueda implementarlas en una o en miles de instancias.
- Minimizar el tiempo de inactividad. Si su aplicación utiliza la plataforma informática local EC2, esto CodeDeploy ayuda a maximizar la disponibilidad de la aplicación. Durante una implementación in situ, CodeDeploy realiza una actualización progresiva en todas las instancias de Amazon EC2. Es posible especificar el número de instancias que se desactivan a la vez para efectuar las actualizaciones. Durante una implementación blue/green, la última revisión de la aplicación se

instala en las instancias de sustitución. El tráfico se desvía a estas instancias en el momento que desee, bien inmediatamente o cuando haya terminado de probar el nuevo entorno. Para ambos tipos de implementación, CodeDeploy realiza un seguimiento del estado de las aplicaciones de acuerdo con las reglas que configure.

- **Detención y reversión.** En caso de que haya errores, puede detener y revertir la implementación de forma automática o manual.
- **Control centralizado.** Puede iniciar sus implementaciones y realizar un seguimiento del estado de las mismas a través de la CodeDeploy consola o del AWS CLI. Recibirá un informe que indica el momento en que se implementó cada revisión de la aplicación en cada instancia de Amazon EC2.
- **Fácil de adoptar.** CodeDeploy es independiente de la plataforma y funciona con cualquier aplicación. Puedes reutilizar fácilmente tu código de configuración. CodeDeploy también se puede integrar con su proceso de lanzamiento de software o con su cadena de herramientas de entrega continua.
- **Implementaciones simultáneas.** Si tiene más de una aplicación que utiliza la plataforma informática local EC2, CodeDeploy puede implementarlas simultáneamente en el mismo conjunto de instancias.

Descripción general de las plataformas informáticas CodeDeploy

CodeDeploy puede implementar aplicaciones en tres plataformas informáticas:

- **EC2/en las instalaciones:** describe las instancias de servidores físicos que pueden ser instancias en la nube de Amazon EC2, servidores en las instalaciones o ambas cosas. Las aplicaciones creadas mediante la plataforma de informática EC2/en las instalaciones pueden estar formadas por archivos ejecutables, archivos de configuración, imágenes, etc.

Las implementaciones que usan la plataforma de informática EC2/en las instalaciones administran la forma en que el tráfico se envía a las instancias mediante un tipo de implementación local o azul/verde. Para obtener más información, consulte [Descripción general de los tipos de implementación CodeDeploy](#).


- **AWS Lambda:** se usa para implementar aplicaciones que constan de una versión actualizada de una función Lambda. AWS Lambda administra la función Lambda en un entorno informático sin servidor compuesto por una estructura informática de alta disponibilidad. Toda la administración de los recursos de cómputo la realiza. AWS Lambda Para obtener más información, consulte

[Aplicaciones y informática sin servidor](#). Para obtener más información acerca de AWS Lambda las funciones Lambda, consulte [AWS Lambda](#)

Puede administrar la forma en que el tráfico se desplaza a las versiones actualizadas de la función Lambda durante una implementación eligiendo una configuración canaria, lineal o all-at-once lineal.

- Amazon ECS: se utiliza para implementar una aplicación en contenedor de Amazon ECS como un conjunto de tareas. CodeDeploy realiza una implementación azul/verde mediante la instalación de una versión actualizada de la aplicación como un nuevo conjunto de tareas de reemplazo. CodeDeploy redirige el tráfico de producción del conjunto de tareas de la aplicación original al conjunto de tareas de reemplazo. Cuando la implementación se realiza correctamente, se termina el conjunto de tareas original. Para obtener más información acerca de Amazon ECS, consulte [Amazon Elastic Container Service](#).

Puede administrar la forma en que el tráfico se desplaza al conjunto de tareas actualizado durante una implementación eligiendo una configuración canaria, lineal o all-at-once lineal.

 Note

Las implementaciones azul/verde de Amazon ECS son compatibles con y. CodeDeploy AWS CloudFormation Los detalles de estas implementaciones se describen en secciones posteriores.

En la siguiente tabla se describe cómo se utilizan CodeDeploy los componentes con cada plataforma de cómputo. Para obtener más información, consulte:

- [Trabajar con grupos de despliegue en CodeDeploy](#)
- [Trabajar con despliegues en CodeDeploy](#)
- [Trabajar con configuraciones de implementación en CodeDeploy](#)
- [Trabajar con revisiones de aplicaciones para CodeDeploy](#)
- [Trabajar con aplicaciones en CodeDeploy](#)

CodeDeploy componente	EC2/en las instalaciones	AWS Lambda	Amazon ECS
Grupo de implementación	Implementa una revisión en un conjunto de instancias.	Implementa una nueva versión de una función de Lambda sin servidor en una infraestructura informática de alta disponibilidad.	Especifica el servicio de Amazon ECS con la aplicación en contenedor para implementar como un conjunto de tareas, un oyente de producción y de prueba opcional que se utiliza para enviar tráfico a la aplicación implementada, cuándo redirigir el tráfico y terminar el conjunto de tareas original de la

CodeDeploy componente	EC2/en las instalaciones	AWS Lambda	Amazon ECS
			aplicación implementada y configuración opcional de activador, alarma y restauración.

CodeDeploy componente	EC2/en las instalaciones	AWS Lambda	Amazon ECS
Implementación	Implementa una nueva revisión que consta de una aplicación y AppSpec un archivo. AppSpec Especifica a cómo implementar la aplicación en las instancias de un grupo de implementación.	Desvía el tráfico de producción desde una versión de una función de Lambda a una nueva versión de la misma función. El AppSpec archivo especifica qué versión de la función Lambda se debe implementar.	Implementa a una versión actualizada de una aplicación contenerizada de Amazon ECS como un nuevo conjunto de tareas de reemplazo. CodeDeploy y redirige el tráfico de producción del conjunto de tareas con la versión original al nuevo conjunto de tareas de reemplazo con la versión

CodeDeploy componente	EC2/en las instalaciones	AWS Lambda	Amazon ECS
			actualizada. Cuando se completa la implementación, se termina el conjunto de tareas original.
Configuración de implementación	Configuración que determina la velocidad de implementación y el número mínimo de instancias que deben estar en buen estado en cualquier momento durante una implementación.	Configuración que determina cómo se desvía el tráfico a las versiones de la función de Lambda actualizadas.	Configuración que determina cómo se desvía el tráfico al conjunto de tareas actualizado de Amazon ECS.

CodeDeploy componente	EC2/en las instalaciones	AWS Lambda	Amazon ECS
Revisión	Combinación de un AppSpec archivo y archivos de aplicación, como ejecutables, archivos de configuración, etc.	AppSpec Archivo que especifica qué función de Lambda se debe implementar y qué funciones de Lambda pueden ejecutar pruebas de validación durante los ganchos de eventos del ciclo de vida de la implementación.	<p>Un AppSpec archivo que especifica:</p> <ul style="list-style-type: none"> • La definición de tareas de Amazon ECS para el servicio de Amazon ECS con la aplicación en contenedores que se va a implementar. • El contenedor en el que se implementa la aplicación

CodeDeploy componente	EC2/en las instalaciones	AWS Lambda	Amazon ECS
			<p>actualizada.</p> <ul style="list-style-type: none">• Un puerto para el contenedor donde se redirige el tráfico de producción.• Ajustes de configuración de red opcionales y funciones de Lambda que pueden ejecutar pruebas de validación durante enlaces de

CodeDeploy componente	EC2/en las instalaciones	AWS Lambda	Amazon ECS
			eventos de ciclo de vida de implementación.
Aplicación	Colección de grupos de implementación y revisiones. Una aplicación de EC2/en las instalaciones utiliza la plataforma de informática de EC2/en las instalaciones.	Colección de grupos de implementación y revisiones. Una aplicación utilizada para una AWS Lambda implementación usa la plataforma de procesamiento AWS Lambda sin servidor.	Colección de grupos de implementación y revisiones. Una aplicación utilizada para una implementación de Amazon ECS utiliza la plataforma de informática de Amazon ECS.

Descripción general de los tipos de implementación CodeDeploy

CodeDeploy ofrece dos opciones de tipos de despliegue:

- **Implementación local:** la aplicación de cada instancia del grupo de implementación se para, se instala la última revisión de la aplicación, y se inicia y valida la nueva versión de la aplicación.

Puede utilizar un equilibrador de carga de modo que se cancele el registro de cada instancia durante su implementación y, a continuación, vuelva a ponerse en servicio una vez completada la implementación. Solo las implementaciones que usan la plataforma de informática de EC2/en las instalaciones pueden utilizar implementaciones locales. Para obtener más información acerca de las implementaciones locales, consulte [Información general de la implementación local](#).

Note

AWS Las implementaciones de Lambda y Amazon ECS no pueden usar un tipo de implementación local.

- Implementación azul/verde: el funcionamiento de la implementación depende de la plataforma de informática que utilice:
 - Azul/verde en una plataforma de informática EC2/en las instalaciones: las instancias de un grupo de implementación (el entorno original) se sustituyen por otro conjunto de instancias (el entorno de sustitución) siguiendo estos pasos:
 - Las instancias se aprovisionan en el entorno de sustitución.
 - La revisión de aplicación más reciente se instala en las instancias de sustitución.
 - Existe un tiempo de espera opcional para actividades como las pruebas de aplicaciones y la verificación del sistema.
 - Las instancias del entorno de sustitución se registran en uno o más equilibradores de carga de Elastic Load Balancing, lo que provoca que el tráfico se redirija a ellos. Las instancias del entorno original se anulan del registro y se pueden cerrar o mantener en ejecución para otros usos.

Note

Si utiliza una plataforma de informática EC2/en las instalaciones, tenga en cuenta que las implementaciones azul/verde solo funcionan con instancias de Amazon EC2.

- Azul/verde en una plataforma informática o AWS Lambda Amazon ECS: el tráfico se desplaza en incrementos según una configuración canaria, lineal o all-at-once implementación.
- Implementaciones azules/verdes AWS CloudFormation: el tráfico pasa de los recursos actuales a los recursos actualizados como parte de una actualización de la pila. AWS CloudFormation Actualmente, solo se admiten implementaciones de ECS blue/green (azul/verde).

Para obtener más información acerca de las implementaciones blue/green, consulte [Información general de la implementación azul/verde](#).

Note

Con el CodeDeploy agente, puede realizar una implementación en una instancia en la que haya iniciado sesión sin necesidad de una aplicación, un grupo de implementación o incluso una cuenta. AWS Para obtener más información, consulte [Utilice el CodeDeploy agente para validar un paquete de despliegue en una máquina local](#).

Temas

- [Información general de la implementación local](#)
- [Información general de la implementación azul/verde](#)

Información general de la implementación local

Note

AWS Las implementaciones de Lambda y Amazon ECS no pueden usar un tipo de implementación local.

Así es como funciona una implementación local:

1. En primer lugar, debe crear contenido desplegable en su máquina de desarrollo local o entorno similar y, a continuación, añadir un archivo de especificaciones de la aplicación (archivo).
AppSpec El AppSpec archivo es exclusivo de. CodeDeploy Define las acciones de despliegue que CodeDeploy desea ejecutar. Agrupa el contenido desplegable y el AppSpec archivo en un archivo y, a continuación, lo carga en un depósito o GitHub repositorio de Amazon S3. Este archivo de almacenamiento se denomina revisión de la aplicación (o simplemente revisión).
2. A continuación, debe proporcionar CodeDeploy información sobre su implementación, como el GitHub depósito o repositorio de Amazon S3 desde el que extraer la revisión y en qué conjunto de instancias de Amazon EC2 se debe implementar su contenido. CodeDeploy llama grupo de implementación a un conjunto de instancias de Amazon EC2. Un grupo de implementación

contiene instancias de Amazon EC2 etiquetadas individualmente, instancias de Amazon EC2 en grupos de Amazon EC2 Auto Scaling o ambas.

Cada vez que cargue correctamente una nueva revisión de una aplicación que desee implementar en el grupo de implementaciones, el paquete correspondiente se marca como revisión de destino para ese grupo de implementaciones. En otras palabras, la revisión de la aplicación que debe implementarse en cada momento es la revisión de destino. También es la revisión que se extrae para las implementaciones automáticas.

3. A continuación, el CodeDeploy agente de cada instancia sondea CodeDeploy para determinar qué y cuándo extraer del GitHub depósito o repositorio de Amazon S3 especificado.
4. Por último, el CodeDeploy agente de cada instancia extrae la revisión de destino del GitHub depósito o repositorio de Amazon S3 y, siguiendo las instrucciones del AppSpec archivo, despliega el contenido en la instancia.

CodeDeploy mantiene un registro de las implementaciones para que pueda obtener el estado de las mismas, los parámetros de configuración de las mismas, el estado de las instancias, etc.

Información general de la implementación azul/verde

Se utiliza una implementación azul/verde para actualizar las aplicaciones y, al mismo tiempo, minimizar las interrupciones causadas por los cambios de una nueva versión de la aplicación. CodeDeploy aprovisiona la nueva versión de la aplicación junto con la versión anterior antes de redirigir el tráfico de producción.

- AWS Lambda: el tráfico pasa de una versión de una función Lambda a una nueva versión de la misma función Lambda.
- Amazon ECS: se desvía el tráfico desde un conjunto de tareas de su servicio de Amazon ECS a un conjunto de tareas de sustitución actualizado en el mismo servicio de Amazon ECS.
- EC2/en las instalaciones: se desvía el tráfico desde un conjunto de instancias en el entorno original a un conjunto de instancias de sustitución.


Todas las implementaciones de AWS Lambda y Amazon ECS son de color azul/verde. Una implementación de EC2/en las instalaciones puede ser local o azul/verde. Una implementación azul/verde ofrece diversas ventajas con respecto a la implementación local:

- Puede instalar y probar una aplicación en el nuevo entorno de sustitución e implementarla en producción con solo redirigir el tráfico.

- Si utiliza la plataforma de informática de EC2/en las instalaciones, es más rápido y seguro volver a cambiar a la versión más reciente de una aplicación. Esto se debe a que el tráfico puede volver a dirigirse a las instancias originales, siempre que no se hayan terminado. Con una implementación in situ, las versiones deben revertirse a la versión anterior de la aplicación.
- Si utiliza la plataforma informática local EC2, se aprovisionan nuevas instancias para una implementación azul/verde y reflejan la mayoría de las configuraciones de servidor. up-to-date Esto ayuda a evitar la clase de problemas que a veces aparecen en las instancias que llevan mucho tiempo en ejecución.
- Si utiliza la plataforma de procesamiento AWS Lambda, usted controla cómo se transfiere el tráfico de la versión original de la función AWS Lambda a la nueva versión de la función AWS Lambda.
- Si utiliza la plataforma de informática de Amazon ECS, puede controlar cómo se desvía el tráfico del conjunto de tareas original al nuevo conjunto de tareas.

Una implementación azul/verde AWS CloudFormation puede usar uno de los siguientes métodos:

- AWS CloudFormation plantillas para despliegues: cuando configuras despliegues con AWS CloudFormation plantillas, los despliegues se activan mediante actualizaciones. AWS CloudFormation Cuando cambias un recurso y subes un cambio de plantilla, una actualización de la pila AWS CloudFormation inicia la nueva implementación. Para obtener una lista de los recursos que puede utilizar en AWS CloudFormation las plantillas, consulte [AWS CloudFormation plantillas de CodeDeploy referencia](#).
- Desde despliegues azules/verdes hasta AWS CloudFormation: puedes utilizarlos AWS CloudFormation para gestionar tus despliegues azules/verdes mediante actualizaciones de paquetes. Defina los recursos blue y green (azules y verdes), además de especificar la configuración de enrutamiento y estabilización del tráfico dentro de la plantilla de pila. A continuación, si actualizas los recursos seleccionados durante una actualización de pila, AWS CloudFormation generas todos los recursos verdes necesarios, desplaza el tráfico en función de los parámetros de enrutamiento de tráfico especificados y elimina los recursos azules. Para obtener más información, consulte [Automatizar las implementaciones azul/verde de Amazon ECS CodeDeploy mediante su uso AWS CloudFormation](#) en la Guía del AWS CloudFormation usuario.

 Note

Solo se admite para implementaciones azul/verde de Amazon ECS.

La forma de configurar una implementación blue/green depende la plataforma de informática que utilice la implementación.

Implementación azul/verde en una plataforma informática o AWS Lambda Amazon ECS

Si utiliza la AWS Lambda plataforma informática Amazon ECS, debe indicar cómo se desplaza el tráfico de la AWS Lambda función original o del conjunto de tareas de Amazon ECS a la nueva función o conjunto de tareas. Para indicar cómo se desvía el tráfico, debe especificar una de las siguientes configuraciones de implementación:

- valor controlado
- lineal
- all-at-once

Para obtener información sobre cómo se desplaza el tráfico en configuraciones canarias, lineales o de all-at-once despliegue, consulte [Configuración de implementación](#).

Para obtener más información sobre la configuración de implementación de Lambda, consulte [Configuraciones de implementación en una plataforma de informática de AWS Lambda](#).

Para obtener más información sobre la configuración de implementación de Amazon ECS, consulte [Configuraciones de implementación en una plataforma de informática de Amazon ECS](#).

Implementación azul/verde en una plataforma de informática EC2/en las instalaciones


Note

Debe usar instancias de Amazon EC2 para las implementaciones azul/verde en la plataforma de informática de EC2/en las instalaciones. No se admiten instancias en las instalaciones para el tipo de implementación azul/verde.

Si utiliza la plataforma de informática de EC2/en las instalaciones, tenga en cuenta lo siguiente:

Debe tener una o varias instancias de Amazon EC2 con etiquetas de identificación de Amazon EC2 o un grupo de Amazon EC2 Amazon EC2 Auto Scaling. Las instancias también deben cumplir los requisitos siguientes:

- Cada instancia de Amazon EC2 debe tener asociado el perfil de instancia de IAM correcto.
- El CodeDeploy agente debe estar instalado y en ejecución en cada instancia.

 Note

Normalmente también hay una revisión de la aplicación ejecutándose en las instancias del entorno original, pero esto no es un requisito para una implementación blue/green.

Al crear un grupo de implementaciones para utilizarlo en implementaciones blue/green, puede elegir el modo de especificar el entorno de sustitución:

Copie un grupo de Auto Scaling de Amazon EC2 existente: durante la implementación azul/verde, CodeDeploy crea las instancias para su entorno de reemplazo durante la implementación. Con esta opción, CodeDeploy utiliza el grupo de Auto Scaling de Amazon EC2 que especifique como plantilla para el entorno de reemplazo, que incluye el mismo número de instancias en ejecución y muchas otras opciones de configuración.

Elegir las instancias manualmente: puede especificar las instancias consideradas de sustitución mediante etiquetas de instancia de Amazon EC2, nombres de grupo de Amazon EC2 Auto Scaling o ambos. Si elige esta opción, no tendrá que especificar las instancias del entorno de sustitución hasta el momento de crear una implementación.

Así es como funciona:

1. Ya cuenta con instancias o un grupo de Amazon EC2 Auto Scaling que sirven como entorno original. La primera vez que ejecute una implementación blue/green, normalmente usará instancias que ya se habrán utilizado en una implementación in situ.
2. En una CodeDeploy aplicación existente, se crea un grupo de implementaciones azul o verde en el que, además de las opciones necesarias para una implementación local, se especifica lo siguiente:
 - El equilibrador de carga que dirigirá el tráfico del entorno original al entorno de sustitución durante el proceso de implementación azul/verde.
 - Si el tráfico debe dirigirse al entorno de sustitución de inmediato o esperar a que lo redirija manualmente.
 - La velocidad con la que se dirige el tráfico a las instancias de sustitución.
 - Si las instancias sustituidas deben terminarse o mantenerse en ejecución.

3. Ahora creará una implementación para este grupo de implementaciones en la que ocurrirá lo siguiente:
 - a. Si decide copiar un grupo de Amazon EC2 Auto Scaling, se preparan instancias para el entorno de sustitución.
 - b. La revisión de aplicación especificada para la implementación se instala en las instancias de sustitución.
 - c. Si ha especificado un tiempo de espera en la configuración del grupo de implementaciones, la implementación se pausa. Este es el momento en el que puede realizar pruebas y verificaciones en el entorno de sustitución. Si no redirige manualmente el tráfico antes de que finalice el periodo de espera, la implementación se detiene.
 - d. Las instancias del entorno de sustitución se registran con un equilibrador de carga de Elastic Load Balancing y el tráfico comienza a dirigirse a ellas.
 - e. Las instancias del entorno original se borran del registro y se terminan o se mantienen en ejecución, según lo que haya especificado para el grupo de implementaciones.

Despliegue azul/verde mediante AWS CloudFormation

Puede gestionar las implementaciones CodeDeploy azules y verdes modelando sus recursos con una plantilla. AWS CloudFormation

Cuando modelas tus recursos azules/verdes con una AWS CloudFormation plantilla, creas una actualización de pila AWS CloudFormation que actualiza tu conjunto de tareas. El tráfico de producción pasa del conjunto de tareas original del servicio a un conjunto de tareas de sustitución o bien todo a la vez, con implementaciones lineales y tiempos de procesamiento, o bien con implementaciones de valores controlados. La actualización de la pila inicia una implementación en CodeDeploy. Puede ver el estado y el historial de la implementación en la plantilla CodeDeploy, pero no puede crear ni administrar CodeDeploy recursos fuera de la AWS CloudFormation plantilla.

Note

En el caso de las implementaciones en azul o verde AWS CloudFormation, no se crea una CodeDeploy aplicación o un grupo de implementaciones.

Este método solo admite implementaciones azul/verde de Amazon ECS. Para obtener más información sobre las implementaciones azul/verde, consulte [AWS CloudFormation Cree una implementación azul/verde de Amazon ECS mediante AWS CloudFormation](#)

Esperamos tener noticias tuyas

Agradecemos sus comentarios. [Para ponerse en contacto con nosotros, visite el foro. CodeDeploy](#)

Temas

- [Primary Components](#)
- [Deployments](#)
- [Application Specification Files](#)

CodeDeploy componentes principales

Antes de empezar a trabajar con el servicio, debe familiarizarse con los componentes principales del proceso de CodeDeploy implementación.

Temas

- [Aplicación](#)
- [Plataforma de informática](#)
- [Configuración de implementación](#)
- [Grupo de implementación](#)
- [Tipo de implementación](#)
- [Perfil de instancia IAM](#)
- [Revisión](#)
- [Rol de servicio](#)
- [Revisión de destino](#)
- [Otros componentes](#)

Aplicación

Una aplicación es un nombre que identifica de forma exclusiva la aplicación que desea implementar. CodeDeploy utiliza este nombre, que funciona como contenedor, para garantizar que se haga referencia a la combinación correcta de revisión, configuración de despliegue y grupo de despliegue durante un despliegue.

Plataforma de informática

Una plataforma de cómputo es una plataforma en la que se CodeDeploy despliega una aplicación. Existen tres plataformas de informática:

- EC2/en las instalaciones: describe las instancias de servidores físicos que pueden ser instancias en la nube de Amazon EC2, servidores en las instalaciones o ambas cosas. Las aplicaciones creadas mediante la plataforma de informática EC2/en las instalaciones pueden estar formadas por archivos ejecutables, archivos de configuración, imágenes, etc.

Las implementaciones que usan la plataforma de informática EC2/en las instalaciones administran la forma en que el tráfico se envía a las instancias mediante un tipo de implementación local o azul/verde. Para obtener más información, consulte [Descripción general de los tipos de implementación CodeDeploy](#).

- AWS Lambda: se usa para implementar aplicaciones que constan de una versión actualizada de una función Lambda. AWS Lambda administra la función Lambda en un entorno informático sin servidor compuesto por una estructura informática de alta disponibilidad. Toda la administración de los recursos de cómputo la realiza. AWS Lambda Para obtener más información, consulte [Aplicaciones y informática sin servidor](#). Para obtener más información acerca de AWS Lambda las funciones Lambda, consulte. [AWS Lambda](#)

Puede administrar la forma en que el tráfico se desplaza a las versiones actualizadas de la función Lambda durante una implementación eligiendo una configuración canaria, lineal o all-at-once lineal.

- Amazon ECS: se utiliza para implementar una aplicación en contenedor de Amazon ECS como un conjunto de tareas. CodeDeploy realiza una implementación azul/verde mediante la instalación de una versión actualizada de la aplicación como un nuevo conjunto de tareas de reemplazo. CodeDeploy redirige el tráfico de producción del conjunto de tareas de la aplicación original al conjunto de tareas de reemplazo. Cuando la implementación se realiza correctamente, se termina el conjunto de tareas original. Para obtener más información acerca de Amazon ECS, consulte [Amazon Elastic Container Service](#).

Puede administrar la forma en que el tráfico se desplaza al conjunto de tareas actualizado durante una implementación eligiendo una configuración canaria, lineal o all-at-once lineal.

Note

Las implementaciones azul/verde de Amazon ECS son compatibles con y. CodeDeploy AWS CloudFormation Los detalles de estas implementaciones se describen en secciones posteriores.

Configuración de implementación

Una configuración de despliegue es un conjunto de reglas de despliegue y condiciones de éxito y fracaso que se utilizan CodeDeploy durante un despliegue. Si su implementación utiliza la plataforma de informática de EC2/en las instalaciones, puede especificar el número mínimo de instancias en buen estado para la implementación. Si su implementación utiliza la AWS Lambda plataforma informática Amazon ECS, puede especificar cómo se enruta el tráfico a la función Lambda actualizada o al conjunto de tareas de ECS.

Para obtener más información sobre cómo especificar el número mínimo de hosts en buen estado para una implementación que usa la plataforma de informática de EC2/en las instalaciones, consulte [Acerca del número mínimo de instancias en buen estado](#).

Las siguientes configuraciones de implementación especifican cómo se dirige el tráfico durante una implementación que usa la plataforma de informática de ECS o Lambda:

- **Valor controlado:** el tráfico se desvía en dos incrementos. Puede elegir opciones de valor controlado predefinidas que especifiquen el porcentaje de tráfico desviado al conjunto de tareas de ECS o a la función de Lambda actualizada en el primer incremento y el intervalo, en minutos, antes de que el tráfico restante se desvíe en el segundo incremento.
- **Lineal:** el tráfico se desvía en incrementos iguales con el mismo número de minutos entre incrementos. Puede elegir opciones lineales predefinidas que especifiquen el porcentaje de tráfico desviado en cada incremento y el número de minutos entre cada incremento.
- **R II-at-once:** Todo el tráfico se desplaza de la función Lambda original o del conjunto de tareas de ECS a la función o conjunto de tareas actualizado de una sola vez.

Grupo de implementación

Un grupo de implementación es un conjunto de instancias individuales. Un grupo de implementación contiene instancias etiquetadas individualmente, instancias de Amazon EC2 en grupos Amazon EC2

Auto Scaling o ambas. Para obtener información acerca de etiquetas de instancias de Amazon EC2, consulte [Trabajo con etiquetas mediante la consola](#). Para obtener más información acerca de las instancias en las instalaciones, consulte [Working with On-Premises Instances](#). Para obtener más información sobre Amazon EC2 Auto Scaling, consulte [Integración CodeDeploy con Amazon EC2 Auto Scaling](#).

Tipo de implementación

Un tipo de implementación es un método que se utiliza para hacer que la última revisión de la aplicación esté disponible en las instancias de un grupo de implementación. Existen dos tipos de implementaciones:

- **Implementación local:** la aplicación de cada instancia del grupo de implementación se para, se instala la última revisión de la aplicación, y se inicia y valida la nueva versión de la aplicación. Puede utilizar un equilibrador de carga de modo que se cancele el registro de cada instancia durante su implementación y, a continuación, vuelva a ponerse en servicio una vez completada la implementación. Solo las implementaciones que usan la plataforma de informática de EC2/en las instalaciones pueden utilizar implementaciones locales. Para obtener más información acerca de las implementaciones locales, consulte [Información general de la implementación local](#).
- **Implementación azul/verde:** el funcionamiento de la implementación depende de la plataforma de informática que utilice:
 - **Azul/verde en una plataforma de informática EC2/en las instalaciones:** las instancias de un grupo de implementación (el entorno original) se sustituyen por otro conjunto de instancias (el entorno de sustitución) siguiendo estos pasos:
 - Las instancias se aprovisionan en el entorno de sustitución.
 - La revisión de aplicación más reciente se instala en las instancias de sustitución.
 - Existe un tiempo de espera opcional para actividades como las pruebas de aplicaciones y la verificación del sistema.
 - Las instancias del entorno de sustitución se registran en uno o más equilibradores de carga de Elastic Load Balancing, lo que provoca que el tráfico se redirija a ellos. Las instancias del entorno original se anulan del registro y se pueden cerrar o mantener en ejecución para otros usos.

Note

Si utiliza una plataforma de informática EC2/en las instalaciones, tenga en cuenta que las implementaciones azul/verde solo funcionan con instancias de Amazon EC2.

- Azul/verde en una plataforma informática o AWS Lambda Amazon ECS: el tráfico se desplaza en incrementos según una configuración canaria, lineal o all-at-once implementación.
- Implementaciones azules/verdes AWS CloudFormation: el tráfico pasa de los recursos actuales a los recursos actualizados como parte de una actualización de la pila. AWS CloudFormation Actualmente, solo se admiten implementaciones de ECS blue/green (azul/verde).

Para obtener más información acerca de las implementaciones blue/green, consulte [Información general de la implementación azul/verde](#).

Note

Las implementaciones azul/verde de Amazon ECS son compatibles con y. CodeDeploy AWS CloudFormation Los detalles de estas implementaciones se describen en secciones posteriores.

Perfil de instancia IAM

Un perfil de instancia de IAM es un rol de IAM que se vincula a las instancias de Amazon EC2. Este perfil incluye los permisos necesarios para acceder a los GitHub depósitos o repositorios de Amazon S3 en los que se almacenan las aplicaciones. Para obtener más información, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#).

Revisión

Una revisión es una versión de su aplicación. Una revisión de despliegue de AWS Lambda es un archivo con formato YAML o JSON que especifica información sobre la función de Lambda que se va a implementar. Una revisión de implementación local de EC2/es un archivo de almacenamiento que contiene el contenido fuente (código fuente, páginas web, archivos ejecutables y scripts de implementación) y un archivo de especificaciones de la aplicación (archivo). AppSpec AWS Las revisiones Lambda se pueden almacenar en buckets de Amazon S3. Las revisiones de EC2/on-premise se almacenan en depósitos o repositorios de Amazon S3. GitHub En el caso de Amazon S3,

una revisión se identifica de forma exclusiva mediante su clave de objeto de Amazon S3 y ETag, su versión, o ambos elementos. GitHubEn efecto, una revisión se identifica de forma exclusiva por su ID de confirmación.

Rol de servicio

Un rol de servicio es un rol de IAM que otorga permisos a un AWS servicio para que pueda acceder a AWS los recursos. Las políticas que se adjuntan a la función de servicio determinan a qué AWS recursos puede acceder el servicio y las acciones que puede realizar con esos recursos. Para CodeDeploy, un rol de servicio se usa para lo siguiente:

- Para leer las etiquetas aplicadas a las instancias o los nombres de grupo de Amazon EC2 Auto Scaling asociados a las instancias. Esto le CodeDeploy permite identificar las instancias en las que puede implementar aplicaciones.
- Para realizar operaciones en instancias, grupos de Amazon EC2 Auto Scaling y equilibradores de carga de Elastic Load Balancing.
- Publicar información en los temas de Amazon SNS para que se puedan enviar notificaciones cuando se produzcan eventos de implementación o instancia específicos.
- Recuperar información sobre CloudWatch las alarmas a fin de configurar la supervisión de las alarmas para las implementaciones.

Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).

Revisión de destino

Una revisión de destino es la versión más reciente de la revisión de la aplicación que ha cargado en su repositorio y que desea implementar en las instancias de un grupo de implementación. Es decir, revisión de la aplicación que se va a incluir en la implementación actual. También es la revisión que se extrae para las implementaciones automáticas.

Otros componentes

Para obtener información sobre otros componentes del CodeDeploy flujo de trabajo, consulte los temas siguientes:

- [Elige un tipo de CodeDeploy repositorio](#)
- [Deployments](#)
- [Application Specification Files](#)

- [Instance Health](#)
- [Trabajando con el CodeDeploy agente](#)
- [Working with On-Premises Instances](#)

CodeDeploy despliegues

En este tema se proporciona información sobre los componentes y el flujo de trabajo de las implementaciones en CodeDeploy. El proceso de implementación varía en función de la plataforma informática o el método de implementación (Lambda, Amazon ECS, EC2/on-premise o directo AWS CloudFormation) que utilice para las implementaciones.

Temas

- [Implementaciones en una plataforma de informática de AWS Lambda](#)
- [Implementación en una plataforma de informática de Amazon ECS](#)
- [Implementación azul/verde en una plataforma de informática de EC2/en las instalaciones](#)

Implementaciones en una plataforma de informática de AWS Lambda

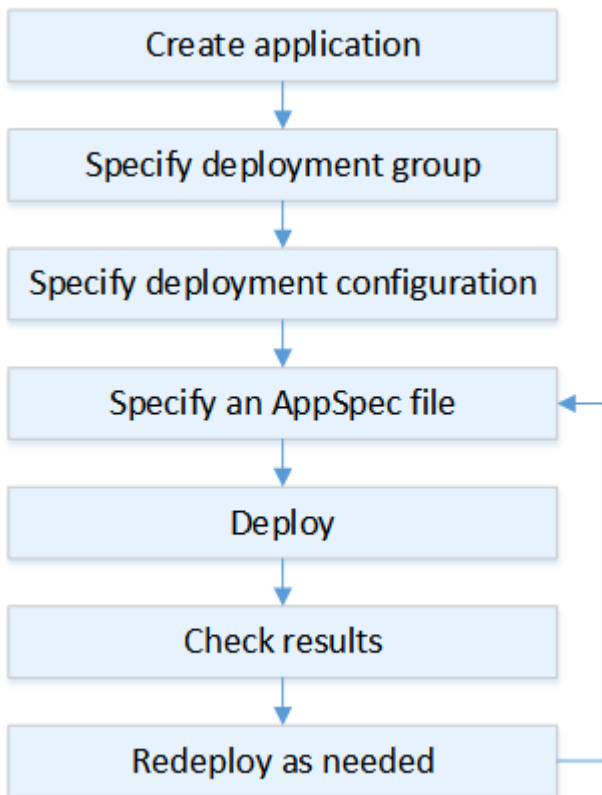
En este tema se proporciona información sobre los componentes y el flujo de trabajo de CodeDeploy las implementaciones que utilizan la plataforma de AWS procesamiento Lambda.

Temas

- [Flujo de trabajo de implementación en una plataforma de informática de AWS Lambda](#)
- [Carga de la revisión de la aplicación](#)
- [Creación de la aplicación y los grupos de implementación](#)
- [Implementación de la revisión de la aplicación](#)
- [Actualización de la aplicación](#)
- [Implementaciones detenidas y que producen error](#)
- [Reimplementaciones y restauración de implementaciones](#)

Flujo de trabajo de implementación en una plataforma de informática de AWS Lambda

En el siguiente diagrama se muestran los principales pasos en la implementación de funciones de AWS Lambda nuevas y actualizadas.



Estos pasos incluyen:

1. Cree una aplicación y asígnele un nombre que identifique las revisiones de la aplicación que desea implementar. Para implementar funciones Lambda, elija la plataforma de procesamiento AWS Lambda al crear la aplicación. CodeDeploy usa este nombre durante una implementación para asegurarse de que hace referencia a los componentes de implementación correctos, como el grupo de implementación, la configuración de la implementación y la revisión de la aplicación. Para obtener más información, consulte [Cree una aplicación con CodeDeploy](#).
2. Configure un grupo de implementaciones especificando el nombre del grupo.
3. Elija una configuración de despliegue para especificar cómo se transfiere el tráfico de la versión original de AWS Lambda la función a la nueva versión de la función Lambda. Para obtener más información, consulte [View Deployment Configuration Details](#).
4. Carga de un archivo de especificaciones de la aplicación (AppSpec archivo) a Amazon S3. El AppSpec archivo especifica una versión de la función Lambda y las funciones Lambda utilizadas para validar la implementación. Si no desea crear un AppSpec archivo, puede especificar una versión de la función Lambda y las funciones de validación del despliegue de Lambda

directamente en la consola mediante YAML o JSON. Para obtener más información, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).

5. Implemente la revisión de la aplicación en el grupo de implementación. AWS CodeDeploy implementa la revisión de la función Lambda que especificó. El tráfico se desplaza a la revisión de la función Lambda mediante el AppSpec archivo de despliegue que eligió al crear la aplicación. Para obtener más información, consulte [Cree una implementación con CodeDeploy](#).
6. Compruebe los resultados de la implementación. Para obtener más información, consulte [Supervisión de los despliegues en CodeDeploy](#).

Carga de la revisión de la aplicación

Coloque un AppSpec archivo en Amazon S3 o introdúzcalo directamente en la consola o AWS CLI. Para obtener más información, consulte [Application Specification Files](#).

Creación de la aplicación y los grupos de implementación

Un grupo de CodeDeploy implementación en una plataforma de procesamiento AWS Lambda identifica una colección de uno o más AppSpec archivos. Cada AppSpec archivo puede implementar una versión de la función Lambda. Un grupo de implementaciones también define un conjunto de opciones de configuración para implementaciones futuras, como alarmas y configuraciones de restauración.

Implementación de la revisión de la aplicación

Ahora está listo para implementar la revisión de la función especificada en el AppSpec archivo en el grupo de implementación. Puede usar la CodeDeploy consola o el comando [create-deployment](#). Hay parámetros que puede especificar para controlar la implementación, que incluyen la revisión, el grupo de implementaciones y la configuración de la implementación.

Actualización de la aplicación

Puede realizar actualizaciones en la aplicación y, a continuación, utilizar la CodeDeploy consola o ejecutar el comando [create-deployment](#) para realizar una revisión.

Implementaciones detenidas y que producen error

Puede usar la CodeDeploy consola o el comando [stop-deployment](#) para detener una implementación. Cuando se intenta detener la implementación, sucede una de estas tres cosas:

- La implementación se detiene y la operación devuelve el estado “realizada correctamente”. En este caso, no se ejecutan más eventos del ciclo de vida de implementación en el grupo de implementaciones para la implementación detenida.
- La implementación no se detiene inmediatamente y la operación devuelve el estado “pendiente”. En este caso, podrían seguir ejecutándose algunos eventos del ciclo de vida de implementación en el grupo de implementaciones. Después de finalizada la operación pendiente, las llamadas posteriores para detener la implementación devuelven el estado “realizada correctamente”.
- La implementación no se puede detener y la operación devuelve un error. Para obtener más información, consulta la [ErrorInformation](#) referencia sobre [errores comunes](#) en la AWS CodeDeploy API.

Al igual que las implementaciones detenidas, las implementaciones con errores pueden hacer que algunos eventos del ciclo de vida de implementación ya se hayan ejecutado. Para averiguar por qué ha fallado una implementación, puede usar la CodeDeploy consola o analizar los datos del archivo de registro de la implementación fallida. Para obtener más información, consulte [Limpieza de revisiones de aplicación y archivos de registro](#) y [Vea los datos de registro de las implementaciones locales CodeDeploy de EC2/](#).

Reimplementaciones y restauración de implementaciones

CodeDeploy implementa las reversiones al volver a implementar, como una nueva implementación, una revisión previamente implementada.

Puede configurar un grupo de implementación para que se restaure automáticamente si se dan ciertas condiciones, por ejemplo, si falla una implementación o si se alcanza el umbral de monitorización de alarmas. Además, puede anular la configuración de restauración especificada para el grupo de implementaciones en una implementación individual.

También puede optar por restaurar una implementación que produjo error manualmente, volviendo a implementar una revisión implementada anteriormente.

En todos los casos, a la implementación nueva o restaurada se le asigna su propio ID de implementación. La lista de despliegues que puede ver en la CodeDeploy consola muestra cuáles son el resultado de un despliegue automático.

Para obtener más información, consulte [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

Implementación en una plataforma de informática de Amazon ECS

En este tema se proporciona información sobre los componentes y el flujo de trabajo de CodeDeploy las implementaciones que utilizan la plataforma informática Amazon ECS.

Temas

- [Antes de empezar una implementación de Amazon ECS](#)
- [Flujo de trabajo de implementación \(alto nivel\) en una plataforma de informática de Amazon ECS](#)
- [¿Qué sucede durante una implementación de Amazon ECS?](#)
- [Carga de la revisión de la aplicación](#)
- [Creación de la aplicación y los grupos de implementación](#)
- [Implementación de la revisión de la aplicación](#)
- [Actualización de la aplicación](#)
- [Implementaciones detenidas y que producen error](#)
- [Reimplementaciones y restauración de implementaciones](#)
- [Implementaciones azul/verde de Amazon ECS a través de AWS CloudFormation](#)

Antes de empezar una implementación de Amazon ECS

Antes de empezar una implementación de aplicación de Amazon ECS, debe tener listo lo siguiente. Algunos requisitos se especifican al crear el grupo de implementaciones y otros se especifican en el AppSpec archivo.

Requisito	Donde se especifica
Clúster de Amazon ECS	Grupo de implementación
Servicio de Amazon ECS	Grupo de implementación
Equilibrador de carga de aplicación y equilibrador de carga de red	Grupo de implementación
Agente de escucha de producción	Grupo de implementación
Agente de escucha de prueba (opcional)	Grupo de implementación

Requisito	Donde se especifica
Dos grupos de destino	Grupo de implementación
Definición de tarea de Amazon ECS	AppSpec archivo
Nombre de contenedor	AppSpec archivo
Puerto del contenedor	AppSpec archivo

Clúster de Amazon ECS

Un clúster de Amazon ECS es una agrupación lógica de tareas o servicios. Al crear el grupo de implementación de la CodeDeploy aplicación, debe especificar el clúster de Amazon ECS que contiene su servicio de Amazon ECS. Para obtener más información, consulte [Clústeres de Amazon ECS](#) en la Guía del usuario de Amazon Elastic Container Service.

Servicio de Amazon ECS

Un servicio de Amazon ECS le ayuda a mantener y ejecutar un número determinado de instancias de una definición de tareas en un clúster de Amazon ECS. Su servicio Amazon ECS debe estar habilitado para CodeDeploy. De forma predeterminada, hay un servicio de Amazon ECS habilitado para implementaciones de Amazon ECS. Al crear su grupo de implementación, puede elegir implementar un servicio de Amazon ECS que se encuentra en su clúster de Amazon ECS. Para obtener más información, consulte [Servicios de Amazon ECS](#) en la Guía del usuario de Amazon Elastic Container Service.

Equilibrador de carga de aplicación y equilibrador de carga de red

Debe usar Elastic Load Balancing con el servicio de Amazon ECS que desee actualizar con una implementación de Amazon ECS. Puede usar un equilibrador de carga de aplicación o un equilibrador de carga de red. Le recomendamos un equilibrador de carga de aplicación para que pueda aprovechar características tales como el mapeo de puertos dinámico, el enrutamiento basado en rutas y las reglas de prioridad. El balanceador de carga se especifica al crear el grupo de implementación de la CodeDeploy aplicación. Para obtener más información, consulte [Configure un balanceador de carga, grupos objetivo y agentes de escucha para las implementaciones de CodeDeploy Amazon ECS](#) y [Creación de un equilibrador de carga](#) en la Guía del usuario de Amazon Elastic Container Service.

Uno o dos oyentes

El equilibrador de carga utiliza un oyente para dirigir el tráfico hacia los grupos de destino. Es obligatorio un agente de escucha de producción. Puede especificar un segundo agente de escucha de prueba opcional que dirija el tráfico hacia conjunto de tareas de sustitución mientras ejecuta las pruebas de validación. Especifique uno o ambos agentes de escucha al crear su grupo de implementaciones. Si utiliza la consola de Amazon ECS para crear su servicio de Amazon ECS, los oyentes se crean automáticamente. Para obtener más información, consulte [Oyentes para equilibradores de carga de aplicación](#) en la Guía del usuario de Elastic Load Balancing y [Creación de un servicio](#) en la Guía del usuario de Amazon Elastic Container Service.

Dos grupos de destino de Amazon ECS

Un grupo de destino se utiliza para dirigir el tráfico a un destino registrado. Una implementación de Amazon ECS requiere dos grupos de destino: uno para el conjunto de tareas original de la aplicación de Amazon ECS y otro para su conjunto de tareas de sustitución. Durante la implementación, CodeDeploy crea un conjunto de tareas de reemplazo y redirige el tráfico del conjunto de tareas original al nuevo. Los grupos de destino se especifican al crear el grupo de despliegue de la CodeDeploy aplicación.

Durante una implementación, CodeDeploy determina qué grupo objetivo está asociado al conjunto de tareas de su servicio Amazon ECS que tiene el estado PRIMARY (este es el conjunto de tareas original), asocia un grupo objetivo al conjunto de tareas y, a continuación, asocia el otro grupo objetivo al conjunto de tareas de reemplazo. Si realiza otra implementación, el grupo de destino asociado al conjunto de tareas original de la implementación actual se asocia al siguiente conjunto de tareas de sustitución de implementación. Para obtener más información, consulte [Grupos de destino para equilibradores de carga de aplicación](#) en la Guía del usuario de Elastic Load Balancing.

Una definición de tarea de Amazon ECS

Se requiere una definición de tarea para ejecutar el contenedor de Docker que contiene su aplicación de Amazon ECS. El ARN de la definición de la tarea se especifica en el archivo de CodeDeploy AppSpec la aplicación. Para obtener más información, consulte [Definiciones de tareas de Amazon ECS](#) en la Guía del usuario de Amazon Elastic Container Service y [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#).

Un contenedor de la aplicación de Amazon ECS

Un contenedor de Docker es una unidad de software que empaqueta código y sus dependencias para que pueda ejecutarse su aplicación. Un contenedor aísla la aplicación, por lo que se puede

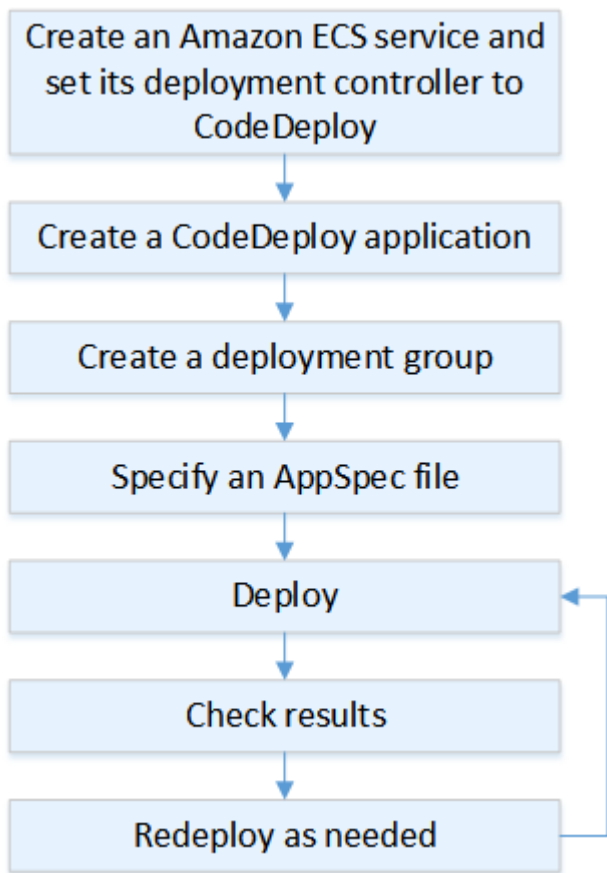
ejecutar en distintos entornos informáticos. El equilibrador de carga dirige el tráfico hacia un contenedor en el conjunto de tareas de la aplicación de Amazon ECS. El nombre del contenedor se especifica en el AppSpec archivo de CodeDeploy la aplicación. El contenedor especificado en el AppSpec archivo debe ser uno de los contenedores especificados en la definición de tareas de Amazon ECS. Para obtener más información, consulte [¿Qué es Amazon Elastic Container Service?](#) en la Guía del usuario de Amazon Elastic Container Service y [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#).

Un puerto para su conjunto de tareas de sustitución

Durante la implementación de Amazon ECS, el balanceador de carga dirige el tráfico a este puerto en el contenedor especificado en el AppSpec archivo de la CodeDeploy aplicación. El puerto se especifica en el AppSpec archivo de la CodeDeploy aplicación. Para obtener más información, consulte [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#).

Flujo de trabajo de implementación (alto nivel) en una plataforma de informática de Amazon ECS

En el diagrama siguiente se muestran los principales pasos en la implementación de servicios de Amazon ECS actualizados.



Estos pasos incluyen:

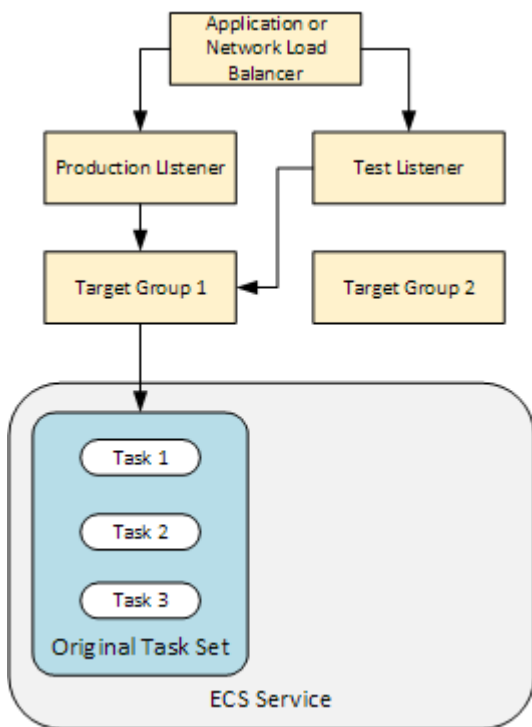
1. Cree una AWS CodeDeploy aplicación especificando un nombre que represente de forma exclusiva lo que desea implementar. Para implementar una aplicación Amazon ECS, en su AWS CodeDeploy aplicación, elija la plataforma informática Amazon ECS. CodeDeploy utiliza una aplicación durante una implementación para hacer referencia a los componentes de la implementación correctos, como el grupo de implementación, los grupos objetivo, los oyentes, el comportamiento de redireccionamiento del tráfico y la revisión de la aplicación. Para obtener más información, consulte [Cree una aplicación con CodeDeploy](#).
2. Configure un grupo de implementaciones especificando:
 - El nombre del grupo de implementaciones.
 - El nombre de servicio y clúster de Amazon ECS. El controlador de despliegue del servicio Amazon ECS debe estar configurado en CodeDeploy.
 - El agente de escucha de producción, un agente de escucha de prueba opcional y los grupos de destino utilizados durante una implementación.

- La configuración de implementación como, por ejemplo, cuándo redirigir el tráfico de producción al conjunto de tareas de Amazon ECS de sustitución en el servicio de Amazon ECS y cuándo terminar el conjunto de tareas de Amazon ECS original en el servicio de Amazon ECS.
 - Las configuraciones opcionales, como los desencadenadores, alarmas y comportamiento de restauración.
3. Especifique un archivo de especificaciones de la aplicación (AppSpec archivo). Puede cargarlo en Amazon S3, introducirlo en la consola en formato YAML o JSON, o especificarlo con el AWS CLI o el SDK. El AppSpec archivo especifica una definición de tarea de Amazon ECS para la implementación, un nombre de contenedor y una asignación de puertos que se utilizan para enrutar el tráfico, y las funciones Lambda que se ejecutan después de los enganches del ciclo de vida de la implementación. El nombre del contenedor debe ser un contenedor de su definición de tareas de Amazon ECS. Para obtener más información, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).
 4. Implemente la revisión de su aplicación. AWS CodeDeploy redirige el tráfico de la versión original de un conjunto de tareas de su servicio Amazon ECS a un nuevo conjunto de tareas de reemplazo. Los grupos de destino especificados en el grupo de implementaciones se utilizan para enviar tráfico a los conjuntos de tareas original y de sustitución. Cuando se completa la implementación, se termina el conjunto de tareas original. Puede especificar un agente de escucha de prueba opcional para enviar tráfico de prueba a la versión de sustitución antes de redirigir el tráfico a la misma. Para obtener más información, consulte [Cree una implementación con CodeDeploy](#).
 5. Compruebe los resultados de la implementación. Para obtener más información, consulte [Supervisión de los despliegues en CodeDeploy](#).

¿Qué sucede durante una implementación de Amazon ECS?

Antes de iniciar una implementación de Amazon ECS con un agente de escucha de prueba, debe configurar sus componentes. Para obtener más información, consulte [Antes de empezar una implementación de Amazon ECS](#).

En el siguiente diagrama se muestra la relación entre estos componentes cuando una implementación de Amazon ECS está lista para comenzar.



Cuando comienza la implementación, los eventos del ciclo de vida de la implementación empiezan a ejecutarse uno tras otro. Algunos eventos del ciclo de vida son enlaces que solo ejecutan las funciones de Lambda especificadas en el AppSpec archivo. Los eventos de ciclo de vida de la implementación de la siguiente tabla se indican en el orden en el que se ejecutan. Para obtener más información, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#).

Evento del ciclo de vida	Acción del evento del ciclo de vida
BeforeInstall (un enlace para funciones de Lambda)	Ejecutar las funciones de Lambda.
Instalación	Configurar el conjunto de tareas de sustitución.
AfterInstall (un enlace para funciones de Lambda)	Ejecutar las funciones de Lambda.
AllowTestTraffic	Dirigir el tráfico del oyente de prueba al grupo de destino 2.
AfterAllowTestTraffic (un enlace para funciones de Lambda)	Ejecutar las funciones de Lambda.

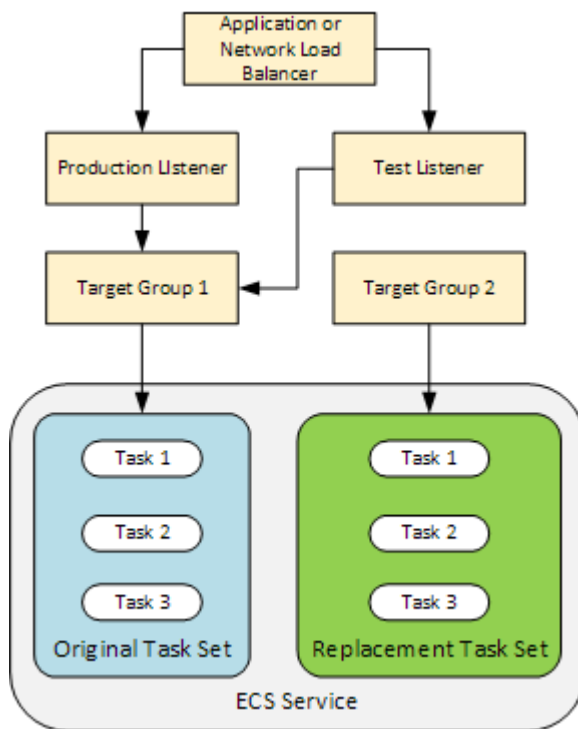
Evento del ciclo de vida	Acción del evento del ciclo de vida
<code>BeforeAllowTraffic</code> (un enlace para funciones de Lambda)	Ejecutar las funciones de Lambda.
<code>AllowTraffic</code>	Dirigir el tráfico desde el oyente de producción al grupo de destino 2.
<code>AfterAllowTraffic</code>	Ejecutar las funciones de Lambda.

Note

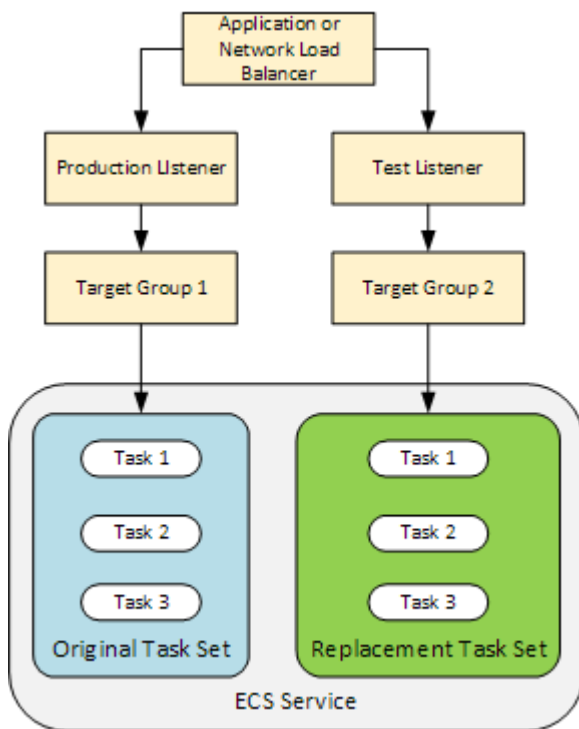
Las funciones de Lambda de un enlace son opcionales.

1. Ejecute cualquier función Lambda especificada en el `BeforeInstall` enlace del AppSpec archivo.
2. Durante el evento de ciclo de vida `Install`:
 - a. Se crea un conjunto de tareas de sustitución en el servicio de Amazon ECS.
 - b. La aplicación en contenedor actualizada se instala en el conjunto de tareas de sustitución.
 - c. El segundo grupo de destino se asocia al conjunto de tareas de sustitución.

En el siguiente diagrama se muestran los componentes de implementación con el nuevo conjunto de tareas de sustitución. La aplicación en contenedor está dentro del conjunto de tareas. El conjunto de tareas se compone de tres tareas. (Una aplicación puede tener cualquier número de tareas). El segundo grupo de destino se asocia ahora al conjunto de tareas de sustitución.



3. Ejecute cualquier función Lambda especificada en el `AfterInstall` enlace del AppSpec archivo.
4. Se invoca el evento `AllowTestTraffic`. Durante este evento del ciclo de vida, el agente de escucha de prueba dirige el tráfico a la aplicación en contenedor actualizada.



5.

Ejecute cualquier función Lambda especificada en el `AfterAllowTestTraffic` enlace del AppSpec archivo. Las funciones de Lambda pueden validar la implementación mediante el tráfico de prueba. Por ejemplo, una función Lambda puede servir tráfico al agente de escucha de prueba y realizar un seguimiento de las métricas del conjunto de tareas de sustitución. Si las reversiones están configuradas, puede configurar una CloudWatch alarma que active una reversión cuando falle la prueba de validación de la función Lambda.

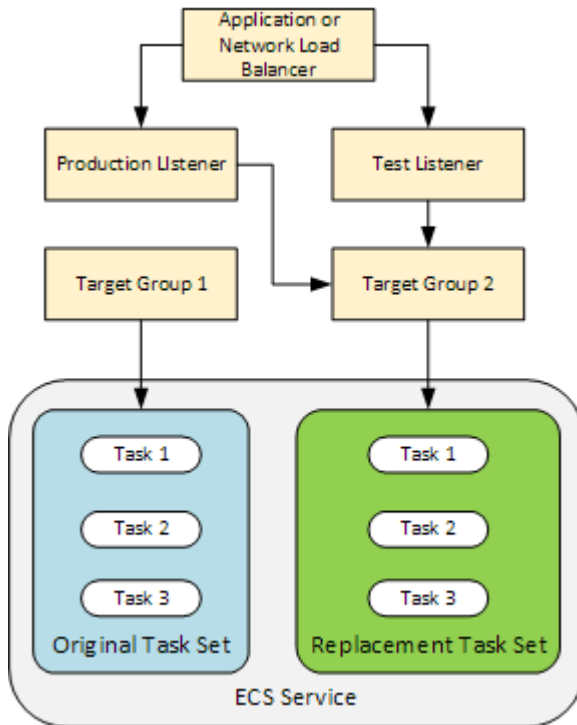
Una vez realizadas las pruebas de validación, se produce una de las situaciones siguientes:

- Si no se supera la validación y se han configurado las restauraciones, el estado de la implementación se marca como `Failed` y los componentes vuelven al estado que tenían cuando se inició la implementación.
- Si no se supera la validación y no se han configurado las restauraciones, el estado de la implementación se marca como `Failed` y los componentes permanecen en su estado actual.
- Si se supera la validación, la implementación continúa hasta el enlace `BeforeAllowTraffic`.

Para obtener más información, consulte [Supervisión de las implementaciones con CloudWatch alarmas en CodeDeploy](#), [Restauraciones automáticas](#) y [Configuración de las opciones avanzadas de un grupo de implementación](#).

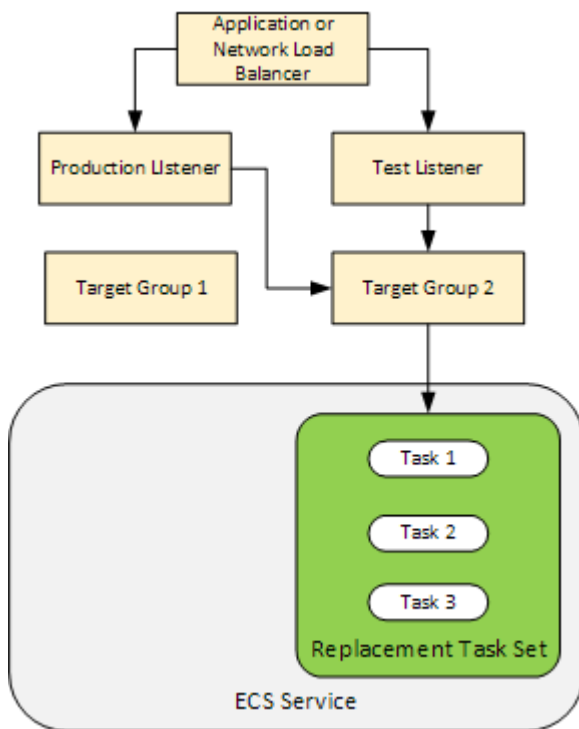
6. Ejecute cualquier función Lambda especificada en el `BeforeAllowTraffic` enlace del AppSpec archivo.

7. Se invoca el evento `AllowTraffic`. El tráfico se desvía del conjunto de tareas original al conjunto de tareas de sustitución. En el siguiente diagrama se muestra el conjunto de tareas de sustitución recibiendo tráfico de producción.



8. Ejecute cualquier función Lambda especificada en el `AfterAllowTraffic` enlace del AppSpec archivo.

9. Después de que todos los eventos se ejecuten correctamente, el estado de la implementación se establece en `Succeeded` y el conjunto de tareas original se elimina.



Carga de la revisión de la aplicación

Coloque un AppSpec archivo en Amazon S3 o introdúzcalo directamente en la consola o AWS CLI. Para obtener más información, consulte [Application Specification Files](#).

Creación de la aplicación y los grupos de implementación

Un grupo de CodeDeploy implementación en una plataforma informática de Amazon ECS identifica a los oyentes que sirven el tráfico a la aplicación Amazon ECS actualizada y a los dos grupos de destino utilizados durante la implementación. Un grupo de implementación también define un conjunto de opciones de configuración, como alarmas y configuraciones de restauración.

Implementación de la revisión de la aplicación

Ahora está listo para implementar el servicio de Amazon ECS actualizado especificado en su grupo de implementación. Puede usar la CodeDeploy consola o el comando [create-deployment](#). Hay parámetros que puede especificar para controlar la implementación, que incluyen el grupo de revisión y de implementaciones.

Actualización de la aplicación

Puede realizar actualizaciones en la aplicación y, a continuación, utilizar la CodeDeploy consola o ejecutar el comando [create-deployment](#) para realizar una revisión.

Implementaciones detenidas y que producen error

Puede usar la CodeDeploy consola o el comando [stop-deployment](#) para detener una implementación. Cuando se intenta detener la implementación, sucede una de estas tres cosas:

- La implementación se detiene y la operación devuelve el estado “realizada correctamente”. En este caso, no se ejecutan más eventos del ciclo de vida de implementación en el grupo de implementaciones para la implementación detenida.
- La implementación no se detiene inmediatamente y la operación devuelve el estado “pendiente”. En este caso, podrían seguir ejecutándose algunos eventos del ciclo de vida de implementación en el grupo de implementaciones. Después de finalizada la operación pendiente, las llamadas posteriores para detener la implementación devuelven el estado “realizada correctamente”.
- La implementación no se puede detener y la operación devuelve un error. Para obtener más información, consulta la [información sobre errores](#) y la referencia sobre [errores comunes](#) en la AWS CodeDeploy API.

Reimplementaciones y restauración de implementaciones

CodeDeploy implementa las reversiones redireccionando el tráfico del conjunto de tareas de reemplazo al conjunto de tareas original.

Puede configurar un grupo de implementación para que se restaure automáticamente si se dan ciertas condiciones, por ejemplo, si falla una implementación o si se alcanza el umbral de monitorización de alarmas. Además, puede anular la configuración de restauración especificada para el grupo de implementaciones en una implementación individual.

También puede optar por restaurar una implementación que produjo error manualmente, volviendo a implementar una revisión implementada anteriormente.

En todos los casos, a la implementación nueva o restaurada se le asigna su propio ID de implementación. La CodeDeploy consola muestra una lista de las implementaciones que son el resultado de una implementación automática.

Si vuelve a realizar una implementación, el grupo de destino asociado al conjunto de tareas original de la implementación actual se asocia al siguiente conjunto de tareas de sustitución de la nueva implementación.

Para obtener más información, consulte [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

Implementaciones azul/verde de Amazon ECS a través de AWS CloudFormation

Puede utilizarlo AWS CloudFormation para gestionar las implementaciones azul/verde de Amazon ECS mediante CodeDeploy. Para obtener más información, consulte [Cree una implementación azul/verde de Amazon ECS mediante AWS CloudFormation](#).

Note

La administración de las implementaciones azul/verde de Amazon ECS con no AWS CloudFormation está disponible en la región de Asia Pacífico (Osaka).

Implementación azul/verde en una plataforma de informática de EC2/en las instalaciones

En este tema se proporciona información sobre los componentes y el flujo de trabajo de las CodeDeploy implementaciones que utilizan la plataforma informática local EC2. Para obtener información sobre las implementaciones azul/verde, consulte [Información general de la implementación azul/verde](#).

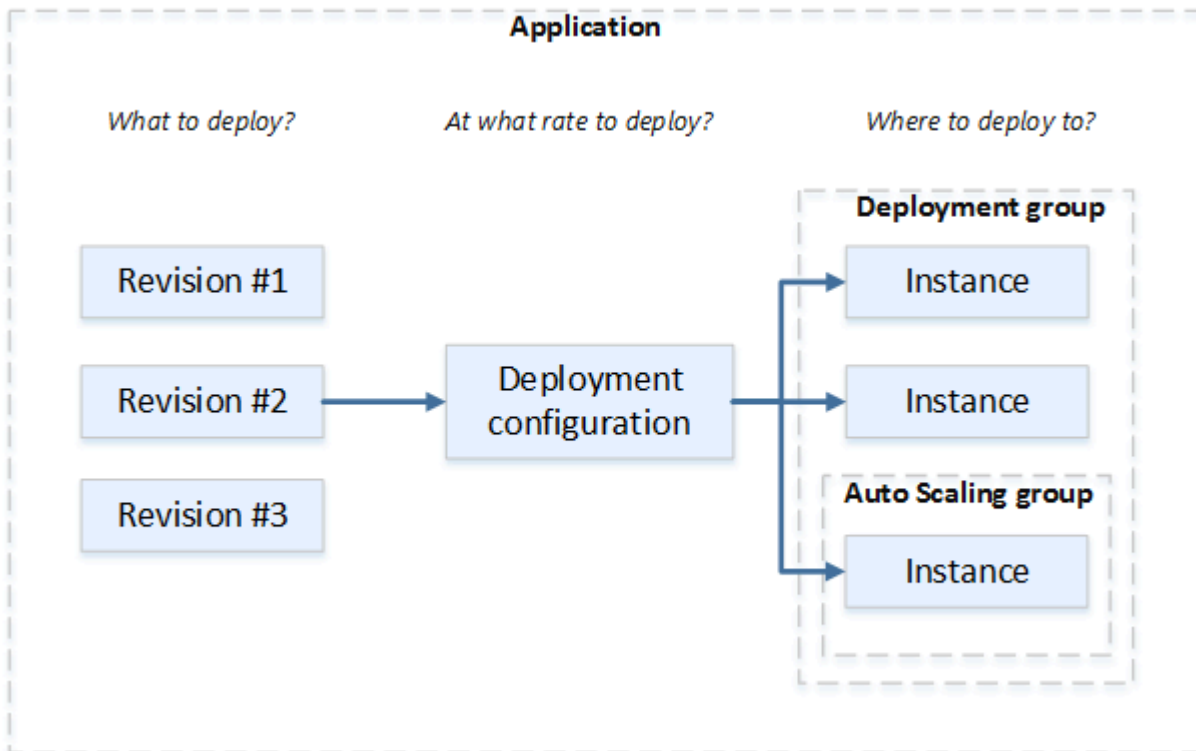
Temas

- [Componentes de una implementación en una plataforma de informática EC2/en las instalaciones](#)
- [Flujo de trabajo de implementación en una plataforma de informática de EC2/en las instalaciones](#)
- [Configuración de instancias](#)
- [Carga de la revisión de la aplicación](#)
- [Creación de la aplicación y los grupos de implementación](#)
- [Implementación de la revisión de la aplicación](#)
- [Actualización de la aplicación](#)

- [Implementaciones detenidas y que producen error](#)
- [Reimplementaciones y restauración de implementaciones](#)

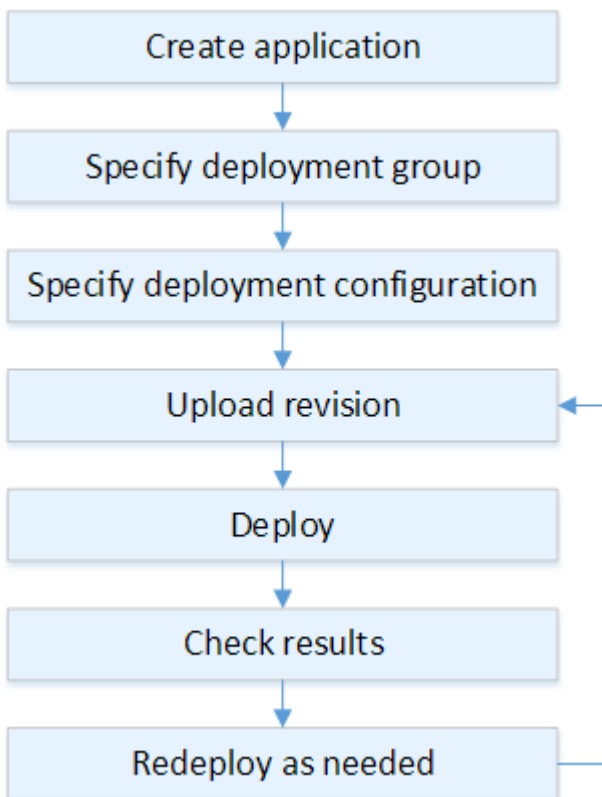
Componentes de una implementación en una plataforma de informática EC2/en las instalaciones

El siguiente diagrama muestra los componentes de una CodeDeploy implementación en una plataforma informática local de EC2.



Flujo de trabajo de implementación en una plataforma de informática de EC2/en las instalaciones

En el diagrama siguiente se muestran los principales pasos en la implementación de revisiones de aplicaciones:



Estos pasos incluyen:

1. Cree una aplicación y asígnele un nombre que identifique de forma exclusiva las revisiones de la aplicación que desea implementar y la plataforma informática de la aplicación. CodeDeploy usa este nombre durante una implementación para asegurarse de que hace referencia a los componentes de implementación correctos, como el grupo de implementación, la configuración de la implementación y la revisión de la aplicación. Para obtener más información, consulte [Cree una aplicación con CodeDeploy](#).
2. Configure un grupo de implementaciones especificando un tipo de implementación y las instancias en las que desea implementar las revisiones de la aplicación. Una implementación in situ actualiza instancias con la última revisión de la aplicación. Una implementación "blue/green" registra un conjunto de instancias de sustitución para el grupo de implementaciones con un balanceador de carga y cancela el registro de las instancias originales.

Puede especificar las etiquetas aplicadas a las instancias, los nombres de grupos de Amazon EC2 Auto Scaling o ambas cosas.

Si especificas un grupo de etiquetas en un grupo de despliegues, se CodeDeploy despliega en instancias que tengan aplicada al menos una de las etiquetas especificadas. Si especificas dos

o más grupos de etiquetas, se CodeDeploy despliega solo en las instancias que cumplen los criterios de cada uno de los grupos de etiquetas. Para obtener más información, consulte [Tagging Instances for Deployments](#).

En todos los casos, las instancias deben configurarse para usarse en una implementación (es decir, deben estar etiquetadas o pertenecer a un grupo de Auto Scaling de Amazon EC2) y tener el CodeDeploy agente instalado y en ejecución.

Le proporcionamos una AWS CloudFormation plantilla que puede utilizar para configurar rápidamente una instancia de Amazon EC2 basada en Amazon Linux o Windows Server. También le proporcionamos el CodeDeploy agente independiente para que pueda instalarlo en instancias de Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) o Windows Server. Para obtener más información, consulte [Cree un grupo de despliegue con CodeDeploy](#).

También puede especificar las siguientes opciones:

- Notificaciones de Amazon SNS. Cree activadores que envíen notificaciones a los suscriptores de un tema de Amazon SNS cuando ocurran eventos específicos, como eventos de éxito o error, en las implementaciones e instancias. Para obtener más información, consulte [Monitoring Deployments with Amazon SNS Event Notifications](#).
 - Administración de implementaciones basadas en alarmas. Implemente el monitoreo de CloudWatch alarmas de Amazon para detener las implementaciones cuando sus métricas superen o caigan por debajo de los umbrales establecidos. CloudWatch
 - Restauraciones de implementaciones automáticas. Configure una implementación para que se restaure automáticamente a la última revisión correcta conocida si se produce un error en una implementación o si se supera el umbral de la alarma.
3. Especifique una configuración de implementación para indicar en cuántas instancias se deben implementar simultáneamente las revisiones de la aplicación y describir las condiciones de éxito y fracaso de la implementación. Para obtener más información, consulte [View Deployment Configuration Details](#).
 4. Cargue una revisión de la aplicación en Amazon S3 o GitHub. Además de los archivos que desee implementar y de los scripts que desee ejecutar durante la implementación, debe incluir un archivo de especificaciones de la aplicación (AppSpec archivo). Este archivo contiene las instrucciones de implementación, como, por ejemplo, dónde se deben copiar los archivos en cada instancia y cuándo se deben ejecutar los scripts de implementación. Para obtener más información, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).

5. Implemente la revisión de la aplicación en el grupo de implementaciones. El CodeDeploy agente de cada instancia del grupo de implementación copia la revisión de la aplicación de Amazon S3 o GitHub a la instancia. A continuación, el CodeDeploy agente separa la revisión y, utilizando el AppSpec archivo, copia los archivos en las ubicaciones especificadas y ejecuta los scripts de implementación. Para obtener más información, consulte [Cree una implementación con CodeDeploy](#).
6. Compruebe los resultados de la implementación. Para obtener más información, consulte [Supervisión de los despliegues en CodeDeploy](#).
7. Vuelva a implementar una revisión. Se recomienda hacer esto si se necesita corregir un error en el contenido de origen o ejecutar los scripts de la implementación en un orden diferente o solucionar una implementación que dio error. Para ello, reagrupe el contenido fuente revisado, los scripts de despliegue y el AppSpec archivo en una nueva revisión y, a continuación, cargue la revisión en el GitHub depósito o repositorio de Amazon S3. Luego ejecute una nueva implementación en el mismo grupo de implementaciones con la nueva revisión. Para obtener más información, consulte [Cree una implementación con CodeDeploy](#).

Configuración de instancias

Debe configurar instancias antes de implementar revisiones en la aplicación por primera vez. Si la revisión de una aplicación requiere tres servidores de producción y dos servidores de copia de seguridad, lanzará o utilizará cinco instancias.

Para aprovisionar instancias manualmente:

1. Instale el CodeDeploy agente en las instancias. El CodeDeploy agente se puede instalar en instancias de Amazon Linux, Ubuntu Server, RHEL y Windows Server.
2. Habilite el etiquetado si está utilizando etiquetas para identificar instancias en un grupo de implementación. CodeDeploy se basa en etiquetas para identificar y agrupar las instancias en grupos CodeDeploy de implementación. Si bien los tutoriales de Introducción utilizaron ambos, puede usar una clave o un valor para definir una etiqueta para un grupo de implementaciones.
3. Inicie instancias de Amazon EC2 con un perfil de instancia de IAM adjunto. El perfil de la instancia de IAM debe adjuntarse a una instancia de Amazon EC2 a medida que se lanza para que CodeDeploy el agente verifique la identidad de la instancia.
4. Cree un rol de servicio. Proporcione acceso al servicio para CodeDeploy poder ampliar las etiquetas de su AWS cuenta.

Para una implementación inicial, la AWS CloudFormation plantilla hace todo esto por ti. Crea y configura nuevas instancias únicas de Amazon EC2 basadas en Amazon Linux o Windows Server con CodeDeploy el agente ya instalado. Para obtener más información, consulte [Trabajar con instancias para CodeDeploy](#).

Note

Para una implementación azul/verde, puede elegir entre usar las instancias que ya tiene para el entorno de reemplazo o dejar que se le CodeDeploy aprovisionen nuevas instancias como parte del proceso de implementación.

Carga de la revisión de la aplicación

Coloca un AppSpec archivo en la carpeta raíz de la estructura de carpetas de contenido fuente de la aplicación. Para obtener más información, consulte [Application Specification Files](#).

Empaquete la estructura de carpetas del contenido de origen de la aplicación en un formato de archivo de almacenamiento como zip, tar o tar comprimido. Suba el archivo de almacenamiento (la revisión) a un depósito o GitHub repositorio de Amazon S3.

Note

Los formatos de archivo tar y tar comprimido (.tar y .tar.gz) no son compatibles con las instancias de Windows Server.

Creación de la aplicación y los grupos de implementación

Un grupo de CodeDeploy implementación identifica un conjunto de instancias en función de sus etiquetas, los nombres de los grupos de Auto Scaling de Amazon EC2 o ambos. Se pueden implementar varias revisiones de la aplicación en la misma instancia. Una revisión de la aplicación se puede implementar en múltiples instancias.

Por ejemplo, podría agregar una etiqueta "Prod" a los tres servidores de producción y "Backup" a los dos servidores de backup. Estas dos etiquetas se pueden usar para crear dos grupos de implementación diferentes en la CodeDeploy aplicación, lo que le permite elegir qué conjunto de servidores (o ambos) debe participar en una implementación.

Puede utilizar varios grupos de etiquetas en un grupo de implementaciones para restringir las implementaciones con un conjunto más pequeño de instancias. Para obtener más información, consulte [Tagging Instances for Deployments](#).

Implementación de la revisión de la aplicación

Ahora está listo para implementar la revisión de la aplicación desde Amazon S3 o GitHub en el grupo de implementación. Puede usar la CodeDeploy consola o el comando [create-deployment](#). Hay parámetros que puede especificar para controlar la implementación, que incluyen la revisión, el grupo de implementaciones y la configuración de la implementación.

Actualización de la aplicación

Puede realizar actualizaciones en la aplicación y, a continuación, utilizar la CodeDeploy consola o ejecutar el comando [create-deployment](#) para realizar una revisión.

Implementaciones detenidas y que producen error

Puede usar la CodeDeploy consola o el comando [stop-deployment](#) para detener una implementación. Cuando se intenta detener la implementación, sucede una de estas tres cosas:

- La implementación se detiene y la operación devuelve el estado “realizada correctamente”. En este caso, no se ejecutan más eventos del ciclo de vida de implementación en el grupo de implementaciones para la implementación detenida. Es posible que algunos archivos se hayan copiado y que algunos scripts ya se hayan ejecutado en una o varias de las instancias del grupo de implementaciones.
- La implementación no se detiene inmediatamente y la operación devuelve el estado “pendiente”. En este caso, podrían seguir ejecutándose algunos eventos del ciclo de vida de implementación en el grupo de implementaciones. Es posible que algunos archivos se hayan copiado y que algunos scripts ya se hayan ejecutado en una o varias de las instancias del grupo de implementaciones. Después de finalizada la operación pendiente, las llamadas posteriores para detener la implementación devuelven el estado “realizada correctamente”.
- La implementación no se puede detener y la operación devuelve un error. Para obtener más información, consulta la [ErrorInformation](#) referencia sobre [errores comunes](#) en la AWS CodeDeploy API.

Al igual que las implementaciones detenidas, las que dan error pueden dar lugar a que algunos eventos del ciclo de vida de implementación ya se hayan ejecutado en una o varias de las instancias

del grupo de implementaciones. Para averiguar por qué ha fallado una implementación, puede usar la CodeDeploy consola, [get-deployment-instance](#) ejecutar el comando o analizar los datos del archivo de registro de la implementación fallida. Para obtener más información, consulte [Limpieza de revisiones de aplicación y archivos de registro](#) y [Vea los datos de registro de las implementaciones locales CodeDeploy de EC2/](#).

Reimplementaciones y restauración de implementaciones

CodeDeploy implementa las reversiones al volver a implementar, como una nueva implementación, una revisión previamente implementada.

Puede configurar un grupo de implementación para que se restaure automáticamente si se dan ciertas condiciones, por ejemplo, si falla una implementación o si se alcanza el umbral de monitorización de alarmas. Además, puede anular la configuración de restauración especificada para el grupo de implementaciones en una implementación individual.

También puede optar por restaurar una implementación que produjo error manualmente, volviendo a implementar una revisión implementada anteriormente.

En todos los casos, a la implementación nueva o restaurada se le asigna su propio ID de implementación. La lista de despliegues que puede ver en la CodeDeploy consola muestra cuáles son el resultado de un despliegue automático.

Para obtener más información, consulte [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

CodeDeploy archivos de especificación de la aplicación (AppSpec)

[Un archivo de especificaciones de la aplicación \(AppSpec archivo\), que es exclusivo de CodeDeploy, es un archivo con formato YAML o con formato JSON](#). El AppSpec archivo se usa para administrar cada implementación como una serie de enlaces de eventos del ciclo de vida, que se definen en el archivo.

Para obtener información sobre cómo crear un AppSpec archivo bien formado, consulte [CodeDeploy AppSpec referencia de archivo](#).

Temas

- [AppSpec archivos en una plataforma informática Amazon ECS](#)
- [AppSpec archivos en una plataforma AWS Lambda informática](#)

- [AppSpec archivos en una plataforma informática local EC2/](#)
- [Cómo usa el CodeDeploy agente el AppSpec archivo](#)

AppSpec archivos en una plataforma informática Amazon ECS

Si su aplicación utiliza la plataforma informática Amazon ECS, el AppSpec archivo se puede formatear con YAML o JSON. También se puede escribir directamente en un editor en la consola. El AppSpec archivo se utiliza para especificar:

- El nombre del servicio de Amazon ECS y el nombre del contenedor y el puerto utilizado para dirigir el tráfico hacia el nuevo conjunto de tareas.
- Las funciones que se van a usar como pruebas de validación.

Puede ejecutar funciones Lambda de validación después de los eventos del ciclo de vida de implementación. Para obtener más información, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#), [AppSpec estructura de archivos para las implementaciones de Amazon ECS](#) y [AppSpec Ejemplo de archivo para una implementación de Amazon ECS](#).

AppSpec archivos en una plataforma AWS Lambda informática

Si la aplicación utiliza la plataforma de procesamiento AWS Lambda, el AppSpec archivo se puede formatear con YAML o JSON. También se puede escribir directamente en un editor en la consola. El AppSpec archivo se utiliza para especificar:

- La versión de la AWS Lambda función que se va a implementar.
- Las funciones que se van a usar como pruebas de validación.

Puede ejecutar funciones Lambda de validación después de los eventos del ciclo de vida de implementación. Para obtener más información, consulte [AppSpec sección de «ganchos» para una implementación de AWS Lambda](#).

AppSpec archivos en una plataforma informática local EC2/

Si su aplicación utiliza la plataforma informática local EC2, el archivo siempre tiene formato YAML. AppSpec El archivo se utiliza para:

- Asignar los archivos de origen de la revisión de la aplicación a sus destinos en la instancia.

- Especificar permisos personalizados para los archivos implementados.
- Especificar los scripts que se van a ejecutar en cada instancia en las distintas fases del proceso de implementación.

Puede ejecutar scripts en una instancia después de muchos de los eventos individuales del ciclo de vida de la implementación. CodeDeploy ejecuta solo los scripts especificados en el archivo, pero esos scripts pueden llamar a otros scripts de la instancia. Puede ejecutar cualquier tipo de script, siempre y cuando sea compatible con el sistema operativo que se ejecuta en las instancias. Para obtener más información, consulte [AppSpec Sección de «enganches» para una implementación local de EC2/](#).

Cómo usa el CodeDeploy agente el AppSpec archivo

Durante el despliegue, el CodeDeploy agente busca el nombre del evento actual en la sección de enlaces del AppSpec archivo. Si no se encuentra el evento, el CodeDeploy agente pasa al siguiente paso. Si se encuentra el evento, el CodeDeploy agente recupera la lista de scripts que se van a ejecutar. Los scripts se ejecutan de forma secuencial, en el orden en que aparecen en el archivo. El estado de cada script se registra en el archivo de registro del CodeDeploy agente de la instancia.

Si un script se ejecuta correctamente, devuelve un código de salida de 0 (cero).

Note

El CodeDeploy agente no se utiliza en una implementación de AWS Lambda o Amazon ECS.

Durante el evento de instalación, el CodeDeploy agente usa las asignaciones definidas en la sección de archivos del AppSpec archivo para determinar qué carpetas o archivos copiar de la revisión a la instancia.

Si el CodeDeploy agente instalado en el sistema operativo no coincide con lo que aparece en el AppSpec archivo, se produce un error en la implementación.

Para obtener información sobre los archivos de registro del CodeDeploy agente, consulte [Trabajando con el CodeDeploy agente](#).

Empezar con CodeDeploy

Temas

- [Paso 1: Configurar](#)
- [Paso 2: Crear un rol de servicio para CodeDeploy](#)
- [Paso 3: Limitar los permisos del CodeDeploy usuario](#)
- [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#)

Paso 1: Configurar

Antes de usarlo AWS CodeDeploy por primera vez, debe completar los pasos de configuración. Los pasos incluyen la creación de una AWS cuenta (si aún no tiene una) y un usuario administrativo con acceso programático.

En esta guía, el usuario administrativo se denomina usuario CodeDeploy administrativo.

Inscríbase en un 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 [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 .

Ahora ha creado e iniciado sesión como usuario CodeDeploy administrativo.

Concesión de acceso programático

Los usuarios necesitan acceso programático si quieren interactuar con personas AWS ajenas a AWS Management Console. La forma de conceder el acceso programático depende del tipo de usuario que acceda. AWS

Para conceder acceso programático a los usuarios, elija una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal (Usuarios administrados en el IAM Identity Center)	Usa credenciales temporales para firmar las solicitudes programáticas a los AWS CLI, AWS SDK o las API. AWS	Siga las instrucciones de la interfaz que desea utilizar: <ul style="list-style-type: none"> • Para ello AWS CLI, consulte Configuración del uso AWS IAM Identity Center en AWS CLI la Guía del AWS Command Line Interface usuario. • Para obtener AWS información sobre los SDK, las herramientas y AWS las

¿Qué usuario necesita acceso programático?	Para	Mediante
		<p>API, consulte la autenticación del IAM Identity Center en la Guía de referencia de AWS los SDK y las herramientas.</p>
IAM	<p>Utilice credenciales temporales para firmar las solicitudes programáticas a los AWS SDK o las AWS CLI API. AWS</p>	<p>Siga las instrucciones de Uso de credenciales temporales con AWS recursos de la Guía del usuario de IAM.</p>
IAM	<p>(No recomendado) Utilice credenciales de larga duración para firmar las solicitudes programáticas a los AWS CLI AWS SDK o las API. AWS</p>	<p>Siga las instrucciones de la interfaz que desea utilizar:</p> <ul style="list-style-type: none"> • Para ello AWS CLI, consulte Autenticación con credenciales de usuario de IAM en la Guía del usuario.AWS Command Line Interface • Para obtener información AWS sobre los SDK y las herramientas, consulte Autenticarse con credenciales de larga duración en la Guía de referencia de los AWS SDK y las herramientas. • Para obtener información AWS sobre las API, consulte Administrar las claves de acceso para los usuarios de IAM en la Guía del usuario de IAM.

⚠ Important

Le recomendamos encarecidamente que configure el usuario CodeDeploy administrativo como una identidad de personal (un usuario gestionado en el Centro de identidades de IAM) con el. AWS CLI Muchos de los procedimientos de esta guía dan por sentado que se utiliza AWS CLI para realizar las configuraciones.

⚠ Important

Si configura el AWS CLI, es posible que se le pida que especifique una AWS región. Especifique una de las regiones admitidas que se encuentran en la lista de [Regiones y puntos de conexión](#) de la Referencia general de AWS.

Paso 2: Crear un rol de servicio para CodeDeploy

En AWS, las funciones de servicio se utilizan para conceder permisos a un AWS servicio para que pueda acceder a AWS los recursos. Las políticas que adjunte al rol de servicio determinarán a qué recursos de tendrá acceso el servicio y qué puede hacer con esos recursos.

La función de servicio para la que cree CodeDeploy debe tener los permisos necesarios para su plataforma informática. Si implementa más de una plataforma de informática, cree un rol de servicio para cada una. Para añadir permisos, adjunta una o más de las siguientes políticas AWS proporcionadas:

Para las implementaciones de EC2/en las instalaciones, adjunte la política **AWSCodeDeployRole**. Proporciona los permisos necesarios para que su rol de servicio:

- Leer las etiquetas en sus instancias o identificar sus instancias de Amazon EC2 mediante nombres de grupo de Amazon EC2 Auto Scaling.
- Leer, crear, actualizar y eliminar grupos de Amazon EC2 Auto Scaling, enlaces de ciclo de vida y políticas de escalado.
- Publicar información para temas de Amazon SNS.
- Recupere información sobre CloudWatch las alarmas.
- Leer y actualizar Elastic Load Balancing.

Note

Si crea un grupo de escalado automático con una plantilla de inicio, debe añadir los siguientes permisos:

- `ec2:RunInstances`
- `ec2:CreateTags`
- `iam:PassRole`

Para obtener más información, consulte [Paso 2: Crear un rol de servicio](#), [Creación de una plantilla de lanzamiento para un grupo de escalado automático](#) y [Compatibilidad con las plantillas de lanzamiento](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

En las implementaciones de Amazon ECS, si desea obtener acceso total a los servicios de soporte técnico, asocie la política **AWSCodeDeployRoleForECS**. Proporciona los permisos necesarios para que su rol de servicio:

- Leer, actualizar y eliminar conjuntos de tareas de Amazon ECS.
- Actualizar grupos de destino de Elastic Load Balancing, oyentes y reglas.
- Invoque AWS Lambda funciones.
- Acceder a archivos de revisión en buckets de Amazon S3.
- Recupere información sobre CloudWatch las alarmas.
- Publicar información para temas de Amazon SNS.

En las implementaciones de Amazon ECS, si desea obtener acceso limitado a los servicios de soporte técnico, asocie la política **AWSCodeDeployRoleForECSLimited**. Proporciona los permisos necesarios para que su rol de servicio:

- Leer, actualizar y eliminar conjuntos de tareas de Amazon ECS.
- Recupera información sobre CloudWatch las alarmas.
- Publicar información para temas de Amazon SNS.

Para las implementaciones de AWS Lambda, si desea permitir la publicación en Amazon SNS, adjunte la política **AWSCodeDeployRoleForLambda**. Proporciona los permisos necesarios para que su rol de servicio:

- Lea, actualice e invoque funciones y alias AWS Lambda .
- Acceder a archivos de revisión en buckets de Amazon S3.
- Recupere información sobre CloudWatch las alarmas.
- Publicar información para temas de Amazon SNS.

Para las implementaciones de AWS Lambda, si desea limitar el acceso a Amazon SNS, adjunte la política. **AWSCodeDeployRoleForLambdaLimited** Proporciona los permisos necesarios para que su rol de servicio:

- Lea, actualice e invoque funciones y alias AWS Lambda .
- Acceder a archivos de revisión en buckets de Amazon S3.
- Recupere información sobre CloudWatch las alarmas.

Como parte de la configuración del rol de servicio, también actualizará su relación de confianza para especificar a qué puntos de enlace desea concederle acceso.

Puede crear un rol de servicio con la consola de IAM, las API de IAM o las AWS CLI API de IAM.

Temas

- [Creación de un rol de servicio \(consola\)](#)
- [Creación de un rol de servicio \(CLI\)](#)
- [Obtención del ARN del rol de servicio \(consola\)](#)
- [Obtención del ARN del rol de servicio \(CLI\)](#)

Creación de un rol de servicio (consola)

1. [Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.](https://console.aws.amazon.com/iam/)
2. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
3. Elija el AWS servicio y, en Caso de uso, elija CodeDeploy en la lista desplegable.
4. Elija su caso de uso:
 - Para las implementaciones de EC2/on-premise, elija. CodeDeploy
 - Para las implementaciones de AWS Lambda, elija Lambda. CodeDeploy

- Para las implementaciones de Amazon ECS, elija CodeDeploy - ECS.
5. Elija Siguiente.
 6. En la página Añadir permisos, se muestra la política de permisos correcta para cada caso de uso. Elija Siguiente.
 7. En la página Asignar nombre, revisar y crear, en Nombre del rol, ingrese un nombre para el rol de servicio (por ejemplo, **CodeDeployServiceRole**) y luego elija Crear rol.

También puede ingresar una descripción para este rol de servicio en el cuadro Descripción del rol.

8. Si desea que este rol de servicio tenga permiso de acceso a todos los puntos de enlace contemplados actualmente, ya ha terminado con este procedimiento.

Para restringir el acceso de este rol de servicio a algunos puntos de conexión, continúe con los pasos restantes de este procedimiento.

9. En la lista de roles, busque y seleccione el rol que acaba de crear (CodeDeployServiceRole).
10. Seleccione la pestaña Relaciones de confianza.
11. Elija Editar la política de confianza.

Debe aparecer la siguiente política, que concede al rol de servicio el permiso de acceso a todos los puntos de enlace compatibles:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codedeploy.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para conceder acceso al rol de servicio a solo algunos puntos de conexión admitidos, reemplace el contenido del recuadro de texto de la política con la política siguiente. Quite las líneas de los puntos de conexión a los que desea evitar el acceso y, a continuación, elija Actualizar política.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codedeploy.us-east-1.amazonaws.com",
          "codedeploy.us-east-2.amazonaws.com",
          "codedeploy.us-west-1.amazonaws.com",
          "codedeploy.us-west-2.amazonaws.com",
          "codedeploy.ca-central-1.amazonaws.com",
          "codedeploy.ap-east-1.amazonaws.com",
          "codedeploy.ap-northeast-1.amazonaws.com",
          "codedeploy.ap-northeast-2.amazonaws.com",
          "codedeploy.ap-northeast-3.amazonaws.com",
          "codedeploy.ap-southeast-1.amazonaws.com",
          "codedeploy.ap-southeast-2.amazonaws.com",
          "codedeploy.ap-southeast-3.amazonaws.com",
          "codedeploy.ap-southeast-4.amazonaws.com",
          "codedeploy.ap-south-1.amazonaws.com",
          "codedeploy.ap-south-2.amazonaws.com",
          "codedeploy.ca-central-1.amazonaws.com",
          "codedeploy.eu-west-1.amazonaws.com",
          "codedeploy.eu-west-2.amazonaws.com",
          "codedeploy.eu-west-3.amazonaws.com",
          "codedeploy.eu-central-1.amazonaws.com",
          "codedeploy.eu-central-2.amazonaws.com",
          "codedeploy.eu-north-1.amazonaws.com",
          "codedeploy.eu-south-1.amazonaws.com",
          "codedeploy.eu-south-2.amazonaws.com",
          "codedeploy.il-central-1.amazonaws.com",
          "codedeploy.me-central-1.amazonaws.com",
          "codedeploy.me-south-1.amazonaws.com",
          "codedeploy.sa-east-1.amazonaws.com"
        ]
      }
    }
  ]
}
```

```
    },
    "Action": "sts:AssumeRole"
  }
]
}
```

Para obtener más información sobre la creación de funciones de servicio, consulte [Creación de una función para delegar permisos a un AWS servicio](#) en la Guía del usuario de IAM.

Creación de un rol de servicio (CLI)

1. En el equipo de implementación, crea un archivo de texto con un nombre, como por ejemplo, `CodeDeployDemo-Trust.json`. Este archivo se utiliza para poder CodeDeploy trabajar en su nombre.

Realice una de las siguientes acciones siguientes:

- Para permitir el acceso a todas AWS las regiones compatibles, guarde el siguiente contenido en el archivo:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codedeploy.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```


- Para conceder acceso solo a algunas regiones, escriba el siguiente contenido en el archivo y elimine las líneas de las regiones a las que no desee conceder acceso:

```
{
  "Version": "2012-10-17",
```

```
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": [

        "codedeploy.us-east-1.amazonaws.com",
        "codedeploy.us-east-2.amazonaws.com",
        "codedeploy.us-west-1.amazonaws.com",
        "codedeploy.us-west-2.amazonaws.com",
        "codedeploy.ca-central-1.amazonaws.com",
        "codedeploy.ap-east-1.amazonaws.com",
        "codedeploy.ap-northeast-1.amazonaws.com",
        "codedeploy.ap-northeast-2.amazonaws.com",
        "codedeploy.ap-northeast-3.amazonaws.com",
        "codedeploy.ap-southeast-1.amazonaws.com",
        "codedeploy.ap-southeast-2.amazonaws.com",
        "codedeploy.ap-southeast-3.amazonaws.com",
        "codedeploy.ap-southeast-4.amazonaws.com",
        "codedeploy.ap-south-1.amazonaws.com",
        "codedeploy.ap-south-2.amazonaws.com",
        "codedeploy.ca-central-1.amazonaws.com",
        "codedeploy.eu-west-1.amazonaws.com",
        "codedeploy.eu-west-2.amazonaws.com",
        "codedeploy.eu-west-3.amazonaws.com",
        "codedeploy.eu-central-1.amazonaws.com",
        "codedeploy.eu-central-2.amazonaws.com",
        "codedeploy.eu-north-1.amazonaws.com",
        "codedeploy.eu-south-1.amazonaws.com",
        "codedeploy.eu-south-2.amazonaws.com",
        "codedeploy.il-central-1.amazonaws.com",
        "codedeploy.me-central-1.amazonaws.com",
        "codedeploy.me-south-1.amazonaws.com",
        "codedeploy.sa-east-1.amazonaws.com"


      ]
    },
    "Action": "sts:AssumeRole"
  }
]
```

 Note

No incluya una coma después del último punto de enlace de la lista.

- Desde el mismo directorio, ejecute el comando `create-role` para crear un rol de servicio con el nombre **CodeDeployServiceRole** basado en la información del archivo de texto que acaba de crear:

```
aws iam create-role --role-name CodeDeployServiceRole --assume-role-policy-document
file://CodeDeployDemo-Trust.json
```

 Important

Asegúrese de incluir `file://` antes del nombre de archivo. Es obligatorio en este comando.

En la salida del comando, anote el valor de la entrada `Arn` para el objeto `Role`. Lo necesitará más adelante al crear grupos de implementaciones. Si no recuerda el valor, siga las instrucciones indicadas en [Obtención del ARN del rol de servicio \(CLI\)](#).

- La política administrada que utilice dependerá de la plataforma de informática.
 - Para una implementación en una plataforma de informática de EC2/en las instalaciones:

Llame al comando `attach-role-policy` para conceder al rol de servicio

CodeDeployServiceRole los permisos correspondientes a la política administrada de IAM denominada **AWSCodeDeployRole**. Por ejemplo:

```
aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSCodeDeployRole
```

- Si su implementación se realiza en una plataforma de procesamiento AWS Lambda:

Llame al comando `attach-role-policy` para conceder al rol de servicio

CodeDeployServiceRole los permisos correspondientes a la política administrada de IAM denominada **AWSCodeDeployRoleForLambda** o **AWSCodeDeployRoleForLambdaLimited**. Por ejemplo:

```
aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSCodeDeployRoleForLambda
```

- Si su implementación se realiza en una plataforma de informática de Amazon ECS:

Llame al comando `attach-role-policy` para conceder al rol de servicio

CodeDeployServiceRole los permisos correspondientes a la política administrada de IAM denominada **AWSCodeDeployRoleForECS** o **AWSCodeDeployRoleForECSLimited**. Por ejemplo:

```
aws iam attach-role-policy --role-name CodeDeployServiceRole --policy-arn
arn:aws:iam::aws:policy/AWSCodeDeployRoleForECS
```

Para obtener más información sobre la creación de funciones de servicio, consulte [Creación de una función para un AWS servicio](#) en la Guía del usuario de IAM.

Obtención del ARN del rol de servicio (consola)

Para obtener el ARN del rol de servicio con la consola de IAM:

1. [Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Seleccione Roles en el panel de navegación.
3. En el cuadro Filter (Filtro), escriba **CodeDeployServiceRole** y, a continuación, pulse Intro.
4. Elige CodeDeployServiceRole.
5. Anote el valor del campo Role ARN (ARN de rol).

Obtención del ARN del rol de servicio (CLI)

Para usar el AWS CLI para obtener el ARN del rol de servicio, llame al `get-role` comando contra el rol de servicio denominado: **CodeDeployServiceRole**

```
aws iam get-role --role-name CodeDeployServiceRole --query "Role.Arn" --output text
```

El valor devuelto es el ARN del rol de servicio.

Paso 3: Limitar los permisos del CodeDeploy usuario

Por motivos de seguridad, le recomendamos que limite los permisos del usuario administrativo que creó [Paso 1: Configurar](#) a solo los necesarios para crear y administrar las implementaciones en CodeDeploy él.

Utilice la siguiente serie de procedimientos para limitar los permisos del usuario CodeDeploy administrativo.

Antes de empezar

- Asegúrese de haber creado un usuario CodeDeploy administrativo en el Centro de identidades de IAM siguiendo las instrucciones que se indican en [Paso 1: Configurar](#).

Para crear un conjunto de permisos

Asignará este conjunto de permisos al usuario CodeDeploy administrativo más adelante.

1. Inicie sesión en la AWS IAM Identity Center consola AWS Management Console y ábrala en <https://console.aws.amazon.com/singlesignon/>.
2. En el panel de navegación, elija Conjuntos de permisos y, a continuación, elija Crear conjunto de permisos.
3. Elija Conjunto de permisos personalizado.
4. Elija Siguiente.
5. Elija Política insertada.
6. Elimine el código de muestra.
7. Añada el código de política siguiente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeDeployAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "autoscaling:*",
        "codedeploy:*",
        "ec2:*",
        "lambda:*",
```

```

    "ecs:*",
    "elasticloadbalancing:*",
    "iam:AddRoleToInstanceProfile",
    "iam:AttachRolePolicy",
    "iam:CreateInstanceProfile",
    "iam:CreateRole",
    "iam>DeleteInstanceProfile",
    "iam>DeleteRole",
    "iam>DeleteRolePolicy",
    "iam:GetInstanceProfile",
    "iam:GetRole",
    "iam:GetRolePolicy",
    "iam:ListInstanceProfilesForRole",
    "iam:ListRolePolicies",
    "iam:ListRoles",
    "iam:PutRolePolicy",
    "iam:RemoveRoleFromInstanceProfile",
    "s3:*",
    "ssm:*"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeDeployRolePolicy",
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": "arn:aws:iam::account-ID:role/CodeDeployServiceRole"
}
]
}

```

En esta política, sustituya *arn:aws:iam::account-ID:role/* por el valor ARN del rol de servicio en *CodeDeployServiceRole* el que creó. CodeDeploy [Paso 2: Crear un rol de servicio para CodeDeploy](#) Puede encontrar el valor del ARN en la página de detalles del rol de servicio en la consola de IAM.

La política anterior le permite implementar una aplicación en una plataforma de informática de AWS Lambda, una plataforma de informática de EC2/en las instalaciones y una plataforma de informática de Amazon ECS.

Puede utilizar las AWS CloudFormation plantillas que se proporcionan en esta documentación para lanzar instancias de Amazon EC2 que sean compatibles con CodeDeploy. Para usar AWS CloudFormation plantillas para crear aplicaciones, grupos de despliegues o configuraciones de despliegues, debe proporcionar acceso a AWS CloudFormation (y a AWS los servicios y acciones que AWS CloudFormation dependan de ellos) añadiendo el `cloudformation:*` permiso a la política de permisos del usuario CodeDeploy administrativo, de la siguiente manera:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        ...
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

8. Elija Siguiente.
9. En Nombre del conjunto de permisos, introduzca:

CodeDeployUserPermissionSet

10. Elija Siguiente.
11. En la página Revisión, revise la información y, a continuación, elija Crear grupo.

Para asignar el conjunto de permisos al usuario administrativo CodeDeploy


1. En el panel de navegación, elija y Cuentas de AWS, a continuación, active la casilla de verificación situada junto a la casilla en la Cuenta de AWS que ha iniciado sesión actualmente.
2. Seleccione el botón Asignar usuarios o grupos.
3. Elija la pestaña Usuarios.
4. Seleccione la casilla de verificación situada junto al usuario CodeDeploy administrativo.
5. Elija Siguiente.

6. Seleccione la casilla situada junto a `CodeDeployUserPermissionSet`.
7. Elija Siguiente.
8. Revise la información y seleccione Enviar.

Ahora ha asignado el usuario `CodeDeploy` administrativo y `CodeDeployUserPermissionSet` el suyo Cuenta de AWS, uniéndolos.

Para cerrar sesión y volver a iniciarla como usuario `CodeDeploy` administrativo

1. Antes de cerrar sesión, asegúrese de tener la URL del portal de AWS acceso y el nombre de usuario y la contraseña de un solo uso del usuario `CodeDeploy` administrativo.

 Note

Si no dispone de esta información, vaya a la página de detalles del usuario `CodeDeploy` administrativo del Centro de Identidad de IAM, seleccione Restablecer la contraseña, Generar una contraseña de un solo uso [...] y vuelva a restablecer la contraseña para que aparezca la información en la pantalla.

2. Cerrar sesión en AWS.
3. Pegue la URL del portal de AWS acceso en la barra de direcciones del navegador.
4. Inicie sesión como `CodeDeploy` usuario administrativo.

Aparece un cuadro de Cuenta de AWS en la pantalla.

5. Elija y Cuenta de AWS, Cuenta de AWS a continuación, elija el nombre del usuario `CodeDeploy` administrativo y el conjunto de permisos.
6. Junto a `CodeDeployUserPermissionSet`, seleccione Consola de administración.

Aparece el `AWS Management Console` . Ahora ha iniciado sesión como usuario `CodeDeploy` administrativo con los permisos limitados. Ahora puede realizar operaciones `CodeDeploy` relacionadas, y solo operaciones `CodeDeploy` relacionadas, como este usuario.

Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2

Note

Si utiliza la plataforma informática Amazon ECS o AWS Lambda, omita este paso.

Sus instancias de Amazon EC2 necesitan permiso para acceder a los GitHub depósitos o repositorios de Amazon S3 en los que se almacenan las aplicaciones. Para lanzar instancias de Amazon EC2 compatibles con CodeDeploy, debe crear una función de IAM adicional, un perfil de instancia. Estas instrucciones le indicarán cómo crear un perfil de instancia de IAM para asociarlo a las instancias Amazon EC2. Esta función otorga al CodeDeploy agente permiso para acceder a los GitHub depósitos o repositorios de Amazon S3 en los que se almacenan sus aplicaciones.

Puede crear un perfil de instancia de IAM con la AWS CLI consola de IAM o las API de IAM.

Note

Puede adjuntar un perfil de instancias de IAM a una instancia de Amazon EC2 en el momento de lanzarla, o bien, adjuntarlo a una instancia ya lanzada anteriormente. Para obtener más información, consulte [Perfiles de instancia](#).

Temas

- [Creación de un perfil de instancia de IAM para las instancias de Amazon EC2 \(CLI\)](#)
- [Creación de un perfil de instancia de IAM para las instancias de Amazon EC2 \(consola\)](#)

Creación de un perfil de instancia de IAM para las instancias de Amazon EC2 (CLI)

En estos pasos suponemos que ya ha seguido las instrucciones de los tres primeros pasos de [Empezar con CodeDeploy](#).

1. En el equipo de implementación, crea un archivo de texto con el nombre `CodeDeployDemo-EC2-Trust.json`. Pegue el contenido siguiente, que permite a Amazon EC2 actuar en su nombre:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. En el mismo directorio, cree un archivo de texto con el nombre `CodeDeployDemo-EC2-Permissions.json`. Pegue el siguiente contenido:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Note

Le recomendamos que restrinja esta política únicamente a los buckets de Amazon S3 a los que las instancias de Amazon EC2 deban tener acceso. Asegúrese de dar acceso a los buckets de Amazon S3 que contienen el CodeDeploy agente. De lo contrario, podría producirse un error cuando el CodeDeploy agente esté instalado o actualizado en las instancias. Para conceder al perfil de instancia de IAM acceso únicamente a algunos buckets del kit de CodeDeploy recursos de Amazon S3, utilice la siguiente política, pero elimine las líneas de los buckets a los que desee impedir el acceso:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::replace-with-your-s3-bucket-name/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
        "arn:aws:s3:::aws-codedeploy-il-central-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
        "arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-1/*",
        "arn:aws:s3:::aws-codedeploy-ap-south-2/*",
        "arn:aws:s3:::aws-codedeploy-me-central-1/*",
        "arn:aws:s3:::aws-codedeploy-me-south-1/*",
        "arn:aws:s3:::aws-codedeploy-sa-east-1/*"
      ]
    }
  ]
}
```

```
}
```

Note

Si quieres utilizar la [autorización de IAM](#) o los puntos de enlace de Amazon Virtual Private Cloud (VPC) CodeDeploy, tendrás que añadir más permisos. Consulte [Uso CodeDeploy con Amazon Virtual Private Cloud](#) para obtener más información.

- Desde el mismo directorio, llame al comando `create-role` para crear un rol de IAM denominado **CodeDeployDemo-EC2-Instance-Profile**, basado en la información del primer archivo:

Important

Asegúrese de incluir `file://` antes del nombre de archivo. Es obligatorio en este comando.

```
aws iam create-role --role-name CodeDeployDemo-EC2-Instance-Profile --assume-role-policy-document file://CodeDeployDemo-EC2-Trust.json
```

- Desde el mismo directorio, llame al comando `put-role-policy` para conceder al rol **CodeDeployDemo-EC2-Instance-Profile** los permisos correspondientes a la información del segundo archivo:

Important

Asegúrese de incluir `file://` antes del nombre de archivo. Es obligatorio en este comando.

```
aws iam put-role-policy --role-name CodeDeployDemo-EC2-Instance-Profile --policy-name CodeDeployDemo-EC2-Permissions --policy-document file://CodeDeployDemo-EC2-Permissions.json
```

- Llame al `attach-role-policy` para conceder permisos a Amazon EC2 Systems Manager para que SSM pueda instalar CodeDeploy el agente. Esta política no es necesaria si tiene pensado

instalar el agente desde el bucket de Amazon S3 público con la línea de comandos. Obtenga más información sobre la [instalación del agente de CodeDeploy](#).

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonSSMManagedInstanceCore --role-name CodeDeployDemo-EC2-Instance-Profile
```

6. Ejecute el comando `create-instance-profile` seguido del comando `add-role-to-instance-profile` para crear un perfil de instancia de IAM denominado **CodeDeployDemo-EC2-Instance-Profile**. El perfil de instancia permite a Amazon EC2 pasar el rol de IAM denominado **CodeDeployDemo-EC2-Instance-Profile** a una instancia de Amazon EC2 en el momento de lanzarla por primera vez:

```
aws iam create-instance-profile --instance-profile-name CodeDeployDemo-EC2-
Instance-Profile
aws iam add-role-to-instance-profile --instance-profile-name CodeDeployDemo-EC2-
Instance-Profile --role-name CodeDeployDemo-EC2-Instance-Profile
```

Si necesita obtener el nombre del perfil de la instancia de IAM, consulte [list-instance-profiles-for-role](#) en la sección IAM de la Referencia.AWS CLI

Ya ha creado un perfil de instancia de IAM para asociarlo a las instancias de Amazon EC2. Para obtener más información, consulte [Roles de IAM para Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Creación de un perfil de instancia de IAM para las instancias de Amazon EC2 (consola)

1. [Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. En el panel de navegación de la consola de IAM, elija Políticas y después Crear política.
3. En la página Especificar permisos, seleccione JSON.
4. Elimine el código de muestra de JSON.
5. Pegue el siguiente código:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}

```

Note

Le recomendamos que restrinja esta política únicamente a los buckets de Amazon S3 a los que las instancias de Amazon EC2 deban tener acceso. Asegúrese de dar acceso a los buckets de Amazon S3 que contienen el CodeDeploy agente. De lo contrario, podría producirse un error cuando el CodeDeploy agente esté instalado o actualizado en las instancias. Para conceder al perfil de instancia de IAM acceso únicamente a algunos buckets del kit de CodeDeploy recursos de Amazon S3, utilice la siguiente política, pero elimine las líneas de los buckets a los que desee impedir el acceso:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::replace-with-your-s3-bucket-name/*",
        "arn:aws:s3:::aws-codedeploy-us-east-2/*",
        "arn:aws:s3:::aws-codedeploy-us-east-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-1/*",
        "arn:aws:s3:::aws-codedeploy-us-west-2/*",
        "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
        "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
        "arn:aws:s3:::aws-codedeploy-eu-central-2/*",

```



```
"arn:aws:s3:::aws-codedeploy-eu-north-1/*",
"arn:aws:s3:::aws-codedeploy-eu-south-1/*",
"arn:aws:s3:::aws-codedeploy-eu-south-2/*",
"arn:aws:s3:::aws-codedeploy-il-central-1/*",
"arn:aws:s3:::aws-codedeploy-ap-east-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
"arn:aws:s3:::aws-codedeploy-ap-south-1/*",
"arn:aws:s3:::aws-codedeploy-ap-south-2/*",
"arn:aws:s3:::aws-codedeploy-me-central-1/*",
"arn:aws:s3:::aws-codedeploy-me-south-1/*",
"arn:aws:s3:::aws-codedeploy-sa-east-1/*"
  ]
}
]
```

Note

Si quieres utilizar la [autorización de IAM](#) o los puntos de enlace de Amazon Virtual Private Cloud (VPC) CodeDeploy, tendrás que añadir más permisos. Consulte [Uso CodeDeploy con Amazon Virtual Private Cloud](#) para obtener más información.

6. Elija Siguiente.
7. En la página Revisar y crear, en Nombre de la política, escriba **CodeDeployDemo-EC2-Permissions**.
8. (Opcional) En Descripción, escriba una descripción para la política.
9. Elija Crear política.
10. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
11. En Caso de uso, elija el caso de uso EC2.
12. Elija Siguiente.

13. En la lista de políticas, seleccione la casilla de verificación situada junto a la política que acaba de crear (CodeDeployDemo-EC2-Permissions). Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
14. Para usar Systems Manager para instalar o configurar el CodeDeploy agente, seleccione la casilla de verificación situada junto a AmazonSSM ManagedInstanceCore. Esta política AWS gestionada permite que una instancia utilice la funcionalidad principal del servicio Systems Manager. Si es necesario, utilice el cuadro de búsqueda para encontrar la política. Esta política no es necesaria si tiene pensado instalar el agente desde el bucket de Amazon S3 público con la línea de comandos. Obtenga más información sobre la [instalación del agente de CodeDeploy](#).
15. Elija Siguiente.
16. En la página Asignar nombre, revisar y crear, en Nombre del rol, ingrese un nombre para el rol de servicio (por ejemplo, **CodeDeployDemo-EC2-Instance-Profile**) y luego elija Crear rol.

También puede ingresar una descripción para este rol de servicio en el cuadro Descripción del rol.

Ya ha creado un perfil de instancia de IAM para asociarlo a las instancias de Amazon EC2. Para obtener más información, consulte [Roles de IAM para Amazon EC2](#) en la Guía del usuario de Amazon EC2.

Integraciones de productos y servicios con CodeDeploy

De forma predeterminada, CodeDeploy se integra con varios AWS servicios y productos y servicios de socios. La siguiente información puede ayudarle a CodeDeploy configurarse para integrarse con los productos y servicios que utiliza.

- [Integración con otros AWS servicios](#)
- [Integración con productos y servicios de socios](#)
- [Ejemplos de integración de la comunidad](#)

Integración con otros AWS servicios

CodeDeploy está integrado con los siguientes AWS servicios:

Amazon CloudWatch

[Amazon CloudWatch](#) es un servicio de supervisión de los recursos de la AWS nube y las aplicaciones en las que se ejecuta AWS. Puedes usar Amazon CloudWatch para recopilar y rastrear métricas, recopilar y monitorear archivos de registro y configurar alarmas. CodeDeploy admite las siguientes CloudWatch herramientas:

- CloudWatch alarmas para monitorear sus despliegues y detenerlos cuando las métricas de monitoreo especificadas superen o caigan por debajo de los umbrales que especifique en una regla de CloudWatch alarma. Para usar la supervisión de alarmas, primero debe configurar una alarma y CloudWatch, a continuación, agregarla CodeDeploy a la aplicación o al grupo de implementación, donde las implementaciones deberían detenerse cuando se active la alarma.

Más información:

- [Creación de CloudWatch registros \(alarmas\)](#)
- Amazon CloudWatch Events para detectar y reaccionar ante los cambios en el estado de una instancia o un despliegue en sus CodeDeploy operaciones. Luego, en función de las reglas que cree, CloudWatch Events invoca una o más acciones de destino cuando una implementación o instancia entra en el estado que especifique en una regla.

Más información:

- [Supervisión de despliegues con Amazon Events CloudWatch](#)
- Amazon CloudWatch Logs para monitorear los tres tipos de registros creados por el CodeDeploy agente sin tener que iniciar sesión en las instancias de una en una.

Más información:

- [Vea CodeDeploy los registros en la consola CloudWatch de Logs](#)

Amazon EC2 Auto Scaling

CodeDeploy es compatible con [Amazon EC2 Auto Scaling](#). Este AWS servicio puede lanzar automáticamente instancias de Amazon EC2 en función de los criterios que especifique, por ejemplo:

- Límites superados para la utilización de CPU especificada.
- Lecturas o escrituras en disco.
- Tráfico de red entrante o saliente durante un intervalo de tiempo especificado.

Puede escalar un grupo de instancias de Amazon EC2 siempre que las necesite y, a continuación, utilizarlas CodeDeploy para implementar revisiones de aplicaciones en ellas automáticamente. Amazon EC2 Auto Scaling termina las instancias de Amazon EC2 cuando dejan de ser necesarias.

Más información:

- [Integración CodeDeploy con Amazon EC2 Auto Scaling](#)
- [Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling](#)
- [Bajo el capó: CodeDeploy e integración con Auto Scaling](#)

Amazon Elastic Container Service

Puede utilizarla CodeDeploy para implementar una aplicación contenerizada de Amazon ECS como un conjunto de tareas. CodeDeploy realiza una implementación azul/verde mediante la instalación de una versión actualizada de la aplicación como un nuevo conjunto de tareas de reemplazo. CodeDeploy y redirige el tráfico de producción del conjunto de tareas de la aplicación original al conjunto de tareas de reemplazo. Cuando la implementación se realiza correctamente, se termina el conjunto de tareas original. Para obtener más información acerca de Amazon ECS, consulte [Amazon Elastic Container Service](#).

Puede administrar la forma en que el tráfico se desplaza al conjunto de tareas actualizado durante una implementación eligiendo una configuración canaria, lineal o all-at-once lineal. Para obtener más información acerca de las implementaciones de Amazon ECS, consulte [Implementaciones en una plataforma de informática de Amazon ECS](#).

AWS CloudTrail

CodeDeploy está integrado con [AWS CloudTrail](#). Este servicio captura las llamadas a la API realizadas por o en su nombre CodeDeploy y en su AWS cuenta y entrega los archivos de registro a un bucket de Amazon S3 que especifique. CloudTrail captura las llamadas a la API desde la CodeDeploy consola, desde CodeDeploy los comandos a través de ella o directamente desde las CodeDeploy API. AWS CLI Con la información recopilada por CloudTrail, puede determinar:

- Qué solicitud se hizo a CodeDeploy.
- La dirección IP de origen desde la que se realizó la solicitud.
- Quién ha realizado la solicitud
- Cuando se realizó.

Más información:

- [Monitoring Deployments](#)

AWS Cloud9

[AWS Cloud9](#) es un entorno de desarrollo integrado (IDE) en línea y basado en la nube que puede utilizar para escribir, ejecutar, depurar e implementar código con solo un navegador desde una máquina conectada a Internet. AWS Cloud9 incluye un editor de código, un depurador, un terminal y herramientas esenciales, como Git. AWS CLI

- Puedes usar el AWS Cloud9 IDE para ejecutar, depurar y compilar el código que se encuentra en un GitHub repositorio. Puede ver, modificar y guardar el código utilizando la ventana Environment (Entorno) de su IDE y las pestañas del editor. Cuando estés listo, puedes usar Git en la sesión de AWS Cloud9 terminal para enviar los cambios de código a tu GitHub repositorio y luego usarlo AWS CodeDeploy para implementar tus actualizaciones. Para obtener más información sobre cómo usarlo AWS Cloud9 con GitHub, consulta un [GitHub ejemplo de AWS Cloud9](#).
- Puede utilizar el AWS Cloud9 IDE para actualizar una AWS Lambda función. A continuación, puede utilizarlo AWS CodeDeploy para crear una implementación que desplace el tráfico a la nueva versión de la AWS Lambda función. Para obtener más información, consulte [Trabajar con AWS Lambda funciones en el entorno de desarrollo o AWS Cloud9 integrado \(IDE\)](#).

Para obtener más información AWS Cloud9, consulte [Qué es AWS Cloud9](#) y [Cómo empezar AWS Cloud9](#).

AWS CodePipeline

[AWS CodePipeline](#) es un servicio de entrega continua que permite modelar, visualizar y automatizar los pasos necesarios para lanzar software en un proceso de entrega continua. AWS CodePipeline le permite definir su propio proceso de lanzamiento para que el servicio compile, pruebe e implemente el código cada vez que cambie. Por ejemplo, puede tener tres grupos de implementaciones para una aplicación: Beta, Gamma y Prod. Puede configurar una canalización para que, cada vez que ocurra un cambio en el código fuente, las actualizaciones se implementen en cada uno de los grupos.

Puede configurarlo AWS CodePipeline para usar CodeDeploy para implementar:

- Código en instancias de Amazon EC2, instancias en las instalaciones o ambas.
- Versiones de la función AWS Lambda sin servidor.

Puede crear la CodeDeploy aplicación, el despliegue y el grupo de despliegues para usarlos en una acción de despliegue en una fase, ya sea antes de crear la canalización o en el asistente de creación de canalización.

Más información:

- [AWS guía de DevOps introducción](#): aprenda a utilizar CodePipeline with CodeDeploy para entregar e implementar de forma continua el código fuente en CodeCommit los repositorios de las instancias de Amazon EC2.

- [Tutorial: Crear una canalización simple \(bucket de Amazon S3\)](#)
- [Tutorial sencillo sobre la canalización \(repositorio\) CodeCommit](#)
- [Tutorial: Crear una canalización de cuatro etapas](#)

AWS Modelo de aplicación sin servidor

AWS El modelo de aplicaciones sin servidor (AWS SAM) es un modelo para definir las aplicaciones sin servidor. Se amplía AWS CloudFormation para proporcionar una forma simplificada de definir AWS Lambda las funciones, las API de Amazon API Gateway y las tablas de Amazon DynamoDB requeridas por una aplicación sin servidor. Si ya utiliza AWS SAM, puede añadir preferencias de implementación para empezar CodeDeploy a utilizarlas y administrar la forma en que se desplaza el tráfico durante la implementación de una aplicación AWS Lambda.

Para obtener más información, consulte [AWS Serverless Application Model](#).

Elastic Load Balancing

CodeDeploy es compatible con [Elastic Load Balancing](#), un servicio que distribuye el tráfico de aplicaciones entrante entre varias instancias de Amazon EC2.

En el caso de CodeDeploy las implementaciones, los balanceadores de carga también evitan que el tráfico se dirija a las instancias cuando no están listas, se están implementando actualmente o ya no se necesitan como parte de un entorno.

Más información:

- [Integrating CodeDeploy with Elastic Load Balancing](#)

Temas

- [Integración CodeDeploy con Amazon EC2 Auto Scaling](#)
- [Integración CodeDeploy con Elastic Load Balancing](#)

Integración CodeDeploy con Amazon EC2 Auto Scaling

CodeDeploy es compatible con Amazon EC2 Auto Scaling, un AWS servicio que lanza instancias de Amazon EC2 automáticamente según las condiciones que defina. Estas condiciones pueden incluir que se supere un límite de tiempo específico para el uso de la CPU, que se lea o escriba el disco, o el tráfico de red entrante y saliente. Amazon EC2 Auto Scaling termina las instancias cuando dejan de ser necesarias. Para obtener más información, consulte [Grupo de Amazon EC2 Auto Scaling](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Cuando se lanzan nuevas instancias de Amazon EC2 como parte de un CodeDeploy grupo de Auto Scaling de Amazon EC2, puede implementar sus revisiones en las nuevas instancias automáticamente. También puede coordinar las implementaciones CodeDeploy con instancias de Auto Scaling de Amazon EC2 registradas en los balanceadores de carga de Elastic Load Balancing. Para obtener más información, consulte [Integrating CodeDeploy with Elastic Load Balancing](#) y

[Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2.](#)

Note

Tenga en cuenta que es posible que surjan problemas si asocia varios grupos de implementación con un único grupo de Amazon EC2 Auto Scaling. Por ejemplo, si falla una implementación, la instancia comenzará a cerrarse, pero el resto de las implementaciones en ejecución puede que tarden una hora en agotar el tiempo de espera. Para obtener más información, consulte [Under Evite asociar varios grupos de implementación a un único grupo de Amazon EC2 Auto Scaling the Hood: CodeDeploy y la integración de Auto Scaling de Amazon EC2.](#)

Temas

- [Implementación de CodeDeploy aplicaciones en grupos de Auto Scaling de Amazon EC2](#)
- [Habilitación de implementaciones de terminación durante los eventos de reducción horizontal de Auto Scaling](#)
- [Cómo funciona Amazon EC2 Auto Scaling con CodeDeploy](#)
- [Uso de una AMI personalizada con CodeDeploy Amazon EC2 Auto Scaling](#)

Implementación de CodeDeploy aplicaciones en grupos de Auto Scaling de Amazon EC2

Para implementar una revisión de CodeDeploy la aplicación en un grupo de Auto Scaling de Amazon EC2:

1. Cree o localice un perfil de instancia de IAM que permita al grupo de Amazon EC2 Auto Scaling trabajar con Amazon S3. Para obtener más información, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2.](#)

Note

También puede utilizarlas CodeDeploy para implementar revisiones desde los GitHub repositorios en los grupos de Auto Scaling de Amazon EC2. Si bien las instancias de

Amazon EC2 siguen necesitando un perfil de instancia de IAM, el perfil no necesita permisos adicionales para implementarse desde un repositorio. GitHub

2. Cree o utilice un grupo de Amazon EC2 Auto Scaling, especificando el perfil de instancia de IAM en su configuración o plantilla de lanzamiento. Para obtener más información, consulte [Rol de IAM para aplicaciones que se ejecutan en instancias de Amazon EC2](#).
3. Cree o localice un rol de servicio que permita CodeDeploy crear un grupo de implementación que contenga el grupo Auto Scaling de Amazon EC2.
4. Cree un grupo de implementación especificando el nombre del grupo de Auto Scaling de Amazon EC2, la función de servicio y algunas opciones más. CodeDeploy Para obtener más información, consulte [Creación de un grupo de implementación para una implementación local \(consola\)](#) o [Creación de un grupo de implementación para una implementación local \(consola\)](#).
5. CodeDeploy Utilícela para implementar la revisión en el grupo de implementación que contiene el grupo Auto Scaling de Amazon EC2.

Para obtener más información, consulte [Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling](#).

Habilitación de implementaciones de terminación durante los eventos de reducción horizontal de Auto Scaling

Un despliegue de terminación es un tipo de CodeDeploy despliegue que se activa automáticamente cuando se produce un [evento de escalado](#) interno de Auto Scaling. CodeDeploy realiza el despliegue de terminación justo antes de que el servicio Auto Scaling finalice la instancia. Durante un despliegue de terminación, CodeDeploy no despliega nada. En su lugar, genera eventos de ciclo de vida, que puede enlazar a sus propios scripts para habilitar una funcionalidad de apagado personalizada. Por ejemplo, puede enlazar el evento del ciclo de vida de `ApplicationStop` a un script que apague la aplicación sin problemas antes de que finalice la instancia.

Para obtener una lista de los eventos del ciclo de vida que se CodeDeploy generan durante un despliegue de terminación, consulte [Disponibilidad de los enlaces de eventos del ciclo de vida](#).

Si la implementación de la terminación falla por algún motivo, CodeDeploy permitirá que la finalización de la instancia continúe. Esto significa que la instancia se cerrará aunque CodeDeploy no se haya ejecutado todo el conjunto (o ninguno) de los eventos del ciclo de vida hasta completarse.

Si no habilita las implementaciones de terminación, el servicio Auto Scaling seguirá cancelando las instancias de Amazon EC2 cuando se produzca un evento de escalado interno, CodeDeploy pero no generará eventos de ciclo de vida.

Note

Independientemente de si habilita las implementaciones de terminación o no, si el servicio Auto Scaling finaliza una instancia de Amazon EC2 mientras la implementación está en marcha, puede producirse CodeDeploy una situación de carrera entre los eventos del ciclo de vida generados por Auto Scaling y los servicios. CodeDeploy Por ejemplo, el evento `Terminating` del ciclo de vida (generado por el servicio Auto Scaling) podría anular el `ApplicationStart` evento (generado por la CodeDeploy implementación). En este escenario, es posible que se produzca un error al finalizar la instancia de Amazon EC2 o al CodeDeploy implementarse.

Para permitir la realización CodeDeploy de despliegues de terminación

- Seleccione la casilla Agregar un enlace de terminación a los grupos de Auto Scaling al crear o actualizar su grupo de implementación. Para obtener instrucciones, consulte [Creación de un grupo de implementación para una implementación local \(consola\)](#) o [Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones \(consola\)](#).

Al activar esta casilla de verificación, CodeDeploy se instala un [enlace de ciclo](#) de vida de Auto Scaling en los grupos de Auto Scaling que especifique al crear o actualizar el grupo de CodeDeploy implementación. Este enlace se denomina enlace de terminación y permite las implementaciones de terminación.

Una vez instalado el enlace de terminación, se produce un evento de reducción horizontal (terminación) de la siguiente manera:

1. El servicio de Auto Scaling (o simplemente Auto Scaling) determina que debe producirse un evento de reducción horizontal y se pone en contacto con el servicio de EC2 para terminar una instancia de EC2.
2. El servicio de EC2 comienza la terminación de la instancia EC2. La instancia pasa al estado `Terminating` y, después, al estado `Terminating:Wait`.

3. Durante este proceso `Terminating:Wait`, Auto Scaling ejecuta todos los ganchos del ciclo de vida adjuntos al grupo Auto Scaling, incluido el gancho de terminación instalado por CodeDeploy.
4. El enlace de terminación envía una notificación a la [cola de Amazon SQS](#) que es sondeada. CodeDeploy
5. [Al recibir la notificación, CodeDeploy analiza el mensaje, realiza algunas validaciones y realiza un despliegue de terminación.](#)
6. Mientras se ejecuta la implementación de terminación, CodeDeploy envía latidos cada cinco minutos a Auto Scaling para avisarle de que la instancia aún se está trabajando en la instancia.
7. Hasta ahora, la instancia de EC2 sigue en el estado `Terminating:Wait` (o posiblemente en el estado `Warmup:Pending:Wait`, si ha activado los [grupos de calentamiento de grupos de Auto Scaling](#)).
8. Cuando se complete el despliegue, CodeDeploy indica a Auto Scaling CONTINUE el proceso de terminación de EC2, independientemente de si el despliegue de terminación se realizó correctamente o no.

Cómo funciona Amazon EC2 Auto Scaling con CodeDeploy

Al crear o actualizar un grupo de CodeDeploy implementación para incluir un grupo de Auto Scaling, CodeDeploy accede al grupo de Auto Scaling mediante el rol de CodeDeploy servicio y, a continuación, instala los [enlaces del ciclo de vida de Auto Scaling](#) en sus grupos de Auto Scaling.

Note

Los ganchos del ciclo de vida de Auto Scaling son diferentes de los eventos del ciclo de vida (también denominados ganchos de eventos del ciclo [AppSpec sección de «ganchos»](#) de vida) generados CodeDeploy y descritos en esta guía.

Los ganchos del ciclo de vida de Auto Scaling que CodeDeploy instala son:

- Un enlace de lanzamiento: este enlace notifica CodeDeploy que hay un [evento de escalado horizontal de](#) Auto Scaling en curso y que es CodeDeploy necesario iniciar una implementación de lanzamiento.

Durante una implementación de lanzamiento, CodeDeploy

- Implementa una revisión de la aplicación en la instancia escalada horizontalmente.

- Genera eventos de ciclo de vida para indicar el progreso de la implementación. Puede enlazar estos eventos de ciclo de vida a sus propios scripts para habilitar una funcionalidad de inicio personalizada. Para obtener más información, consulte la tabla en [Disponibilidad de los enlaces de eventos del ciclo de vida](#).

El enlace de lanzamiento y la implementación de lanzamiento asociada siempre están habilitados y no se pueden desactivar.

- Un enlace de terminación: este enlace opcional notifica CodeDeploy que se está produciendo un [evento de escalado](#) automático de Auto Scaling y que CodeDeploy debe iniciar un despliegue de terminación.

Durante una implementación de terminación, CodeDeploy genera eventos del ciclo de vida para indicar el progreso del cierre de la instancia. Para obtener más información, consulte [Habilitación de implementaciones de terminación durante los eventos de reducción horizontal de Auto Scaling](#).

Temas

- [Una vez CodeDeploy instalados los enlaces del ciclo de vida, ¿cómo se utilizan?](#)
- [Cómo se CodeDeploy nombran los grupos de Auto Scaling de Amazon EC2](#)
- [Orden de ejecución de eventos de enlace de ciclo de vida personalizados](#)
- [Eventos de escalado horizontal durante una implementación](#)
- [Eventos de reducción horizontal durante una implementación](#)
- [Orden de los eventos en los scripts AWS CloudFormation cfn-init](#)

Una vez CodeDeploy instalados los enlaces del ciclo de vida, ¿cómo se utilizan?

Una vez instalados los ganchos del ciclo de vida de lanzamiento y finalización, los utilizan CodeDeploy durante los eventos de escalado horizontal y horizontal de grupo de Auto Scaling, respectivamente.

Un evento de escalado horizontal (lanzamiento) se desarrolla de la siguiente manera:

1. El servicio de escalado automático (o simplemente Auto Scaling) determina que debe producirse un evento de escalado horizontal y se pone en contacto con el servicio de EC2 para lanzar una nueva instancia de EC2.
2. El servicio de EC2 lanza una nueva instancia de EC2. La instancia pasa al estado Pending y, después, al estado Pending:Wait.

3. Durante este proceso `Pending:Wait`, Auto Scaling ejecuta todos los ganchos del ciclo de vida adjuntos al grupo Auto Scaling, incluido el gancho de lanzamiento instalado por CodeDeploy.
4. El enlace de lanzamiento envía una notificación a la [cola de Amazon SQS consultada](#) por CodeDeploy.
5. [Al recibir la notificación, CodeDeploy analiza el mensaje, realiza algunas validaciones e inicia una implementación de lanzamiento.](#)
6. Mientras se ejecuta la implementación de lanzamiento, CodeDeploy envía latidos cada cinco minutos a Auto Scaling para avisarle de que la instancia aún se está trabajando en la instancia.
7. Hasta el momento, la instancia EC2 sigue en el estado `Pending:Wait`.
8. Cuando se complete la implementación, CodeDeploy indica a Auto Scaling el proceso de lanzamiento `CONTINUE` o `ABANDON` el de EC2, dependiendo de si la implementación se realizó correctamente o no.
 - Si así CodeDeploy lo indica `CONTINUE`, Auto Scaling continúa con el proceso de lanzamiento, ya sea esperando a que se completen otros enlaces o poniendo la instancia en el estado `Pending:Proceed` y luego en el `InService` estado.
 - Si así CodeDeploy lo indica `ABANDON`, Auto Scaling finaliza la instancia EC2 y reinicia el procedimiento de lanzamiento si es necesario para cumplir con el número deseado de instancias, tal como se define en la configuración de Capacidad deseada de Auto Scaling.

Un evento de reducción horizontal (terminación) se desarrolla de la siguiente manera:

Consulte [Habilitación de implementaciones de terminación durante los eventos de reducción horizontal de Auto Scaling](#).

Cómo se CodeDeploy nombran los grupos de Auto Scaling de Amazon EC2

Durante las implementaciones azul/verde en una plataforma de informática de EC2/en las instalaciones, tiene dos opciones para añadir instancias al entorno de sustitución (verde):

- Utilice instancias que ya existan o que cree manualmente.
- Utilizar la configuración de un grupo de Amazon EC2 Auto Scaling especificado para definir y crear instancias en un nuevo grupo de Amazon EC2 Auto Scaling.

Si elige la segunda opción, le CodeDeploy aprovisiona un nuevo grupo de Auto Scaling de Amazon EC2. Utiliza la siguiente convención para el nombre del grupo:

```
CodeDeploy_deployment_group_name_deployment_id
```

Por ejemplo, si una implementación con ID 10 implementa un grupo de implementación llamado `alpha-deployments`, el grupo de Amazon EC2 Auto Scaling aprovisionado se llama `CodeDeploy_alpha-deployments_10`. Para obtener más información, consulte [Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones \(consola\)](#) y [GreenFleetProvisioningOption](#).

Orden de ejecución de eventos de enlace de ciclo de vida personalizados

Puede añadir sus propios enlaces de ciclo de vida a los grupos de Auto Scaling de Amazon EC2 en los que se CodeDeploy realiza la implementación. Sin embargo, el orden en que se ejecutan esos eventos de enlace del ciclo de vida personalizados no se puede determinar en relación con los eventos del ciclo de vida de la implementación CodeDeploy predeterminados. Por ejemplo, si agrega un enlace de ciclo de vida personalizado denominado `ReadyForSoftwareInstall` a un grupo de Auto Scaling de Amazon EC2, no podrá saber de antemano si se ejecutará antes del primer evento del ciclo de vida de la implementación CodeDeploy predeterminado o después del último.

Para obtener información sobre cómo añadir enlaces de ciclo de vida personalizados a un grupo de Amazon EC2 Auto Scaling, consulte [Añadir enlaces de ciclo de vida](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Eventos de escalado horizontal durante una implementación

Si se produce un evento de escalado horizontal de Auto Scaling mientras está en curso una implementación, las nuevas instancias se actualizarán con la revisión de la aplicación que se implementó antes, no con la última revisión de la aplicación. Si la implementación se realiza correctamente, las instancias antiguas y las instancias recién escaladas horizontalmente hospedarán revisiones de la aplicación diferentes. Para actualizar las instancias con la revisión anterior, inicia CodeDeploy automáticamente una implementación posterior (inmediatamente después de la primera) para actualizar las instancias desactualizadas. Si desea cambiar este comportamiento predeterminado para que las instancias de EC2 desactualizadas permanezcan en la revisión anterior, consulte [Automatic updates to outdated instances](#).

Si desea suspender los procesos de escalado horizontal de Auto Scaling de Amazon EC2 mientras se llevan a cabo las implementaciones, puede hacerlo mediante una configuración en el `common_functions.sh` script que se utiliza para equilibrar la carga con CodeDeploy. Si `HANDLE_PROCS=true`, los siguientes eventos de escalado automático se suspenden automáticamente durante el proceso de implementación:

- AZRebalance
- AlarmNotification
- ScheduledActions
- ReplaceUnhealthy

 Important

Solo el. CodeDeployDefault OneAtLa configuración de despliegue de aTime admite esta funcionalidad.

Para obtener más información sobre cómo `HANDLE_PROCS=true` usar Amazon EC2 Auto Scaling para evitar problemas de implementación, consulte el [Aviso importante sobre la gestión de AutoScaling procesos en aws-codedeploy-sampleson](#). GitHub

Eventos de reducción horizontal durante una implementación

Si un grupo de Auto Scaling comienza a ampliarse mientras se está realizando una CodeDeploy implementación en ese grupo de Auto Scaling, podría producirse una condición de carrera entre el proceso de finalización (incluidos los eventos del ciclo de vida del despliegue de CodeDeploy terminación) y otros eventos CodeDeploy del ciclo de vida en la instancia de terminación. La implementación en esa instancia específica puede fallar si la instancia se termina antes de que se completen todos los eventos CodeDeploy del ciclo de vida. Además, la CodeDeploy implementación general puede fallar o no, en función de cómo haya establecido el valor mínimo de hosts en buen estado en la configuración de la implementación.

Orden de los eventos en los scripts AWS CloudFormation `cfn-init`

Si utiliza `cfn-init` (o `cloud-init`) para ejecutar scripts en instancias basadas en Linux recién provisionadas, las implementaciones podrían fallar a menos que controle estrictamente el orden de los eventos que se producen después que se inicia la instancia.

Ese orden debe ser:

1. Se inicia la instancia recién provisionada.
2. Se ejecutan todos los scripts de proceso de arranque `cfn-init` hasta su finalización.
3. El CodeDeploy agente comienza.
4. Se implementa la última revisión de la aplicación en la instancia.

Si el orden de los eventos no se controla cuidadosamente, el CodeDeploy agente podría iniciar una implementación antes de que todos los scripts hayan terminado de ejecutarse.

Para controlar el orden de los eventos, utilice una de estas prácticas recomendadas:

- Instale el CodeDeploy agente mediante un `cfn-init` script y colóquelo después de todos los demás scripts.
- Incluya el CodeDeploy agente en una AMI personalizada y utilice un `cfn-init` script para iniciarlo, colocándolo después de todos los demás scripts.

Para obtener información acerca de cómo usar `cfn-init`, consulte [cfn-init](#) en la Guía del usuario de AWS CloudFormation .

Uso de una AMI personalizada con CodeDeploy Amazon EC2 Auto Scaling

Dispone de dos opciones para especificar la AMI base que se debe usar cuando se lancen nuevas instancias de Amazon EC2 en un grupo de Amazon EC2 Auto Scaling:

- Puede especificar una AMI base personalizada que ya tenga el CodeDeploy agente instalado. Dado que el agente ya está instalado, esta opción lanza las nuevas instancias de Amazon EC2 con mayor rapidez que la otra opción. Sin embargo, esta opción ofrece una mayor probabilidad de que las implementaciones iniciales de las instancias de Amazon EC2 fallen, especialmente si CodeDeploy el agente está desactualizado. Si elige esta opción, le recomendamos que actualice periódicamente el CodeDeploy agente en su AMI personalizada base.
- Puede especificar una AMI base que no tenga el CodeDeploy agente instalado y tenerlo instalado cada vez que se lance una nueva instancia en un grupo de Auto Scaling de Amazon EC2. Aunque esta opción lanza las nuevas instancias de Amazon EC2 más lentamente que la otra opción, tiene mayor probabilidad de que las implementaciones iniciales de las instancias sean correctas. Esta opción usa la versión más reciente del CodeDeploy agente.

Integración CodeDeploy con Elastic Load Balancing

Durante CodeDeploy las implementaciones, un balanceador de carga evita que el tráfico de Internet se dirija a las instancias cuando no están listas, se están implementando actualmente o ya no se necesitan como parte de un entorno. Sin embargo, el rol exacto que desempeña el balanceador de carga depende de si se utiliza en una implementación "blue/green" (azul/verde) o una implementación in situ.

Note

El uso de equilibradores de carga de Elastic Load Balancing es obligatorio en las implementaciones azul/verde y opcional en las implementaciones locales.

Tipos de Elastic Load Balancing

Elastic Load Balancing proporciona tres tipos de balanceadores de carga que se pueden usar en CodeDeploy las implementaciones: balanceadores de carga clásicos, balanceadores de carga de aplicaciones y balanceadores de carga de red.

Equilibrador de carga clásico

Enruta y equilibra la carga en la capa de transporte (TCP/SSL) o la capa de aplicación (HTTP/HTTPS). Es compatible con una VPC.

Note

Los equilibradores de carga clásicos no son compatibles con las implementaciones de Amazon ECS.

Equilibrador de carga de aplicación

Enruta y equilibra la carga en la capa de la aplicación (HTTP/HTTPS) y admite el enrutamiento basado en rutas. Puede dirigir las solicitudes a puertos de cada instancia EC2 o instancia de contenedor de su nube privada virtual (VPC).

Note

Los grupos objetivo del Equilibrador de carga de aplicación deben tener un tipo de destino `instance` para las implementaciones en instancias EC2 y `IP` para las implementaciones de Fargate. Para obtener más información, consulte [Tipo de destino](#).

Equilibrador de carga de red

Enruta y equilibra la carga en la capa de transporte (capa 4 de TCP/UDP) basándose en la información de las direcciones que extrae del encabezado del paquete TCP y no de su contenido.

Los balanceadores de carga de red pueden gestionar ráfagas de tráfico, conservar la IP de origen del cliente y utilizar una IP fija mientras dura la vida útil del balanceador de carga.

Para obtener más información sobre los equilibradores de carga de Elastic Load Balancing, consulte los temas siguientes:

- [¿Qué es Elastic Load Balancing?](#)
- [¿Qué es un equilibrador de carga clásico?](#)
- [¿Qué es un equilibrador de carga de aplicación?](#)
- [¿Qué es un equilibrador de carga de red?](#)

Implementaciones azul/verde

Redireccionar el tráfico de instancias detrás de un balanceador de cargas de Elastic Load Balancing es fundamental para las implementaciones CodeDeploy azules y ecológicas.

Durante una implementación "blue/green" (azul/verde), el balanceador de carga permite dirigir el tráfico a las nuevas instancias de un grupo de implementaciones en el que se haya implementado la última revisión de la aplicación (el entorno de sustitución) de acuerdo con las reglas que especifique y, a continuación, bloquear el tráfico de las instancias antiguas en las que se ejecutaba la revisión de la aplicación previa (entorno original).

Una vez que las instancias de un entorno de sustitución se hayan registrado en uno o varios equilibradores de carga, se cancela el registro de las instancias del entorno original y, si así lo decide, se terminan las instancias.

Para una implementación azul/verde, puede especificar uno o más equilibradores de carga clásicos, grupos de destino de equilibrador de carga de aplicación o grupos de destino de equilibrador de carga de red en su grupo de implementación. Utiliza la CodeDeploy consola o AWS CLI para añadir los balanceadores de carga a un grupo de implementaciones.

Para obtener más información sobre los equilibradores de carga en las implementaciones "blue/green" (azul/verde), consulte los siguientes temas:

- [Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2](#)
- [Creación de una aplicación para una implementación azul/verde \(consola\)](#)

- [Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones \(consola\)](#)

Implementaciones in situ

Durante una implementación local, el balanceador de carga impide que el tráfico de Internet se dirija a la instancia en la que está ocurriendo, y la vuelve a poner a disposición una vez finalizada dicha implementación.

Si no se utiliza un balanceador de carga durante una implementación local, es posible que el tráfico de Internet aún se dirija a una instancia durante el proceso de implementación. Como resultado, los clientes podrían encontrarse con aplicaciones web dañadas, incompletas o anticuadas. Cuando utilizas un balanceador de cargas de Elastic Load Balancing con una implementación local, las instancias de un grupo de implementación se cancelan del balanceador de cargas, se actualizan con la última revisión de la aplicación y, a continuación, se vuelven a registrar en el balanceador de cargas como parte del mismo grupo de implementación una vez que la implementación se ha realizado correctamente. CodeDeploy esperará hasta 1 hora para que la instancia se recupere correctamente tras el balanceador de carga. Si el balanceador de cargas no marca la instancia como en buen estado durante el período de espera, CodeDeploy pasa a la siguiente instancia o no se realiza la implementación, según la configuración de la implementación.

Para una implementación local, puede especificar uno o más equilibradores de carga clásicos, grupos de destino de equilibrador de carga de aplicación o grupos de destino de equilibrador de carga de red. Puedes especificar los balanceadores de carga como parte de la configuración del grupo de implementación, o puedes usar un script proporcionado por él CodeDeploy para implementar los balanceadores de carga.

Especificación de un equilibrador de carga de una implementación local mediante un grupo de implementación

Para añadir balanceadores de carga a un grupo de implementación, usa la CodeDeploy consola o. AWS CLI Para obtener información sobre cómo especificar un balanceador de carga en un grupo de implementaciones para implementaciones in situ, consulte los siguientes temas:

- [Creación de una aplicación para una implementación local \(consola\)](#)
- [Creación de un grupo de implementación para una implementación local \(consola\)](#)
- [Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2](#)

Especificación de un equilibrador de carga de una implementación local mediante un script

Utilice los pasos que se indican en el siguiente procedimiento para usar scripts de ciclo de vida para configurar el balanceo de carga para implementaciones locales.

Note

Deberías usar el CodeDeployDefault.OneAtUna configuración de despliegue aTime solo cuando se utiliza un script para configurar un equilibrador de carga para un despliegue in situ. No se admiten las ejecuciones simultáneas y las. CodeDeployDefault.OneAtUna configuración de tiempo garantiza la ejecución en serie de los scripts. Para obtener más información acerca de las configuraciones de implementación, consulte [Trabajar con configuraciones de implementación en CodeDeploy](#).

En el repositorio de CodeDeploy muestras GitHub, incluimos instrucciones y ejemplos que puede adaptar para usar los balanceadores de carga de CodeDeploy Elastic Load Balancing. En estos repositorios se incluyen tres scripts de ejemplo (`register_with_elb.sh`, `deregister_from_elb.sh` y `common_functions.sh`) que proporcionan todo el código que necesitará para comenzar. Solo tiene que editar los marcadores de posición de estos tres scripts y después hacer referencia a estos scripts en el archivo `appspec.yml`.

Para configurar las implementaciones in situ en CodeDeploy instancias de Amazon EC2 que estén registradas en los balanceadores de carga de Elastic Load Balancing, haga lo siguiente:

1. Descargue los ejemplos del tipo de balanceador de carga que desea utilizar para una implementación in situ:
 - [Equilibrador de carga clásico](#)
 - [Equilibrador de carga de aplicación o Equilibrador de carga de red \(se puede usar el mismo script para ambos tipos\)](#)
2. Asegúrese de que cada una de las instancias de Amazon EC2 de destino la AWS CLI tenga instalada.
3. Asegúrese de que cada una de sus instancias de Amazon EC2 de destino tenga un perfil de instancia de IAM asociado con los permisos `elasticloadbalancing:*` y `autoscaling:*` como mínimo.
4. En el directorio de código fuente de la aplicación, incluya los scripts de evento de ciclo de vida de implementación (`register_with_elb.sh`, `deregister_from_elb.sh` y `common_functions.sh`).

5. En la `appspec.yml` revisión de la aplicación, proporcione instrucciones CodeDeploy para ejecutar el `register_with_elb.sh` script durante el `ApplicationStart` evento y el `deregister_from_elb.sh` script durante el `ApplicationStop` evento.
6. Si la instancia forma parte de un grupo de Amazon EC2 Auto Scaling, puede omitir este paso.

En el script `common_functions.sh`:

- Si utiliza el [Equilibrador de carga clásico](#), especifique los nombres de los equilibradores de carga de Elastic Load Balancing en `ELB_LIST=""` y realice los cambios que necesite en el resto de la configuración de la implementación en el archivo.
 - Si utiliza el [Equilibrador de carga de aplicación o el Equilibrador de carga de red](#), especifique los nombres del grupo de destino de Elastic Load Balancing en `TARGET_GROUP_LIST=""` y realice los cambios que necesite en el resto de la configuración de la implementación en el archivo.
7. Empaquete el código fuente de la aplicación, el archivo `appspec.yml` y los scripts de eventos de ciclo de vida de la implementación en una revisión de la aplicación y después cargue la revisión. Implemente la revisión en las instancias de Amazon EC2. Durante la implementación, los scripts de eventos de ciclo de vida de la implementación cancelarán el registro de la instancia de Amazon EC2 con el equilibrador de carga, esperarán a que se vacíe la conexión y volverán a registrar la instancia de Amazon EC2 con el equilibrador de carga una vez que se complete la implementación.

Integración con productos y servicios de socios

CodeDeploy tiene una integración integrada para los siguientes productos y servicios de los socios:

Ansible

Si ya dispone de un conjunto de manuales de [Ansible](#), pero solo necesita un lugar donde ejecutarlos, la plantilla de Ansible CodeDeploy y demuestra cómo un par de sencillos enlaces de despliegue pueden garantizar que Ansible esté disponible en la instancia de despliegue local y ejecute los manuales de estrategias. Si ya tienes un proceso para crear y mantener tu inventario, también hay un módulo de Ansible

que puedes usar para instalar y ejecutar el agente. CodeDeploy

Más información:

- [Ansible y CodeDeploy](#)

Atlassian: Bamboo y Bitbucket

La CodeDeploy tarea de [Bamboo](#) comprime el directorio que contiene un AppSpec archivo en un archivo.zip, carga el archivo en Amazon S3 y, a continuación, inicia la implementación de acuerdo con la configuración proporcionada en la aplicación. CodeDeploy

La compatibilidad con Bitbucket de Atlassian te CodeDeploy permite enviar código a las instancias de Amazon EC2 directamente desde la interfaz de usuario de Bitbucket, bajo demanda, a cualquiera de tus grupos de implementación. Esto significa que después de actualizar código en el repositorio Bitbucket , no necesita iniciar sesión en la plataforma de integración continua (CI) o en instancias de Amazon EC2 para ejecutar un proceso de implementación manual.

Más información:

- [Uso de la tarea para Bamboo CodeDeploy](#)
- [Anunciamos el soporte de Atlassian Bitbucket para CodeDeploy](#)

Chef

AWS [proporciona dos ejemplos de plantillas para integrar Chef y CodeDeploy](#). El primero es un libro de cocina de Chef que instala e inicia el CodeDeploy agente. Esto le permite continuar administrando su infraestructura de host con Chef mientras la usa. CodeDeploy La segunda plantilla de ejemplo muestra cómo organizarla CodeDeploy para organizar la publicación de libros de cocina y recetas con el chef solo en cada nodo.

Más información:

- [Chef y CodeDeploy](#)

CircleCI

[CircleCI](#) ofrece un conjunto de herramientas de prueba automatizada y de implementación e integración continuas. Tras crear un rol de IAM AWS para usarlo con CircleCI y configurar los parámetros de despliegue en el archivo circle.yml, puede usar CircleCI CodeDeploy y with para crear revisiones de aplicaciones, cargarlas en un bucket de Amazon S3 y, a continuación, iniciar y supervisar las implementaciones.

Más información:

- [Uso de un CircleCI Orb para implementar aplicaciones en AWS CodeDeploy](#)

CloudBees

Puede utilizar el complemento de CodeDeploy y Jenkins, disponible en DEV @cloud, como acción posterior a la creación. [CloudBees](#)

Por ejemplo, puede utilizarlo al final de una canalización de entrega continua para implementar una revisión de aplicación en la flota de servidores.

Más información:

- [CodeDeploy El plugin de Jenkins ahora está disponible en DEV @cloud](#)

Codship

Puedes usar [Codship](#) para implementar revisiones de aplicaciones a través de él. CodeDeploy Puedes usar la interfaz de usuario de Codship para añadirla a la canalización CodeDeploy de despliegue de una sucursal.

Más información:

- [Implemente en CodeDeploy](#)
- [CodeDeploy integración en Codship](#)

GitHub

Se puede utilizar CodeDeploy para implementar revisiones de aplicaciones desde [GitHub](#) los repositorios. También puede activar un despliegue desde un GitHub repositorio siempre que se modifique el código fuente de ese repositorio.

Más información:

- [Integración CodeDeploy con GitHub](#)
- [Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub](#)
- [Implemente automáticamente a partir del GitHub uso CodeDeploy](#)

HashiCorp Cónsul

Puede utilizar la herramienta HashiCorp Consul de código abierto para garantizar el buen estado y la estabilidad de su entorno de aplicaciones al implementar aplicaciones en él. CodeDeploy Utilice Consul para registrar las aplicaciones a descubrir durante la implementación, cambiar las aplicaciones y los nodos a modo de mantenimiento para que las implementaciones los omitan y detener las implementaciones en caso de que las instancias de destino no funcionen correctamente.

Más información:

- [CodeDeploy despliegues con Consul HashiCorp](#)

Jenkins

El complemento de CodeDeploy [Jenkins](#) proporciona un paso posterior a la compilación para su proyecto de Jenkins. Después de finalizar una compilación correctamente, comprime el espacio de trabajo, carga en Amazon S3 y comienza una nueva implementación.

Más información:

- [CodeDeployEl complemento de Jenkins](#)
- [Configuración del complemento de Jenkins para CodeDeploy](#)

Puppet Labs

AWS proporciona plantillas de muestra para [Puppet](#) y CodeDeploy. El primero es un módulo de Puppet que instala e inicia el agente. CodeDeploy Esto le permite continuar administrando su infraestructura de host con Puppet mientras la usa. CodeDeploy La segunda plantilla de ejemplo muestra cómo CodeDeploy y organizar la ejecución de módulos y manifiestos con una marioneta sin maestros en cada nodo.

Más información:

- [Marioneta y CodeDeploy](#)

SaltStack

Puede integrar la [SaltStack](#) infraestructura con CodeDeploy. Puedes usar el CodeDeploy módulo para instalar y ejecutar el CodeDeploy y agente en tus minions o, con un par de sencillos enlaces de despliegue, puedes usarlo CodeDeploy para organizar el funcionamiento de tus Salt States.

Más información:

- [SaltStack y CodeDeploy](#)

TeamCity	<p>Puede usar el complemento CodeDeploy y Runner para implementar aplicaciones directamente desde TeamCity. El complemento añade un paso de TeamCity compilación que prepara y carga una revisión de la aplicación en un bucket de Amazon S3, registra la revisión en una CodeDeploy aplicación, crea una CodeDeploy implementación y, si así lo desea, espera a que se complete la implementación.</p> <p>Más información:</p> <ul style="list-style-type: none">• CodeDeploy Runner (descargar)• CodeDeploy Complemento Runner (documentación)
Travis CI	<p>Puede configurar Travis CI para que active una implementación CodeDeploy después de una compilación correcta.</p> <p>Más información:</p> <ul style="list-style-type: none">• Travis CI y despliegues CodeDeploy

Temas

- [Integración CodeDeploy con GitHub](#)

Integración CodeDeploy con GitHub

CodeDeploy supports [GitHub](#), un servicio de alojamiento e intercambio de código basado en la web. CodeDeploy puede implementar en las instancias revisiones de aplicaciones almacenadas en GitHub repositorios o buckets de Amazon S3. CodeDeploy solo es compatible con GitHub implementaciones locales de EC2.

Temas

- [Implementación de revisiones desde CodeDeploy GitHub](#)

- [GitHub comportamientos con CodeDeploy](#)

Implementación de revisiones desde CodeDeploy GitHub

Para implementar una revisión de una aplicación desde un GitHub repositorio a las instancias:

1. Cree una revisión que sea compatible con CodeDeploy el tipo de instancia de Amazon EC2 en el que realizará la implementación.

Para crear una revisión compatible, siga las instrucciones indicadas en [Planifique una revisión para CodeDeploy](#) y [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).

2. Utilice una GitHub cuenta para añadir la revisión a un GitHub repositorio.

Para crear una GitHub cuenta, consulte [Unirse GitHub](#). Para crear un GitHub repositorio, consulta [Crear un repositorio](#).

3. Utilice la página Crear despliegue de la CodeDeploy consola o el AWS CLI create-deployment comando para implementar la revisión desde el GitHub repositorio en las instancias de destino configuradas para su uso en CodeDeploy las implementaciones.

Si desea create-deployment ejecutar el comando, primero debe usar la página Crear despliegue de la consola para dar CodeDeploy permiso para interactuar GitHub en nombre de la GitHub cuenta que prefiera para la aplicación especificada. Solamente es necesario hacerlo una vez para cada aplicación.

Para obtener información sobre cómo utilizar la página Crear despliegue para realizar un despliegue desde un GitHub repositorio, consulte [Cree una implementación con CodeDeploy](#).

Para obtener información sobre cómo create-deployment ejecutar el comando para implementar desde un GitHub repositorio, consulte [Creación de una implementación en una plataforma de informática EC2/en las instalaciones \(CLI\)](#).

Para obtener información sobre cómo preparar las instancias para su uso en CodeDeploy las implementaciones, consulte [Trabajar con instancias para CodeDeploy](#).

Para obtener más información, consulte [Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub](#).

GitHub comportamientos con CodeDeploy

Temas

- [GitHub autenticación con aplicaciones en CodeDeploy](#)
- [CodeDeploy interacción con repositorios públicos GitHub y privados](#)
- [CodeDeploy interacción con los repositorios gestionados por la organización GitHub](#)
- [Implemente automáticamente desde CodePipeline con CodeDeploy](#)

GitHub autenticación con aplicaciones en CodeDeploy

Tras conceder el CodeDeploy permiso para interactuar con ella GitHub, la asociación entre esa GitHub cuenta y la aplicación se almacena en CodeDeploy. Puede vincular la aplicación a una GitHub cuenta diferente. También puedes revocar el permiso CodeDeploy para interactuar con GitHub ella.

Para vincular una GitHub cuenta a una aplicación en CodeDeploy

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. Elija la aplicación que desee vincular a una GitHub cuenta diferente.
4. Si la aplicación no tiene un grupo de implementación, elija Crear grupo de implementación para crear uno. Para obtener más información, consulte [Cree un grupo de despliegue con CodeDeploy](#). Se requiere un grupo de implementaciones para elegir Create deployment (Crear implementación) en el paso siguiente.
5. En Deployments (Implementaciones), elija Create deployment (Crear implementación).

Note

No es necesario crear una nueva implementación. Actualmente, esta es la única forma de vincular una GitHub cuenta diferente a una aplicación.

6. En Configuración de despliegue, en Tipo de revisión, seleccione Mi aplicación está almacenada en GitHub.
7. Realice una de las siguientes acciones siguientes:
 - Para crear una conexión de AWS CodeDeploy aplicaciones a una GitHub cuenta, cierre sesión GitHub en una pestaña independiente del navegador web. En nombre de GitHub token, escriba un nombre para identificar esta conexión y, a continuación, elija Conectar a GitHub. La página web le pide que autorice CodeDeploy la interacción con GitHub su aplicación. Continúe en el paso 10.
 - Para usar una conexión que ya haya creado, en el nombre del GitHub token, seleccione su nombre y, a continuación, elija Conectar a GitHub. Continúe en el paso 8.
 - Para crear una conexión a otra GitHub cuenta, cierra sesión GitHub en otra pestaña del navegador web. En nombre de GitHub token, escriba un nombre para identificar la conexión y, a continuación, elija Conectar a GitHub. La página web le pide que autorice CodeDeploy la interacción con GitHub su aplicación. Continúe en el paso 10.
8. Si aún no ha iniciado sesión GitHub, siga las instrucciones de la página de inicio de sesión para iniciar sesión con la GitHub cuenta a la que desee vincular la aplicación.
9. Selecciona Autorizar la aplicación. GitHub da CodeDeploy permiso para interactuar GitHub en nombre de la GitHub cuenta en la que se ha iniciado sesión en la aplicación seleccionada.
10. Si no desea crear una implementación, elija Cancel.

Para revocar el permiso para interactuar con CodeDeploy GitHub

1. Inicie sesión para [GitHub](#) usar las credenciales de la GitHub cuenta en la que desea revocar el permiso AWS CodeDeploy .
2. Abra la página de GitHub [solicitudes](#), busque CodeDeploy en la lista de aplicaciones autorizadas y, a continuación, siga el GitHub procedimiento para revocar la autorización de una solicitud.

CodeDeploy interacción con repositorios públicos GitHub y privados

CodeDeploy admite el despliegue de aplicaciones desde GitHub repositorios públicos y privados. Cuando des CodeDeploy permiso para acceder GitHub en tu nombre, CodeDeploy tendrás acceso de lectura y escritura a todos los GitHub repositorios privados a los que tenga acceso tu GitHub cuenta. Sin embargo, CodeDeploy solo lee de los repositorios. GitHub No escribirá en ninguno de tus GitHub repositorios privados.

CodeDeploy interacción con los repositorios gestionados por la organización GitHub

De forma predeterminada, GitHub los repositorios gestionados por una organización (a diferencia de los repositorios privados o públicos de tu cuenta) no permiten el acceso a aplicaciones de terceros, incluidas las siguientes: CodeDeploy. La implementación fallará si las restricciones de aplicaciones de terceros de una organización están habilitadas en GitHub e intentas implementar código desde su GitHub repositorio. Existen dos maneras de resolver este problema.

- Como miembro de la organización, puedes pedirle al propietario de la organización que apruebe el acceso a CodeDeploy. Los pasos para solicitar este acceso dependen de si ya has autorizado CodeDeploy tu cuenta individual:
 - Si has autorizado el acceso a CodeDeploy tu cuenta, consulta [Cómo solicitar la aprobación de la organización para tus solicitudes autorizadas](#).
 - Si aún no has autorizado el acceso a CodeDeploy tu cuenta, consulta [Cómo solicitar la aprobación de una organización para aplicaciones de terceros](#).
- El responsable de la organización puede deshabilitar todas las restricciones a las aplicaciones de terceros. Para obtener información al respecto, consulte [Desactivación de las restricciones a las aplicaciones de terceros en una organización](#).

Para obtener más información, consulte [Acerca de las restricciones a las aplicaciones de terceros](#).

Implemente automáticamente desde CodePipeline con CodeDeploy

Puede activar un despliegue desde a CodePipeline cada vez que cambie el código fuente. Para obtener más información, consulte [CodePipeline](#).

Ejemplos de integración de la comunidad

En las siguientes secciones se incluyen enlaces a entradas de blogs, artículos y ejemplos proporcionados en la comunidad.

Note

Estos enlaces se ofrecen solo con fines informativos y no deben considerarse una lista completa o una anotación del contenido de los ejemplos. AWS no se hace responsable del contenido ni de la exactitud del contenido externo.

Publicaciones de blog

- [Automatizar CodeDeploy el aprovisionamiento en AWS CloudFormation](#)

Aprenda a aprovisionar el despliegue de una aplicación CodeDeploy mediante el uso de. AWS CloudFormation

Fecha de publicación: enero de 2016

- [AWS Toolkit for Eclipse Integración con CodeDeploy \(parte 1\)](#)

[AWS Toolkit for Eclipse Integración con CodeDeploy \(parte 2\)](#)

[AWS Toolkit for Eclipse Integración con CodeDeploy \(parte 3\)](#)

Descubra cómo los desarrolladores de Java pueden usar el CodeDeploy complemento de Eclipse para implementar aplicaciones web AWS directamente desde los entornos de desarrollo de Eclipse.

Fecha de publicación: febrero de 2015

- [Implemente automáticamente a partir del GitHub uso CodeDeploy](#)

Descubra cómo se CodeDeploy pueden utilizar las implementaciones automáticas desde GitHub hasta para crear una end-to-end canalización, desde el control de código fuente hasta sus entornos de prueba o producción.

Fecha de publicación: diciembre de 2014

CodeDeploy tutoriales

En esta sección se incluyen algunos tutoriales que le ayudarán a aprender a usarlo. CodeDeploy

Los procedimientos de estos tutoriales proporcionan sugerencias sobre la ubicación en la que se deben almacenar los archivos (por ejemplo, c:\temp) y los nombres que se deben dar a los depósitos, subcarpetas o archivos (por ejemplo, codedeploydemobucket y CodeDeployDemo -EC2-Trust.json HelloWorldApp, respectivamente), pero no es obligatorio que los utilice. Solo tiene que recordar sustituir sus nombres y ubicaciones de archivos a medida que realiza los procedimientos.

Temas

- [Tutorial: Implementación WordPress en una instancia de Amazon EC2 \(Amazon Linux o Red Hat Enterprise Linux y Linux, macOS o Unix\)](#)
- [Tutorial: Implementación de una aplicación "Hello, World!" aplicación con CodeDeploy \(Windows Server\)](#)
- [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#)
- [Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling](#)
- [Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub](#)
- [Tutorial: Implementación de una aplicación en Amazon ECS](#)
- [Tutorial: Implementación de un servicio de Amazon ECS con una prueba de validación](#)
- [Tutorial: Implemente una función Lambda actualizada con CodeDeploy y el modelo de aplicaciones AWS sin servidor](#)

Tutorial: Implementación WordPress en una instancia de Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux y Linux, macOS o Unix)

En este tutorial WordPress, implementará una herramienta de blogueo de código abierto y un sistema de administración de contenido basado en PHP y MySQL en una sola instancia de Amazon EC2 que ejecute Amazon Linux o Red Hat Enterprise Linux (RHEL).

¿No es lo que está buscando?

- Para practicar la implementación en su lugar en una instancia de Amazon EC2 con Windows Server, consulte [Tutorial: Implementación de una aplicación "Hello, World!" aplicación con CodeDeploy \(Windows Server\)](#).
- Para practicar la implementación en una instancia en las instalaciones en lugar de en una instancia de Amazon EC2, consulte [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#).

Los pasos de este tutorial se presentan desde la perspectiva de un equipo de desarrollo local con Linux, macOS o Unix. Aunque puede completar la mayoría de los pasos en un equipo local con Windows, debe adaptar los pasos que utilizan comandos como `chmod` y `wget`, las aplicaciones, como `sed`, y las rutas de directorio, como `/tmp`.

Antes de empezar este tutorial, debe completar los requisitos previos indicados en [Empezar con CodeDeploy](#). Estas incluyen configurar un usuario, instalarlo o actualizarlo y crear un perfil de instancia de IAM y un rol de servicio. AWS CLI

Temas

- [Paso 1: Lanzar y configurar una instancia de Amazon EC2 de Amazon Linux o Red Hat Enterprise Linux](#)
- [Paso 2: Configurar el contenido fuente para implementarlo en la instancia de Amazon EC2 de Amazon Linux o Red Hat Enterprise Linux](#)
- [Paso 3: Cargue WordPress la aplicación en Amazon S3](#)
- [Paso 4: Implemente su WordPress aplicación](#)
- [Paso 5: Actualizar y volver a implementar la aplicación WordPress](#)
- [Paso 6: Limpiar WordPress la aplicación y los recursos relacionados](#)

Paso 1: Lanzar y configurar una instancia de Amazon EC2 de Amazon Linux o Red Hat Enterprise Linux

Para implementar la WordPress aplicación CodeDeploy, necesitará una instancia de Amazon EC2 que ejecute Amazon Linux o Red Hat Enterprise Linux (RHEL). La instancia de Amazon EC2 requiere una nueva regla de seguridad de entrada que permita las conexiones HTTP. Esta regla es necesaria para poder ver la WordPress página en un navegador una vez que se haya implementado correctamente.

Siga las instrucciones en [Cree una instancia de Amazon EC2 para CodeDeploy](#). Al llegar a las instrucciones sobre la asignación de una etiqueta de instancia de Amazon EC2 a la instancia, asegúrese de especificar la clave de etiqueta de **Name** y el valor de etiqueta de **CodeDeployDemo**. (Si especifica otra clave u otro valor de etiqueta, las instrucciones de [Paso 4: Implemente su WordPress aplicación](#) pueden producir resultados inesperados).

Una vez completadas las instrucciones para lanzar la instancia de Amazon EC2, vuelva a esta página y continúe con la sección siguiente. No continúe en [Cree una aplicación con CodeDeploy](#) como paso siguiente.

Conexión a una instancia de Amazon EC2 de Amazon Linux o RHEL

Una vez lanzada la nueva instancia de Amazon EC2, siga estas instrucciones para practicar la conexión a la instancia.

1. Utilice el comando `ssh` (o un emulador de terminal con la función SSH, como [PuTTY](#)) para conectar con la instancia de Amazon EC2 en Amazon Linux o RHEL. Necesitará la dirección DNS pública de la instancia y la clave privada del par de claves que utilizó al iniciar la instancia de Amazon EC2. Para obtener más información, consulte [Conectar con la instancia](#).

Por ejemplo, si la dirección DNS pública es

ec2-01-234-567-890.compute-1.amazonaws.com y el par de claves de instancia de Amazon EC2 para el acceso SSH es **codedeploydemo.pem**, escriba lo siguiente:

```
ssh -i /path/to/codedeploydemo.pem ec2-  
user@ec2-01-234-567-890.compute-1.amazonaws.com
```

Sustituya `/path/to/codedeploydemo.pem` por la ruta del archivo `.pem` y la dirección DNS de ejemplo por la dirección de la instancia de Amazon EC2 en Amazon Linux o RHEL.

Note

Si aparece un mensaje de error indicando que los permisos del archivo de clave son demasiado abiertos, deberá restringirlos de modo que solo tenga acceso el usuario actual (usted). Por ejemplo, con el comando `chmod` en Linux, macOS o Unix, escriba:

```
chmod 400 /path/to/codedeploydemo.pem
```

- Una vez que haya iniciado sesión, verá el rótulo de AMI para la instancia de Amazon EC2. Para Amazon Linux, debería tener el siguiente aspecto:

```

  _|  _|_ )
  _| (    /  Amazon Linux AMI
  _|\_/_|_/_|

```

- Ahora ya puede cerrar la sesión en la instancia de Amazon EC2 en ejecución.

Warning

No interrumpa ni termine la instancia de Amazon EC2. De lo contrario, CodeDeploy no podrá implementarla.

Adición de una regla de entrada que permite el tráfico HTTP hacia la instancia de Amazon EC2 de Amazon Linux o RHEL

El siguiente paso confirma que la instancia de Amazon EC2 tiene un puerto HTTP abierto para que pueda ver la página principal de la WordPress aplicación desplegada en un navegador.

- [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
- Elija Instancias y, luego, elija su instancia.
- En la pestaña Descripción, en Grupos de seguridad, elija ver reglas de entrada.

Debe aparecer una lista de reglas en el grupo de seguridad con un aspecto similar al siguiente:

```

Security Groups associated with i-1234567890abcdef0
Ports      Protocol    Source      launch-wizard-N
22         tcp        0.0.0.0/0   #

```

- En Grupos de seguridad, elija el grupo de seguridad de la instancia de Amazon EC2. Podría llamarse **launch-wizard-*N***. La ***N*** del nombre es un número asignado al grupo de seguridad cuando se creó la instancia.

Elija la pestaña Inbound (Reglas entrantes). El grupo de seguridad de la instancia está configurado correctamente si se ve una regla con los siguientes valores:

- Tipo: HTTP
 - Protocolo: TCP
 - Rango de puertos: 80
 - Fuente: 0.0.0.0/0
5. Si no ve ninguna regla con estos valores, utilice los procedimientos de [Agregar reglas a un grupo de seguridad](#) para agregarlas a una nueva regla de seguridad.

Paso 2: Configurar el contenido fuente para implementarlo en la instancia de Amazon EC2 de Amazon Linux o Red Hat Enterprise Linux

Ha llegado el momento de configurar el contenido de origen de su aplicación para tener algo que implementar en la instancia.

Temas

- [Obtener el código fuente](#)
- [Crear scripts para ejecutar la aplicación](#)
- [Añadir un archivo de especificación de la aplicación](#)

Obtener el código fuente

Para este tutorial, debe implementar la plataforma de publicación de WordPress contenido desde su máquina de desarrollo en la instancia Amazon EC2 de destino. Para obtener el código WordPress fuente, puede utilizar las llamadas de línea de comandos integradas. O bien, si tiene instalado Git en su equipo de desarrollo, puede utilizarlo en su lugar.

Para estos pasos, suponemos que has descargado una copia del código WordPress fuente en el /tmp directorio de tu máquina de desarrollo. (Puede elegir cualquier directorio que desee, pero no olvide sustituir la ubicación por /tmp siempre que se especifique en estos pasos).

Elige una de las dos opciones siguientes para copiar los archivos WordPress fuente a tu máquina de desarrollo. La primera opción utiliza llamadas a línea de comandos integradas. La segunda opción utiliza Git.

Temas

- [Para obtener una copia del código WordPress fuente \(llamadas de línea de comandos integradas\)](#)

- [Para obtener una copia del código WordPress fuente \(Git\)](#)

Para obtener una copia del código WordPress fuente (llamadas de línea de comandos integradas)

1. Ejecute el `wget` comando para descargar una copia del código WordPress fuente, en forma de `archivo.zip`, en el directorio actual:

```
wget https://github.com/WordPress/WordPress/archive/master.zip
```

2. Ejecute los comandos `unzip`, `mkdir`, `cp` y `rm` para:

- Descomprimir el archivo `.zip master` en el directorio (carpeta) `/tmp/WordPress_Temp`.
- Copiar el contenido descomprimido en la carpeta de destino `/tmp/WordPress`.
- Eliminar la carpeta temporal `/tmp/WordPress_Temp` y el archivo `master`.

Ejecute los comandos de uno en uno:

```
unzip master -d /tmp/WordPress_Temp
```

```
mkdir -p /tmp/WordPress
```

```
cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
```

```
rm -rf /tmp/WordPress_Temp
```

```
rm -f master
```

Esto deja un conjunto limpio de archivos de código WordPress fuente en la `/tmp/WordPress` carpeta.

Para obtener una copia del código WordPress fuente (Git)

1. Descargue e instale [Git](#) en el equipo de desarrollo.
2. En la carpeta `/tmp/WordPress`, ejecute el comando `git init`.

3. Ejecuta el git clone comando para clonar el WordPress repositorio público y haz tu propia copia del mismo en la carpeta de /tmp/WordPress destino:

```
git clone https://github.com/WordPress/WordPress.git /tmp/WordPress
```

Esto deja un conjunto limpio de archivos de código WordPress fuente en la /tmp/WordPress carpeta.

Crear scripts para ejecutar la aplicación

A continuación, cree una carpeta y scripts en el directorio. CodeDeploy utiliza estos scripts para configurar e implementar la revisión de la aplicación en la instancia Amazon EC2 de destino. Puede utilizar cualquier editor de texto para crear los scripts.

1. Cree un directorio de scripts en su copia del código WordPress fuente:

```
mkdir -p /tmp/WordPress/scripts
```

2. Cree un archivo `install_dependencies.sh` en `/tmp/WordPress/scripts`. Añada las líneas siguientes al archivo. Este script `install_dependencies.sh` instala Apache, MySQL y PHP. También añada compatibilidad con MySQL a PHP.

```
#!/bin/bash
sudo amazon-linux-extras install php7.4
sudo yum install -y httpd mariadb-server php
```

3. Cree un archivo `start_server.sh` en `/tmp/WordPress/scripts`. Añada las líneas siguientes al archivo. Este script `start_server.sh` inicia Apache y MySQL.

```
#!/bin/bash
systemctl start mariadb.service
systemctl start httpd.service
systemctl start php-fpm.service
```

4. Cree un archivo `stop_server.sh` en `/tmp/WordPress/scripts`. Añada las líneas siguientes al archivo. Este script `stop_server.sh` detiene Apache y MySQL.

```
#!/bin/bash
isExistApp=pgrep httpd
if [[ -n $isExistApp ]]; then
systemctl stop httpd.service
fi
isExistApp=pgrep mysqld
if [[ -n $isExistApp ]]; then
systemctl stop mariadb.service
fi
isExistApp=pgrep php-fpm
if [[ -n $isExistApp ]]; then
systemctl stop php-fpm.service

fi
```

5. Cree un archivo `create_test_db.sh` en `/tmp/WordPress/scripts`. Añada las líneas siguientes al archivo. Este `create_test_db.sh` script usa MySQL para crear una **test** base de datos WordPress para su uso.

```
#!/bin/bash
mysql -uroot <<CREATE_TEST_DB
CREATE DATABASE IF NOT EXISTS test;
CREATE_TEST_DB
```

6. Por último, cree un script `change_permissions.sh` en `/tmp/WordPress/scripts`. Esto se utiliza para cambiar los permisos de carpeta en Apache.

Important

Este script ha actualizado los permisos de la carpeta `/tmp/WordPress` de modo que cualquiera pueda escribir en ella. Esto es necesario para WordPress poder escribir en su base de datos durante el proceso [Paso 5: Actualizar y volver a implementar la aplicación WordPress](#). Una vez configurada la WordPress aplicación, ejecute el siguiente comando para actualizar los permisos a una configuración más segura:

```
chmod -R 755 /var/www/html/WordPress
```

```
#!/bin/bash
chmod -R 777 /var/www/html/WordPress
```

7. Conceda permisos ejecutables a todos los scripts. En la línea de comando, escriba:

```
chmod +x /tmp/WordPress/scripts/*
```

Añadir un archivo de especificación de la aplicación

A continuación, añada un archivo de especificaciones de la aplicación (AppSpec archivo), un archivo con formato [YAML](#) que se utiliza para: CodeDeploy

- Asignar los archivos de origen de la revisión de la aplicación a sus destinos en la instancia de Amazon EC2 de destino.
- Especificar permisos personalizados para los archivos implementados.
- Especificar los scripts que se van a ejecutar en la instancia de Amazon EC2 de destino durante la implementación.

El AppSpec archivo debe tener un nombre. `appspec.yml` Debe colocarse en el directorio raíz del código fuente de la aplicación. En este tutorial, el directorio raíz es `/tmp/WordPress`.

Mediante su editor de texto, cree un archivo denominado `appspec.yml`. Añada las líneas siguientes al archivo:

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
```

```
runas: root
ApplicationStart:
- location: scripts/start_server.sh
- location: scripts/create_test_db.sh
  timeout: 300
  runas: root
ApplicationStop:
- location: scripts/stop_server.sh
  timeout: 300
  runas: root
```

CodeDeploy utiliza este AppSpec archivo para copiar todos los archivos de la `/tmp/WordPress` carpeta de la máquina de desarrollo a la `/var/www/html/WordPress` carpeta de la instancia Amazon EC2 de destino. Durante la implementación, CodeDeploy ejecuta los scripts especificados como `root` en la `/var/www/html/WordPress/scripts` carpeta de la instancia Amazon EC2 de destino en eventos específicos durante el ciclo de vida de la implementación, como **BeforeInstall** y **AfterInstall**. Si alguno de estos scripts tarda más de 300 segundos (5 minutos) en ejecutarse, CodeDeploy detiene la implementación y marca la implementación como fallida.

Para obtener más información sobre la configuración, consulte [CodeDeploy AppSpec referencia de archivo](#).

Important

Las ubicaciones y el número de espacios entre cada uno de los elementos de este archivo son importantes. Si el espaciado es incorrecto, se produce CodeDeploy un error que puede ser difícil de depurar. Para obtener más información, consulte [AppSpec Espaciado de archivos](#).

Paso 3: Cargue WordPress la aplicación en Amazon S3

Ahora preparará y cargará el contenido de origen en una ubicación desde la que CodeDeploy pueda desplegarlo. Las siguientes instrucciones explican cómo aprovisionar un bucket de Amazon S3, preparar los archivos de la revisión de la aplicación para el bucket, agrupar los archivos de la revisión y, a continuación, enviar la revisión al bucket.

Note

Aunque no se trata en este tutorial, puede utilizarla CodeDeploy para implementar aplicaciones desde GitHub los repositorios hasta las instancias. Para obtener más información, consulte [Integración CodeDeploy con GitHub](#).

Temas

- [Aprovisionar un bucket de Amazon S3](#)
- [Preparación de los archivos de la aplicación para el bucket](#)
- [Agrupación de los archivos de la aplicación en un único archivo de almacenamiento y envío del archivo de almacenamiento](#)

Aprovisionar un bucket de Amazon S3

Cree un contenedor de almacenamiento o bucket en Amazon S3 o utilice un bucket existente. Asegúrese de que puede cargar la revisión en el bucket y que las instancias Amazon EC2 utilizadas en las implementaciones pueden descargar la revisión del bucket.

Puede utilizar la AWS CLI consola de Amazon S3 o las API de Amazon S3 para crear un bucket de Amazon S3. Después de crear el bucket, asegúrese de que concede permisos de acceso al bucket y a su cuenta de AWS .

Note

Los nombres de los buckets deben ser únicos en Amazon S3 para todas AWS las cuentas. Si no puede utilizar **codedeploydemobucket**, pruebe otro nombre de bucket, por ejemplo, **codedeploydemobucket** seguido de un guion y sus iniciales o algún otro identificador único. A continuación, cuando aparezca **codedeploydemobucket** en este tutorial, asegúrese de sustituirlo por el nombre de su bucket.

El bucket de Amazon S3 debe crearse en la misma AWS región en la que se lanzan las instancias de Amazon EC2 de destino. Por ejemplo, si crea el bucket en la región Este de EE. UU. (Norte de Virginia), sus instancias de Amazon EC2 de destino deben lanzarse en la región Este de EE. UU. (Norte de Virginia).

Temas

- [Para crear un bucket de Amazon S3 \(CLI\)](#)
- [Para crear un bucket de Amazon S3 \(consola\)](#)
- [Otorgue permisos al bucket y la AWS cuenta de Amazon S3](#)

Para crear un bucket de Amazon S3 (CLI)

Llame al comando mb para crear un bucket de Amazon S3 denominado **codedeploydemobucket**:

```
aws s3 mb s3://codedeploydemobucket --region region
```

Para crear un bucket de Amazon S3 (consola)

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
2. En la consola de Amazon S3, elija Crear bucket.
3. En el cuadro Bucket name (Nombre del bucket), escriba un nombre para el bucket.
4. En la lista Region (Región), elija la región de destino y, a continuación, seleccione Create (Crear).

Otorgue permisos al bucket y la AWS cuenta de Amazon S3

Debe tener permisos para cargar en el bucket de Amazon S3. Puede especificar estos permisos mediante una política de bucket de Amazon S3. Por ejemplo, en la siguiente política de bucket de Amazon S3, el uso del carácter comodín (*) permite 111122223333 a la AWS cuenta cargar archivos en cualquier directorio del bucket de Amazon S3 denominado codedeploydemobucket:

```
{
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::codedeploydemobucket/*",
      "Principal": {
        "AWS": [
          "111122223333"
        ]
      }
    }
  ]
}
```



```
]
}
```

Para ver el ID de su AWS cuenta, consulte [Cómo encontrar el ID de su AWS cuenta](#).

Ahora es un buen momento para comprobar que el bucket de Amazon S3 permite las solicitudes de descarga de cada instancia de Amazon EC2 participante. Puede especificarlos mediante una política de bucket de Amazon S3. Por ejemplo, en la siguiente política de bucket de Amazon S3, el carácter comodín (*) permite a cualquier instancia de Amazon EC2 asociada a un perfil de instancia de IAM que contenga el ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` descargar desde cualquier directorio al bucket de Amazon S3 llamado `codedeploydemobucket`:

```
{
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::codedeploydemobucket/*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::444455556666:role/CodeDeployDemo"
        ]
      }
    }
  ]
}
```

Para obtener información acerca de cómo generar y adjuntar una política de bucket de Amazon S3, consulte [Ejemplos de política de bucket](#).

Para obtener información acerca de cómo crear y adjuntar una política de IAM, consulte [Trabajo con políticas](#).

Preparación de los archivos de la aplicación para el bucket

Asegúrese de que los archivos de la WordPress aplicación, el AppSpec archivo y los scripts estén organizados en su máquina de desarrollo de forma similar a la siguiente:

```
/tmp/
```

```
|--WordPress/  
  |-- appspec.yml  
  |-- scripts/  
    |-- change_permissions.sh  
    |-- create_test_db.sh  
    |-- install_dependencies.sh  
    |-- start_server.sh  
    |-- stop_server.sh  
  |-- wp-admin/  
    |-- (various files...)  
  |-- wp-content/  
    |-- (various files...)  
  |-- wp-includes/  
    |-- (various files...)  
  |-- index.php  
  |-- license.txt  
  |-- readme.html  
  |-- (various files ending with .php...)
```

Agrupación de los archivos de la aplicación en un único archivo de almacenamiento y envío del archivo de almacenamiento

Agrupe los archivos de la WordPress aplicación y el AppSpec archivo en un archivo de almacenamiento (conocido como revisión de la aplicación).

Note

Es posible que se cobre por almacenar objetos en un bucket y por transferir revisiones de la aplicación dentro y fuera de un bucket. Para obtener más información, consulte [Precios de Amazon S3](#).

1. En el equipo de desarrollo, cambie a la carpeta en la que se almacenan los archivos:

```
cd /tmp/WordPress
```

Note

Si no cambia a esta carpeta, la agrupación del archivo comenzará en su carpeta actual. Por ejemplo, si la carpeta actual es /tmp en lugar de /tmp/WordPress, la agrupación

comenzará con los archivos y subcarpetas de la carpeta `tmp`, lo que puede incluir algo más que la subcarpeta `WordPress`.

2. Llame al comando `create-application` para registrar una aplicación nueva denominada **WordPress_App**.

```
aws deploy create-application --application-name WordPress_App
```

3. Ejecute el comando CodeDeploy [push](#) para agrupar los archivos, cargar las revisiones en Amazon S3 y registrar la información CodeDeploy sobre la revisión cargada, todo en una sola acción.

```
aws deploy push \  
  --application-name WordPress_App \  
  --s3-location s3://codedeploydemobucket/WordPressApp.zip \  
  --ignore-hidden-files
```

Este comando agrupa los archivos del directorio actual (excluyendo los archivos ocultos) en un único archivo denominado **WordPressApp.zip**, carga la revisión en el **codedeploydemobucket** depósito y registra la información CodeDeploy sobre la revisión cargada.

Paso 4: Implemente su WordPress aplicación

Ahora implementa la revisión de la WordPress aplicación de muestra que cargó en Amazon S3. Puede utilizar la consola AWS CLI o la CodeDeploy consola para implementar la revisión y supervisar el progreso de la implementación. Una vez implementada correctamente la revisión de la aplicación, comprobará los resultados.

Temas

- [Implemente la revisión de su aplicación con CodeDeploy](#)
- [Monitorización y solución de problemas de la implementación](#)
- [Comprobación de la implementación](#)

Implemente la revisión de su aplicación con CodeDeploy

Utilice la consola AWS CLI o la consola para implementar la revisión de la aplicación.

Temas

- [Para implementar la revisión de la aplicación \(CLI\)](#)
- [Para implementar la revisión de la aplicación \(consola\)](#)

Para implementar la revisión de la aplicación (CLI)

1. La implementación necesita un grupo de implementaciones. Sin embargo, para crear el grupo de implementaciones, primero necesita un ARN de rol de servicio. Un rol de servicio es un rol de IAM que concede a un servicio permiso para actuar en su nombre. En este caso, el rol de servicio otorga CodeDeploy permiso para acceder a sus instancias de Amazon EC2 para expandir (leer) sus etiquetas de instancia de Amazon EC2.

Ya debe haber seguido las instrucciones de [Creación de un rol de servicio \(CLI\)](#) para crear un rol de servicio. Para obtener el ARN del rol de servicio, consulte [Obtención del ARN del rol de servicio \(CLI\)](#).

2. Ahora que tiene el ARN del rol de servicio, llame al comando `create-deployment-group` para crear un grupo de implementación denominado **WordPress_DepGroup** asociado a la aplicación **WordPress_App**, utilizando la etiqueta de Amazon EC2 denominada **CodeDeployDemo** y la configuración de implementación **CodeDeployDefault.OneAtATime**:

```
aws deploy create-deployment-group \  
  --application-name WordPress_App \  
  --deployment-group-name WordPress_DepGroup \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --ec2-tag-filters Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE \  
  --service-role-arn serviceRoleARN
```

Note

El [create-deployment-group](#) comando permite crear activadores que provoquen el envío de notificaciones de Amazon SNS a los suscriptores del tema sobre eventos específicos en implementaciones e instancias. El comando también admite opciones para revertir automáticamente las implementaciones y configurar alarmas para detener las implementaciones cuando se cumplen los umbrales de monitoreo de CloudWatch las alarmas de Amazon. Los comandos para estas acciones no se incluyen en este tutorial.

3. Antes de crear una implementación, las instancias de su grupo de implementación deben tener el CodeDeploy agente instalado. Puede instalar el agente desde la línea de comandos con AWS Systems Manager mediante el comando siguiente:

```
aws ssm create-association \  
  --name AWS-ConfigureAWSPackage \  
  --targets Key=tag:Name,Values=CodeDeployDemo \  
  --parameters action=Install,name=AWSCodeDeployAgent \  
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Este comando crea una asociación en Systems Manager State Manager que instalará el CodeDeploy agente e intentará actualizarlo todos los domingos a las 2:00 de la mañana. Para obtener más información sobre el CodeDeploy agente, consulte [Trabajar con el CodeDeploy agente](#). Para obtener más información acerca de Systems Manager, consulte [¿Qué es AWS Systems Manager?](#).

4. Ejecute ahora el comando create-deployment para crear una implementación asociada a la aplicación **WordPress_App**, la configuración de implementación **CodeDeployDefault.OneAtATime** y el grupo de implementaciones **WordPress_DepGroup** utilizando la revisión de la aplicación **WordPressApp.zip** del bucket **codedeploydemobucket**:

```
aws deploy create-deployment \  
  --application-name WordPress_App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DepGroup \  
  --s3-location bucket=codedeploydemobucket,bundleType=zip,key=WordPressApp.zip
```


Para implementar la revisión de la aplicación (consola)

1. Antes de usar la CodeDeploy consola para implementar la revisión de la aplicación, necesita un ARN de rol de servicio. Un rol de servicio es un rol de IAM que concede a un servicio permiso para actuar en su nombre. En este caso, el rol de servicio otorga CodeDeploy permiso para acceder a sus instancias de Amazon EC2 para expandir (leer) sus etiquetas de instancia de Amazon EC2.

Ya debe haber seguido las instrucciones de [Creación de un rol de servicio \(consola\)](#) para crear un rol de servicio. Para obtener el ARN del rol de servicio, consulte [Obtención del ARN del rol de servicio \(consola\)](#).


2. Ahora que tiene el ARN, utilice la CodeDeploy consola para implementar la revisión de la aplicación:

Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

3. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
4. En la lista de aplicaciones, elija WordPress_App.
5. En la pestaña Deployment groups (Grupos de implementaciones), elija Create deployment group (Crear grupo de implementaciones).
6. En Nombre de grupo de implementación, escriba **WordPress_DepGroup**.
7. En Tipo de implementación, elija Implementación local.
8. En Configuración de entorno, elija la pestaña Instancias de Amazon EC2.
9. En Configuración del agente con AWS Systems Manager, mantenga los valores predeterminados.
10. En Clave, introduzca **Name**.
11. En Valor, escriba **CodeDeployDemo**.

 Note

Después de escribir **CodeDeployDemo**, aparecerá un 1 en Instancias coincidentes para confirmar que CodeDeploy se ha encontrado una instancia de Amazon EC2 coincidente.

12. En Configuración de implementación, elija CodeDeployDefault. OneAtA Time.
13. En ARN de rol de servicio, elija el ARN del rol de servicio y seleccione Crear grupo de implementación.
14. Elija Crear implementación.
15. En Grupo de implementación, elija **WordPress_DepGroup**.
16. Junto a Tipo de repositorio, elija Mi aplicación está almacenada en Amazon S3. En Ubicación de revisión, introduzca la ubicación de la revisión de la WordPress aplicación de muestra que cargó anteriormente en Amazon S3. Para obtener la ubicación:

- a. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
- b. En la lista de buckets, elija `codedeploydemobucket` (o el nombre del bucket en el que cargó la revisión de la aplicación).
- c. En la lista de objetos, elija `WordPressApp.zip`.
- d. En la pestaña Overview (Información general), copie el valor del campo Link (Enlace) en el portapapeles.

Debe ser similar al siguiente:

`https://s3.amazonaws.com/codedeploydemobucket/WordPressApp.zip`

- e. Vuelva a la CodeDeploy consola y, en la ubicación de revisión, pegue el valor del campo Enlace.
17. Si en la lista File type (Tipo de archivo) aparece un mensaje en el que se indica que no se ha podido detectar el tipo de archivo, seleccione `.zip`.
 18. (Opcional) Escriba un comentario en el cuadro Deployment description.
 19. Expanda Anulaciones de grupos de despliegues y, en Configuración de despliegue, seleccione `CodeDeployDefault.OneAtUna hora`.
 20. Elija Iniciar la implementación. En la página Deployments se mostrará información sobre la implementación que acaba de crear.

Monitorización y solución de problemas de la implementación

Utilice la consola AWS CLI o la consola para supervisar y solucionar los problemas de la implementación.

Temas

- [Para monitorizar y solucionar problemas en la implementación \(CLI\)](#)
- [Para monitorizar y resolver problemas de la implementación \(consola\)](#)

Para monitorizar y solucionar problemas en la implementación (CLI)

1. Obtenga el ID de la implementación; para ello, ejecute el comando `list-deployments` para la aplicación **`WordPress_App`** y el grupo de implementaciones **`WordPress_DepGroup`**:

```
aws deploy list-deployments --application-name WordPress_App --deployment-group-name WordPress_DepGroup --query 'deployments' --output text
```

2. Llame al comando `get-deployment` con el ID de la implementación:

```
aws deploy get-deployment --deployment-id deploymentID --query 'deploymentInfo.status' --output text
```

3. El comando devuelve el estado general de la implementación. Si todo sale bien, el valor es `Succeeded`.

Si el estado general es `Failed`, puede ejecutar comandos como [list-deployment-instances](#) y [get-deployment-instance](#) para solucionar problemas. Si desea conocer más opciones de solución de problemas, consulte [Análisis de archivos de registro para investigar los errores de implementación en instancias](#).

Para monitorizar y resolver problemas de la implementación (consola)

En la página de implementaciones de la CodeDeploy consola, puede supervisar el estado de la implementación en la columna Estado.

Para obtener más información acerca de la implementación, sobre todo si el valor de la columna Status tiene un valor distinto de `Succeeded`:

1. En la tabla Deployments (Implementaciones), elija el nombre de la implementación. Cuando se produce un error en una implementación, aparece un mensaje que describe el motivo del error.
2. En Actividad de instancia, aparece más información sobre la implementación. Cuando se una implementación falla, es posible que pueda determinar las instancias de Amazon EC2 afectadas y el paso del proceso en el que se ha producido el error.
3. Si desea continuar con la solución de problemas, puede aplicar una técnica como la descrita en [View Instance Details](#). También puede analizar los archivos de registro de la implementación en una instancia de Amazon EC2. Para obtener más información, consulte [Análisis de archivos de registro para investigar los errores de implementación en instancias](#).

Comprobación de la implementación

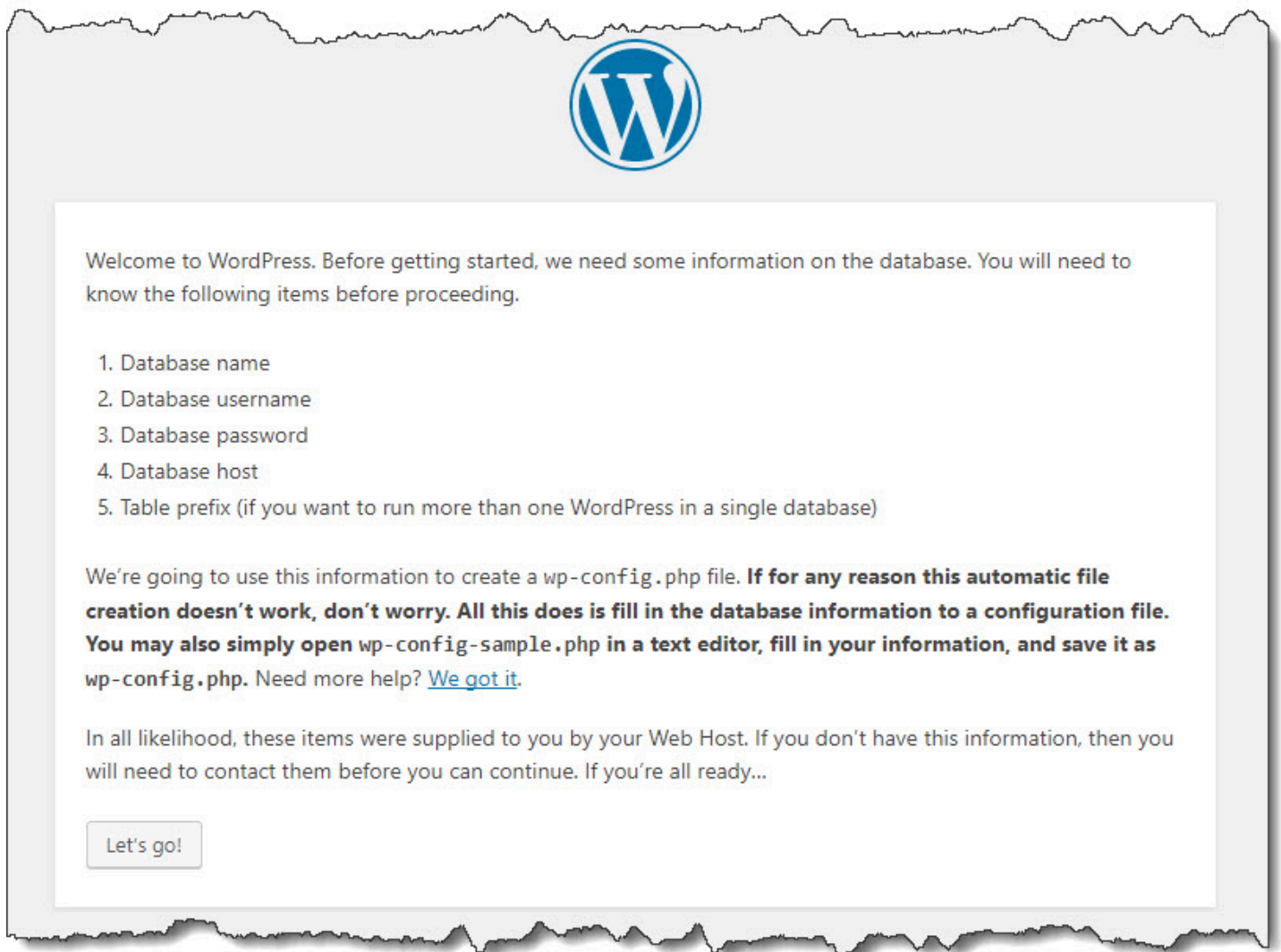
Una vez que la implementación se haya realizado correctamente, compruebe que WordPress la instalación funciona. Utilice la dirección DNS pública de la instancia de Amazon EC2 seguida de /

WordPress, para ver el sitio en un navegador web. (Para obtener el valor de DNS público, en la consola de Amazon EC2, seleccione la instancia de Amazon EC2 y, en la pestaña Descripción, busque el valor Public DNS).

Por ejemplo, si la dirección DNS pública de la instancia de Amazon EC2 es **ec2-01-234-567-890.compute-1.amazonaws.com**, se usará la siguiente dirección URL:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress
```

Al ver el sitio en el navegador, debería ver una página de WordPress bienvenida similar a la siguiente:



Si su instancia de Amazon EC2 no tiene una regla de entrada HTTP agregada a su grupo de seguridad, la página de WordPress bienvenida no aparecerá. Si aparece un mensaje que indica que

el servidor remoto no responde, asegúrese de que el grupo de seguridad de la instancia de Amazon EC2 tiene la regla de entrada. Para obtener más información, consulte [Adición de una regla de entrada que permite el tráfico HTTP hacia la instancia de Amazon EC2 de Amazon Linux o RHEL](#).

Paso 5: Actualizar y volver a implementar la aplicación WordPress

Ahora que has implementado correctamente la revisión de la aplicación, actualiza el WordPress código en la máquina de desarrollo y úsalo CodeDeploy para volver a implementar el sitio. Posteriormente, debería poder ver los cambios en el código en la instancia de Amazon EC2.

Temas

- [Configura el sitio WordPress](#)
- [Modificación del sitio](#)
- [Nueva implementación del sitio](#)

Configura el sitio WordPress

Para ver los efectos del cambio de código, termine de configurar el WordPress sitio para que tenga una instalación completamente funcional.

1. Escriba la URL del sitio web en el navegador web. La URL es la dirección DNS pública de la instancia de Amazon EC2 con una extensión `/WordPress`. Para este WordPress sitio de ejemplo (y ejemplo de la dirección DNS pública de la instancia Amazon EC2), la URL es. **`http://ec2-01-234-567-890.compute-1.amazonaws.com/WordPress`**
2. Si aún no ha configurado el sitio, aparecerá la página de bienvenida WordPress predeterminada. Elija ¡Vamos!.
3. En la página de configuración de la base de datos, escriba los siguientes valores para utilizar la base de datos MySQL predeterminada:
 - Database Name (Nombre de la base de datos): **test**
 - User Name (Nombre del usuario): **root**
 - Contraseña: déjela en blanco.
 - Database Host (Host de base de datos): **localhost**
 - Table Prefix (Prefijo de tabla): **wp_**

Elija Submit para configurar la base de datos.

- Continúe con la configuración del sitio. En la página de bienvenida, rellena los valores que desees y selecciona Instalar WordPress. Cuando se haya completado la instalación, puede iniciar sesión en el panel.

Important

Durante el despliegue de la WordPress aplicación, el **change_permissions.sh** script actualizó los permisos de la `/tmp/WordPress` carpeta para que cualquiera pudiera escribir en ella. Este es un buen momento para ejecutar el siguiente comando para restringir los permisos, de forma que solo usted, el propietario, pueda escribir en ella:

```
chmod -R 755 /var/www/html/WordPress
```

Modificación del sitio

Para modificar el WordPress sitio, vaya a la carpeta de la aplicación en su máquina de desarrollo:

```
cd /tmp/WordPress
```

Para modificar algunos colores del sitio, en el archivo `wp-content/themes/twentyfifteen/style.css`, utilice un editor de texto o `sed` para cambiar `#fff` a `#768331`.

En Linux u otros sistemas con `sed GNU`, utilice:

```
sed -i 's/#fff/#768331/g' wp-content/themes/twentyfifteen/style.css
```

En macOS, Unix u otros sistemas con `sed BSD`, utilice:

```
sed -i '' 's/#fff/#768331/g' wp-content/themes/twentyfifteen/style.css
```

Nueva implementación del sitio

Ahora que ha modificado el código del sitio, utilice Amazon S3 CodeDeploy para volver a implementar el sitio.

Agrupe y cargue los cambios en Amazon S3, como se describe en [Agrupación de los archivos de la aplicación en un único archivo de almacenamiento y envío del archivo de almacenamiento](#). (Cuando siga esas instrucciones, recuerde que no tiene que crear una aplicación). Asigne a la nueva revisión la misma clave que antes (**WordPressApp.zip**). Carguela en el mismo bucket de Amazon S3 que creó anteriormente (por ejemplo, **codedeploydemobucket**).

Utilice la AWS CLI CodeDeploy consola o las CodeDeploy API para volver a implementar el sitio.

Temas

- [Para volver implementar el sitio \(CLI\)](#)
- [Para volver a implementar el sitio \(consola\)](#)

Para volver implementar el sitio (CLI)

Llame al comando `create-deployment` para crear una implementación basada en la revisión recién cargada. Utilice la aplicación **WordPress_App**, la configuración de implementación **CodeDeployDefault.OneAtATime**, el grupo de implementaciones **WordPress_DepGroup** y la revisión **WordPressApp.zip** del bucket **codedeploydemobucket**:

```
aws deploy create-deployment \  
  --application-name WordPress_App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DepGroup \  
  --s3-location bucket=codedeploydemobucket,bundleType=zip,key=WordPressApp.zip
```

Puede comprobar el estado de la implementación como se describe en [Monitorización y solución de problemas de la implementación](#).

Una vez CodeDeploy que haya vuelto a implementar el sitio, vuelva a visitarlo en su navegador web para comprobar que se han cambiado los colores. (Puede que tenga que actualizar el navegador). Si los colores han cambiado, ¡enhorabuena! Ha modificado y vuelto a implementar correctamente su sitio.

Para volver a implementar el sitio (consola)

1. [Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en https://console.aws.amazon.com/codedeploy.](https://console.aws.amazon.com/codedeploy)

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la lista de aplicaciones, elija WordPress_App.
4. En la pestaña Deployment groups (Grupos de implementaciones), elija **WordPress_DepGroup**.
5. Elija Crear implementación.
6. En la página Create deployment:
 - a. En Deployment group (Grupo de implementación), elija **WordPress_DepGroup**.
 - b. En el área Tipo de repositorio, seleccione Mi aplicación está almacenada en Amazon S3 y luego copie el enlace de Amazon S3 de la revisión en el cuadro Ubicación de la revisión. Para encontrar el valor del enlace:
 - i. En una pestaña diferente del navegador:

Inicie sesión en la consola de Amazon S3 AWS Management Console y ábrala en <https://console.aws.amazon.com/s3/>.

Vaya al archivo codedeploydemobucket, ábralo y elija la revisión, **WordPressApp.zip**.
 - ii. Si el panel Propiedades no se muestra en la consola de Amazon S3, elija el botón Propiedades.
 - iii. En el panel de propiedades, copie el valor del campo Enlace en el cuadro Ubicación de la revisión de la CodeDeploy consola.
 - c. Si aparece un mensaje que indica que el tipo de archivo no se pudo detectar, elija .zip.
 - d. Deje el cuadro Deployment description en blanco.
 - e. Expanda Anulaciones de grupos de despliegues y, en Configuración de despliegue, seleccione CodeDeployDefault.OneAtaTime.
 - f. Elija Iniciar la implementación. En la página Deployments se mostrará información sobre la implementación que acaba de crear.
 - g. Puede comprobar el estado de la implementación como se describe en [Monitorización y solución de problemas de la implementación](#).

Una vez CodeDeploy que haya redistribuido el sitio, vuelva a visitarlo en su navegador web para comprobar que se han cambiado los colores. (Puede que tenga que actualizar el navegador). Si los colores han cambiado, ¡enhorabuena! Ha modificado y vuelto a implementar correctamente su sitio.

Paso 6: Limpiar WordPress la aplicación y los recursos relacionados

Ahora has realizado correctamente una actualización del WordPress código y has vuelto a implementar el sitio. Para evitar cargos recurrentes por los recursos que ha creado para este tutorial, debe eliminar:

- Cualquier AWS CloudFormation pila (o cierre cualquier instancia de Amazon EC2, si la creó fuera AWS CloudFormation de ella).
- Todos los buckets de Amazon S3.
- La aplicación WordPress_App en CodeDeploy.
- La asociación de administradores AWS Systems Manager estatales del CodeDeploy agente.

Puede utilizar Amazon S3 AWS CloudFormation, Amazon EC2 y CodeDeploy las consolas o las AWS API para realizar la limpieza. AWS CLI

Temas

- [Para limpiar los recursos \(CLI\)](#)
- [Para limpiar los recursos \(consola\)](#)
- [Sigüientes pasos](#)

Para limpiar los recursos (CLI)

1. Si usó nuestra AWS CloudFormation plantilla para este tutorial, delete-stack ejecute el comando contra la pila nombrada. **CodeDeployDemoStack** Así terminará todas las instancias Amazon EC2 asociadas y eliminará todos los roles de IAM que haya creado la pila:

```
aws cloudformation delete-stack --stack-name CodeDeployDemoStack
```

2. Para eliminar el bucket de Amazon S3, llame al comando `rm` con el modificador `--recursive` para el bucket denominado **codedeploydemobucket**. Así eliminará el bucket y todos los objetos que contiene:

```
aws s3 rm s3://codedeploydemobucket --recursive --region region
```

3. Para eliminar la aplicación `WordPress_App`, realice una llamada al comando `delete-application`. Esto eliminará también todos los registros de grupo de implementaciones asociados y los registros de implementación de la aplicación:

```
aws deploy delete-application --application-name WordPress_App
```

4. Para eliminar la asociación de Systems Manager State Manager, llame al comando `delete-association`.

```
aws ssm delete-association --association-id association-id
```

Para obtener el valor de *id-asociación*, llame al comando `describe-association`.

```
aws ssm describe-association --name AWS-ConfigureAWSPackage --targets  
Key=tag:Name,Values=CodeDeployDemo
```

Si no usó la AWS CloudFormation pila para este tutorial, `terminate-instances` ejecute el comando para terminar cualquier instancia de Amazon EC2 que haya creado manualmente. Indique el ID de la instancia de Amazon EC2 que desea terminar:

```
aws ec2 terminate-instances --instance-ids instanceId
```

Para limpiar los recursos (consola)

Si utilizó nuestra AWS CloudFormation plantilla para este tutorial, elimine la AWS CloudFormation pila asociada.

1. Inicie sesión AWS Management Console y abra la AWS CloudFormation consola en <https://console.aws.amazon.com/cloudformation>.
2. En el cuadro Filtro, escribe el nombre de la AWS CloudFormation pila que creaste anteriormente (por ejemplo, **CodeDeployDemoStack**).

3. Seleccione la casilla junto al nombre de la pila. En el menú Actions (Acciones), elija Delete Stack (Eliminar pila).

AWS CloudFormation elimina la pila, cierra todas las instancias de Amazon EC2 adjuntas y elimina todas las funciones de IAM correspondientes.

Para finalizar las instancias de Amazon EC2 que creó fuera de una AWS CloudFormation pila:

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. En la lista INSTANCES, elija Instances.
3. En el campo de búsqueda, escriba el nombre de la instancia de Amazon EC2 que desee finalizar (por ejemplo, **CodeDeployDemo**) y, a continuación, pulse Intro.
4. Seleccione el nombre de la instancia de Amazon EC2.
5. En el menú Actions, seleccione Instance State y elija Terminate. Cuando se le pregunte, elija Yes, Terminate.

Repita estos pasos para cada instancia.

Para eliminar el bucket de Amazon S3:

1. Inicie sesión en la consola de Amazon S3 AWS Management Console y ábrala en <https://console.aws.amazon.com/s3/>.
2. En la lista de buckets, busque y seleccione el nombre del bucket de Amazon S3 que creó anteriormente (por ejemplo, **codedeploydemobucket**).
3. Para eliminar un bucket, primero debe eliminar su contenido. Seleccione todos los archivos del bucket, como **WordPressApp.zip**. En el menú Actions, elija Delete. Cuando se le pida confirmar la eliminación, elija OK.
4. Una vez vacío, puede eliminar el bucket. En la lista de buckets, elija la línea del bucket (pero no su nombre). Elija Delete bucket y cuando se le pida que confirme, seleccione OK.

Para eliminar la WordPress_App aplicación de CodeDeploy:

1. Inicie sesión AWS Management Console y abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy>.

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la lista de aplicaciones, elija WordPress_App.
4. En la página Application details (Detalles de la aplicación), elija Delete application (Eliminar aplicación).
5. Cuando se le solicite, introduzca el nombre de la aplicación para confirmar que desea eliminarla y seleccione Delete (Eliminar).

Para eliminar la asociación de Systems Manager State Manager:

1. Abra la AWS Systems Manager consola en <https://console.aws.amazon.com/systems-manager>.
2. En el panel de navegación, elija Administrador de estados.
3. Elija la asociación que ha creado y, después, elija Eliminar.

Siguientes pasos

Si ha llegado hasta aquí, ¡enhorabuena! Ha completado correctamente una CodeDeploy implementación y, a continuación, ha actualizado el código de su sitio y lo ha vuelto a implementar.

Tutorial: Implementación de una aplicación "Hello, World!" aplicación con CodeDeploy (Windows Server)

En este tutorial, implementa una única página web en una sola instancia Amazon EC2 de Windows Server que ejecuta Internet Information Services (IIS) como servidor web. Esta página web muestra un mensaje "Hello, World!".

¿No es lo que está buscando?

- Para practicar la implementación en una instancia de Amazon EC2 de Amazon Linux o Red Hat Enterprise Linux (RHEL) en su lugar, consulte [Tutorial: Implementación WordPress en una instancia de Amazon EC2 \(Amazon Linux o Red Hat Enterprise Linux y Linux, macOS o Unix\)](#).

- Para practicar la implementación en una instancia on-premises, consulte [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#).

Los pasos de este tutorial se presentan desde una perspectiva de Windows. Aunque puede completar la mayoría de estos pasos en un equipo local que ejecuta Linux, macOS o Unix, debe adaptar los pasos en los que se usan rutas de directorio basadas en Windows, como `c:\temp`. Además, si desea conectarse a la instancia de Amazon EC2, necesita una aplicación cliente que pueda conectarse mediante el Protocolo de escritorio remoto (RDP) a la instancia de Amazon EC2 que ejecuta Windows Server. (Windows incluye una aplicación cliente de conexión RDP de forma predeterminada).

Antes de empezar este tutorial, debe cumplir los requisitos previos, como configurar el usuario [Empezar con CodeDeploy](#), instalar o actualizar el perfil de instancia de IAM y crear un perfil de instancia de IAM y un rol de servicio. AWS CLI

Temas

- [Paso 1: Lanzar una instancia de Amazon EC2 de Windows Server](#)
- [Paso 2: Configurar el contenido de origen para implementarlo en la instancia de Amazon EC2 de Windows Server](#)
- [Paso 3: Cargar la aplicación "Hello, World!" en Amazon S3](#)
- [Paso 4: Implementar la aplicación Hello World.](#)
- [Paso 5: Actualizar y volver a implementar su aplicación "Hello, World!"](#)
- [Paso 6: Limpiar la aplicación de "Hello, World!" y sus recursos relacionados](#)

Paso 1: Lanzar una instancia de Amazon EC2 de Windows Server

Para implementar la aplicación Hello World CodeDeploy, necesita una instancia de Amazon EC2 que ejecute Windows Server.

Siga las instrucciones en [Cree una instancia de Amazon EC2 para CodeDeploy](#). Cuando esté listo para asignar una etiqueta de instancia de Amazon EC2 a la instancia, asegúrese de especificar la clave de etiqueta **Name** y el valor de etiqueta **CodeDeployDemo**. (Si especifica otra clave u otro valor de etiqueta, las instrucciones de [Paso 4: Implementar la aplicación Hello World.](#) pueden producir resultados inesperados).

Una vez que haya lanzado la instancia de Amazon EC2, vuelva a esta página y continúe con la sección siguiente. No continúe en [Cree una aplicación con CodeDeploy](#) como paso siguiente.

Conexión con la instancia de Amazon EC2

Una vez lanzada la instancia de Amazon EC2, siga estas instrucciones para practicar la conexión a la instancia.

Note

En estas instrucciones se supone que está trabajando con Windows y con la aplicación cliente de conexión a escritorio de Windows. Para obtener más información, consulte [Conexión con la instancia de Windows mediante RDP](#). Es posible que necesite adaptar estas instrucciones para otros sistemas operativos u otras aplicaciones cliente de conexión RDP.

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. En el panel de navegación, bajo Instances, elija Instances.
3. Busque su instancia de Windows Server en la lista y selecciónela.
4. Elija Conectar.
5. Elija Obtener contraseña y, a continuación, elija Seleccionar archivo.
6. Busque y seleccione el archivo de par de claves de la instancia de Amazon EC2 asociado a la instancia Amazon EC2 de Windows Server y, a continuación, elija Abrir.
7. Elija Descifrar contraseña. Apunte la contraseña que se muestra. La necesitará en el paso 10.
8. Elija Download Remote Desktop File (Descargar archivo de escritorio remoto) y abra el archivo.
9. Si se le pide conectarse a pesar de que el editor de la conexión remota no pueda identificarse, hágalo.
10. Escriba la contraseña que ha anotado en el paso 7 y continúe. (Si la aplicación cliente de conexión RDP solicita un nombre de usuario, escriba **Administrator**).
11. Si se le pide que se conecte a pesar de que la identidad del equipo remoto no pueda verificarse, hágalo.
12. Después de haberse conectado, se muestra el escritorio de la instancia de Amazon EC2 que ejecuta Windows Server.
13. Ahora puede desconectar de la instancia de Amazon EC2.

⚠ Warning

No interrumpa ni termine la instancia. De lo contrario, no CodeDeploy podrá realizar la implementación en ella.

Adición de una regla de entrada que permite el tráfico HTTP a la instancia de Amazon EC2 de Windows Server

El siguiente paso confirma que la instancia de Amazon EC2 tiene un puerto HTTP abierto para que se pueda ver la página web implementada en la instancia de Amazon EC2 de Windows Server en un navegador.

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. Elija Instancias y, luego, elija su instancia.
3. En la pestaña Descripción, en Grupos de seguridad, elija ver reglas de entrada.

Debe aparecer una lista de reglas en el grupo de seguridad con un aspecto similar al siguiente:

```
Security Groups associated with i-1234567890abcdef0
Ports      Protocol   Source      launch-wizard-N
22         tcp       0.0.0.0/0   #
```

4. En Grupos de seguridad, elija el grupo de seguridad de la instancia de Amazon EC2. Podría llamarse **launch-wizard-*N***. La ***N*** del nombre es un número asignado al grupo de seguridad cuando se creó la instancia.

Elija la pestaña Inbound (Reglas entrantes). El grupo de seguridad de la instancia está configurado correctamente si se ve una regla con los siguientes valores:

- Tipo: HTTP
 - Protocolo: TCP
 - Rango de puertos: 80
 - Fuente: 0.0.0.0/0
5. Si no ve ninguna regla con estos valores, utilice los procedimientos de [Agregar reglas a un grupo de seguridad](#) para agregarlas a una nueva regla de seguridad.

Paso 2: Configurar el contenido de origen para implementarlo en la instancia de Amazon EC2 de Windows Server

Ha llegado el momento de configurar el contenido de origen de la aplicación para tener algo que implementar en la instancia de Amazon EC2. En este tutorial, implementará una única página web en la instancia de Amazon EC2 que ejecuta Windows Server, que a su vez ejecutará Internet Information Services (IIS) como su servidor web. Esta página web mostrará un mensaje "Hello, World!" simple.

Temas

- [Crear la página web](#)
- [Crear un script para ejecutar la aplicación](#)
- [Añadir un archivo de especificación de la aplicación](#)

Crear la página web

1. Cree un subdirectorio (subcarpeta) denominado HelloWorldApp en su carpeta `c:\temp` y, a continuación, pase a esa carpeta.

```
mkdir c:\temp\HelloWorldApp
cd c:\temp\HelloWorldApp
```

Note

La ubicación de la carpeta no tiene por qué ser `c:\temp` ni el nombre de la subcarpeta, HelloWorldApp. Si utiliza otra ubicación u otro nombre de subcarpeta, asegúrese de utilizarlos en este tutorial.

2. Utilice un editor de texto para crear un archivo dentro de la carpeta. Nombre el archivo `index.html`.

```
notepad index.html
```

3. Añada el siguiente código HTML al archivo y guárdelo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
  <title>Hello, World!</title>
  <style>
    body {
      color: #ffffff;
      background-color: #0188cc;
      font-family: Arial, sans-serif;
      font-size:14px;
    }
  </style>
</head>
<body>
  <div align="center"><h1>Hello, World!</h1></div>
  <div align="center"><h2>You have successfully deployed an application using
CodeDeploy</h2></div>
  <div align="center">
    <p>What to do next? Take a look through the <a href="https://aws.amazon.com/
codedeploy">CodeDeploy Documentation</a>.</p>
  </div>
</body>
</html>
```

Crear un script para ejecutar la aplicación

A continuación, creará un script que CodeDeploy se utilizará para configurar el servidor web en la instancia Amazon EC2 de destino.

1. En la misma subcarpeta donde se guarda el archivo `index.html`, utilice un editor de texto para crear otro archivo. Nombre el archivo `before-install.bat`.

```
notepad before-install.bat
```

2. Añada el siguiente código de script de procesamiento por lotes al archivo y, a continuación, guárdelo.

```
REM Install Internet Information Server (IIS).
c:\Windows\Sysnative\WindowsPowerShell\v1.0\powershell.exe -Command Import-Module -
Name ServerManager
c:\Windows\Sysnative\WindowsPowerShell\v1.0\powershell.exe -Command Install-
WindowsFeature Web-Server
```

Añadir un archivo de especificación de la aplicación

A continuación, añadirá un archivo de especificaciones de la aplicación (AppSpec archivo) además de la página web y el archivo de script por lotes. El AppSpec archivo es un archivo con formato [YAML](#) que se utiliza para: CodeDeploy

- Asignar los archivos de origen de la revisión de la aplicación a sus destinos en la instancia.
- Especificar los scripts que deben ejecutarse en la instancia durante la implementación.

El AppSpec archivo debe tener un nombre. `appspec.yml` Debe ir en la carpeta raíz de código fuente de la aplicación.

1. Utilice un editor de texto para crear otro archivo en la misma carpeta en la que se encuentran `index.html` y `before-install.bat`. Nombre el archivo `appspec.yml`.

```
notepad appspec.yml
```

2. Añada el siguiente código YAML al archivo y guárdelo.

```
version: 0.0
os: windows
files:
  - source: \index.html
    destination: c:\inetpub\wwwroot
hooks:
  BeforeInstall:
    - location: \before-install.bat
      timeout: 900
```

CodeDeploy utilizará este AppSpec archivo para copiar el `index.html` archivo de la carpeta raíz del código fuente de la aplicación a la `c:\inetpub\wwwroot` carpeta de la instancia Amazon EC2 de destino. Durante la implementación, CodeDeploy ejecutará el script `before-install.bat` por lotes en la instancia Amazon EC2 de destino durante el evento del ciclo de vida de la **BeforeInstall** implementación. Si este script tarda más de 900 segundos (15 minutos) en ejecutarse, CodeDeploy detendrá la implementación y marcará la implementación en la instancia de Amazon EC2 como fallida.

Para obtener más información sobre la configuración, consulte [CodeDeploy AppSpec referencia de archivo](#).

Important

Las ubicaciones y el número de espacios entre cada uno de los elementos de este archivo son importantes. Si el espaciado es incorrecto, se CodeDeploy generará un error que puede ser difícil de depurar. Para obtener más información, consulte [AppSpec Espaciado de archivos](#).

Paso 3: Cargar la aplicación "Hello, World!" en Amazon S3

Ahora preparará y cargará su contenido fuente en una ubicación desde la que CodeDeploy pueda desplegarlo. Las siguientes instrucciones explican cómo aprovisionar un bucket de Amazon S3, preparar los archivos de la revisión de la aplicación para el bucket, agrupar los archivos de la revisión y, a continuación, enviar la revisión al bucket.

Note

Aunque no se trata en este tutorial, puede utilizarla CodeDeploy para implementar aplicaciones desde GitHub los repositorios hasta las instancias. Para obtener más información, consulte [Integración CodeDeploy con GitHub](#).

Temas

- [Aprovisionar un bucket de Amazon S3](#)
- [Preparación de los archivos de la aplicación para el bucket](#)
- [Agrupación de los archivos de la aplicación en un único archivo de almacenamiento y envío del archivo de almacenamiento](#)

Aprovisionar un bucket de Amazon S3

Cree un contenedor de almacenamiento o bucket en Amazon S3 o utilice un bucket existente. Asegúrese de que puede cargar la revisión en el bucket y que las instancias Amazon EC2 utilizadas en las implementaciones pueden descargar la revisión del bucket.

Puede utilizar la AWS CLI consola de Amazon S3 o las API de Amazon S3 para crear un bucket de Amazon S3. Después de crear el bucket, asegúrese de conceder permisos de acceso al bucket y a su CodeDeploy usuario.

Note

Los nombres de los buckets deben ser únicos en Amazon S3 para todas AWS las cuentas. Si no puede utilizar **codedeploydemobucket**, pruebe otro nombre de bucket, por ejemplo, **codedeploydemobucket** seguido de un guion y sus iniciales o algún otro identificador único. A continuación, cuando aparezca **codedeploydemobucket** en este tutorial, asegúrese de sustituirlo por el nombre de su bucket.

El bucket de Amazon S3 debe crearse en la misma AWS región en la que se lanzaron las instancias de Amazon EC2 de destino. Por ejemplo, si crea el bucket en la región Este de EE. UU. (Norte de Virginia), sus instancias de Amazon EC2 de destino deben lanzarse en la región Este de EE. UU. (Norte de Virginia).

Temas

- [Para crear un bucket de Amazon S3 \(CLI\)](#)
- [Para crear un bucket de Amazon S3 \(consola\)](#)
- [Conceda permisos al bucket de Amazon S3 y a su AWS cuenta](#)

Para crear un bucket de Amazon S3 (CLI)

Llame al comando mb para crear un bucket de Amazon S3 denominado **codedeploydemobucket**:

```
aws s3 mb s3://codedeploydemobucket --region region
```

Para crear un bucket de Amazon S3 (consola)

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
2. En la consola de Amazon S3, elija Crear bucket.
3. En el cuadro Bucket name (Nombre del bucket), escriba un nombre para el bucket.
4. En la lista Region (Región), elija la región de destino y, a continuación, seleccione Create (Crear).

Conceda permisos al bucket de Amazon S3 y a su AWS cuenta

Debe tener permisos para cargar en el bucket de Amazon S3. Puede especificar estos permisos mediante una política de bucket de Amazon S3. Por ejemplo, en la siguiente política de bucket de Amazon S3, el uso del carácter comodín (*) permite 111122223333 a la AWS cuenta cargar archivos en cualquier directorio del bucket de Amazon S3 denominado `codedeploydemobucket`:

```
{
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::codedeploydemobucket/*",
      "Principal": {
        "AWS": [
          "111122223333"
        ]
      }
    }
  ]
}
```

Para ver el ID de su AWS cuenta, consulte [Cómo encontrar el ID de su AWS cuenta](#).

Ahora es un buen momento para comprobar que el bucket de Amazon S3 permite las solicitudes de descarga de cada instancia de Amazon EC2 participante. Puede especificarlos mediante una política de bucket de Amazon S3. Por ejemplo, en la siguiente política de bucket de Amazon S3, el carácter comodín (*) permite a cualquier instancia de Amazon EC2 asociada a un perfil de instancia de IAM que contenga el ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` descargar desde cualquier directorio al bucket de Amazon S3 llamado `codedeploydemobucket`:

```
{
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",

```

```

    "Resource": "arn:aws:s3:::codedeploydemobucket/*",
    "Principal": {
      "AWS": [
        "arn:aws:iam::444455556666:role/CodeDeployDemo"
      ]
    }
  ]
}

```

Para obtener información acerca de cómo generar y adjuntar una política de bucket de Amazon S3, consulte [Ejemplos de política de bucket](#).

El usuario CodeDeploy administrativo en el que creó también [Paso 1: Configurar](#) debe tener permiso para cargar la revisión en el bucket de Amazon S3. Una forma de especificarlo es mediante una política de IAM, que se añade al conjunto de permisos del usuario o a un rol de IAM (que se deja que el usuario asuma). La siguiente política de IAM permite que el usuario cargue revisiones en cualquier lugar del bucket de Amazon S3 denominado codedeploydemobucket:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
      "Resource": "arn:aws:s3:::codedeploydemobucket/*"
    }
  ]
}

```

Para obtener información sobre cómo crear una política de IAM, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM. Para obtener información sobre cómo añadir una política a un conjunto de permisos, consulte [Crear un conjunto de permisos](#) en la Guía del usuario de AWS IAM Identity Center .

Preparación de los archivos de la aplicación para el bucket

Asegúrese de que la página web, el AppSpec archivo y el script estén organizados en su máquina de desarrollo de la siguiente manera:

```
c:\
```

```
|-- temp\  
    |--HelloWorldApp\  
        |-- appspec.yml  
        |-- before-install.bat  
        |-- index.html
```

Agrupación de los archivos de la aplicación en un único archivo de almacenamiento y envío del archivo de almacenamiento

Agrupe los archivos en un archivo de almacenamiento (se denomina una revisión de la aplicación).

Note

Es posible que se cobre por almacenar objetos en un bucket y por transferir revisiones de la aplicación dentro y fuera de un bucket. Para obtener más información, consulte [Precios de Amazon S3](#).

1. En el equipo de desarrollo, cambie a la carpeta en la que se almacenan los archivos:

```
cd c:\temp\HelloWorldApp
```

Note

Si no cambia a esta carpeta, la agrupación del archivo comenzará en su carpeta actual. Por ejemplo, si la carpeta actual es `c:\temp` en lugar de `c:\temp\HelloWorldApp`, la agrupación comenzará con los archivos y subcarpetas de la carpeta `c:\temp`, lo que puede incluir algo más que la subcarpeta `HelloWorldApp`.

2. Ejecute el `create-application` comando para registrar una nueva aplicación cuyo nombre **HelloWorld_App** sea CodeDeploy:

```
aws deploy create-application --application-name HelloWorld_App
```

3. Ejecute el comando CodeDeploy [push](#) para agrupar los archivos, cargar las revisiones en Amazon S3 y registrar la información CodeDeploy sobre la revisión cargada, todo en una sola acción.

```
aws deploy push --application-name HelloWorld_App --s3-location s3://  
codedeploydemobucket/HelloWorld_App.zip --ignore-hidden-files
```

Este comando agrupa los archivos del directorio actual (excluyendo los archivos ocultos) en un único archivo denominado `HelloWorld_App.zip`, carga la revisión en el **codedeploydemobucket** depósito y registra la información CodeDeploy sobre la revisión cargada.

Paso 4: Implementar la aplicación Hello World.

Ahora implementa la revisión de la aplicación Hello, World! de ejemplo que ha cargado en Amazon S3. Utilice la consola AWS CLI o la CodeDeploy consola para implementar la revisión y supervisar el progreso de la implementación. Una vez implementada correctamente la revisión de la aplicación, comprobará los resultados.

Temas

- [Implemente la revisión de su aplicación con CodeDeploy](#)
- [Monitorización y solución de problemas de la implementación](#)
- [Comprobación de la implementación](#)

Implemente la revisión de su aplicación con CodeDeploy

Puede implementar su aplicación con la CLI o la consola.

Temas

- [Para implementar la revisión de la aplicación \(CLI\)](#)
- [Para implementar la revisión de la aplicación \(consola\)](#)

Para implementar la revisión de la aplicación (CLI)

1. En primer lugar, la implementación necesita un grupo de implementaciones. Sin embargo, para crear el grupo de implementaciones, primero necesita un ARN de rol de servicio. Un rol de servicio es un rol de IAM que concede a un servicio permiso para actuar en su nombre. En este caso, el rol de servicio otorga CodeDeploy permiso para acceder a sus instancias de Amazon EC2 para expandir (leer) sus etiquetas de instancia de Amazon EC2.

Ya debe haber seguido las instrucciones de [Creación de un rol de servicio \(CLI\)](#) para crear un rol de servicio. Para obtener el ARN del rol de servicio, consulte [Obtención del ARN del rol de servicio \(CLI\)](#).

- Ahora que tiene el ARN, llame al comando `create-deployment-group` para crear un grupo de implementación denominado **HelloWorld_DepGroup** asociado a la aplicación **HelloWorld_App**, utilizando la etiqueta de instancia de Amazon EC2 denominada **CodeDeployDemo** y la configuración de implementación denominada **CodeDeployDefault.OneAtATime** con el ARN del rol de servicio:

```
aws deploy create-deployment-group --application-name HelloWorld_App
--deployment-group-name HelloWorld_DepGroup --deployment-
config-name CodeDeployDefault.OneAtATime --ec2-tag-filters
Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE --service-role-arn serviceRoleARN
```

Note

El `create-deployment-group` comando permite crear activadores que provoquen el envío de notificaciones de Amazon SNS a los suscriptores del tema sobre eventos específicos en implementaciones e instancias. El comando también admite opciones para revertir automáticamente las implementaciones y configurar alarmas para detener las implementaciones cuando se cumplen los umbrales de monitoreo de CloudWatch las alarmas de Amazon. Los comandos para estas acciones no se incluyen en este tutorial.

- Antes de crear una implementación, las instancias de su grupo de implementación deben tener el CodeDeploy agente instalado. Puede instalar el agente desde la línea de comandos con AWS Systems Manager mediante el comando siguiente:

```
aws ssm create-association --name AWS-ConfigureAWSPackage
--targets Key=tag:Name,Values=CodeDeployDemo --parameters
action=Install,name=AWSCodeDeployAgent --schedule-expression "cron(0 2 ? * SUN
*)"
```

Este comando crea una asociación en Systems Manager State Manager que instalará el CodeDeploy agente e intentará actualizarlo todos los domingos a las 2:00 de la mañana. Para obtener más información sobre el CodeDeploy agente, consulte [Trabajar con el CodeDeploy agente](#). Para obtener más información acerca de Systems Manager, consulte [¿Qué es AWS Systems Manager?](#).

4. Ejecute ahora el comando `create-deployment` para crear una implementación asociada a la aplicación **HelloWorld_App**, la configuración de implementación **CodeDeployDefault.OneAtATime** y el grupo de implementaciones **HelloWorld_DepGroup** utilizando la revisión de la aplicación **HelloWorld_App.zip** del bucket **codedeploydemobucket**:

```
aws deploy create-deployment --application-name HelloWorld_App --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-location bucket=codedeploydemobucket,bundleType=zip,key=HelloWorld_App.zip
```


Para implementar la revisión de la aplicación (consola)

1. Antes de usar la CodeDeploy consola para implementar la revisión de la aplicación, necesita un ARN de rol de servicio. Un rol de servicio es un rol de IAM que concede a un servicio permiso para actuar en su nombre. En este caso, el rol de servicio otorga CodeDeploy permiso para acceder a sus instancias de Amazon EC2 para expandir (leer) sus etiquetas de instancia de Amazon EC2.

Ya debe haber seguido las instrucciones de [Creación de un rol de servicio \(consola\)](#) para crear un rol de servicio. Para obtener el ARN del rol de servicio, consulte [Obtención del ARN del rol de servicio \(consola\)](#).

2. Ahora que tiene el ARN, puede usar la CodeDeploy consola para implementar la revisión de la aplicación.

Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

3. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
4. Selecciona HelloWorld_App.
5. En la pestaña Deployment groups (Grupos de implementaciones), elija Create deployment group (Crear grupo de implementaciones).
6. En Nombre de grupo de implementación, escriba **HelloWorld_DepGroup**.
7. En Service Role (Rol de servicio), elija el nombre del rol de servicio.

8. En Tipo de implementación, elija Local.
9. En Configuración de entorno, elija la pestaña Instancias de Amazon EC2.
10. En la configuración del agente con AWS Systems Manager, mantenga los valores predeterminados.
11. En Clave, introduzca **Name**.
12. En Valor, escriba **CodeDeployDemo**.
13. En Configuración de despliegue, elija CodeDeployDefault.OneAtATime.
14. En Equilibrador de carga, anule la selección de Habilitar equilibración de carga.
15. Elija Crear grupo de implementación.
16. Elija Crear implementación.
17. En el grupo de implementación, elija _ HelloWorld DepGroup
18. En Tipo de revisión, elija Mi aplicación está almacenada en Amazon S3 y, a continuación, en Ubicación de la revisión, escriba la ubicación de la revisión de la aplicación de ejemplo Hello, World! que cargó previamente en Amazon S3. Para obtener la ubicación:
 - a. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
 - b. En la lista de buckets, elija codedeploydemobucket (o el nombre del bucket en el que cargó la revisión de la aplicación).
 - c. En la lista de objetos, elija HelloWorld_App.zip.
 - d. En la pestaña Overview (Información general), elija Copy path (Ruta de copia).
 - e. Vuelva a la CodeDeploy consola y, en Ubicación de revisión, pegue el valor del campo Enlace.
19. En Revision file type (Tipo de archivo de revisión), elija .zip.
20. (Opcional) Escriba un comentario en Deployment description (Descripción de implementación).
21. Elija Crear implementación. En la página Deployments se mostrará información sobre la implementación que acaba de crear.

Monitorización y solución de problemas de la implementación

Utilice la consola AWS CLI o la consola para supervisar y solucionar los problemas de la implementación.

Temas

- [Para monitorizar y solucionar problemas en la implementación \(CLI\)](#)
- [Para monitorizar y resolver problemas de la implementación \(consola\)](#)

Para monitorizar y solucionar problemas en la implementación (CLI)

1. Obtenga el ID de la implementación; para ello, ejecute el comando `list-deployments` para la aplicación **HelloWorld_App** y el grupo de implementaciones **HelloWorld_DepGroup**:

```
aws deploy list-deployments --application-name HelloWorld_App --deployment-group-name HelloWorld_DepGroup --query "deployments" --output text
```

2. Llame al comando `get-deployment` con el ID de la implementación:

```
aws deploy get-deployment --deployment-id deploymentID --query "deploymentInfo.status" --output text
```

3. El comando devuelve el estado general de la implementación. Si todo sale bien, el valor es `Succeeded`.

Si el estado general es `Failed`, puede ejecutar comandos como [list-deployment-instances](#) y [get-deployment-instance](#) para solucionar problemas. Si desea conocer más opciones de solución de problemas, consulte [Análisis de archivos de registro para investigar los errores de implementación en instancias](#).

Para monitorizar y resolver problemas de la implementación (consola)

En la página de implementaciones de la CodeDeploy consola, puede supervisar el estado de la implementación en la columna Estado.

Para obtener más información acerca de la implementación, sobre todo si el valor de la columna Status tiene un valor distinto de `Succeeded`:

1. En la tabla Deployments (Implementaciones), elija el ID de implementación. Cuando se produce un error en una implementación, aparece un mensaje que describe el motivo del error en la página de detalles de la implementación.
2. Aparece más información sobre las instancias de la implementación. Cuando se una implementación falla, es posible que pueda determinar las instancias de Amazon EC2 afectadas y el paso del proceso en el que se ha producido el error.

3. Si desea realizar más resolución de problemas, puede utilizar una técnica como [View Instance Details](#). También puede analizar los archivos de registro de la implementación en una instancia de Amazon EC2. Para obtener más información, consulte [Análisis de archivos de registro para investigar los errores de implementación en instancias](#).

Comprobación de la implementación

Después que la implementación se realiza correctamente, verifique que funciona la instalación. Utilice la dirección de DNS público de la instancia de Amazon EC2 para ver la página web en un navegador web. (Para obtener el valor de DNS público, en la consola de Amazon EC2, seleccione la instancia de Amazon EC2 y, en la pestaña Descripción, busque el valor Public DNS).

Por ejemplo, si la dirección DNS pública de la instancia de Amazon EC2 es **ec2-01-234-567-890.compute-1.amazonaws.com**, se usará la siguiente dirección URL:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si la operación se realiza correctamente, debería ver una página web Hello, World!.

Paso 5: Actualizar y volver a implementar su aplicación "Hello, World!"

Una vez que hayas implementado correctamente la revisión de la aplicación, actualiza el código de la página web en la máquina de desarrollo y úsala CodeDeploy para volver a implementar el sitio. Una vez implementada nuevamente la revisión, debe poder ver los cambios en el código en la instancia de Amazon EC2.

Temas

- [Modificar la página web](#)
- [Nueva implementación del sitio](#)

Modificar la página web

1. Vaya a la subcarpeta `c:\temp\HelloWorldApp` y use un editor de texto para modificar el archivo `index.html`:

```
cd c:\temp\HelloWorldApp
notepad index.html
```

2. Revise el contenido del archivo `index.html` para cambiar el color de fondo y parte del texto de la página web y, a continuación, guarde el archivo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Hello Again, World!</title>
  <style>
    body {
      color: #ffffff;
      background-color: #66cc00;
      font-family: Arial, sans-serif;
      font-size:14px;
    }
  </style>
</head>
<body>
  <div align="center"><h1>Hello Again, World!</h1></div>
  <div align="center"><h2>You have successfully deployed a revision of an
application using CodeDeploy</h2></div>
  <div align="center">
    <p>What to do next? Take a look through the <a href="https://aws.amazon.com/codedeploy">CodeDeploy Documentation</a>.</p>
  </div>
</body>
</html>
```

Nueva implementación del sitio

Ahora que ha modificado el código, utilice Amazon S3 CodeDeploy para volver a implementar la página web.

Agrupe y cargue los cambios en Amazon S3, como se describe en [Agrupación de los archivos de la aplicación en un único archivo de almacenamiento y envío del archivo de almacenamiento](#).

(Cuando siga esas instrucciones, recuerde que no tiene que crear una nueva aplicación). Asigne a la nueva revisión la misma clave que antes (**HelloWorld_App.zip**). Carguela en el mismo bucket de Amazon S3 que creó anteriormente (por ejemplo, **codedeploydemobucket**).

Utilice la consola AWS CLI o la CodeDeploy consola para volver a implementar el sitio.

Temas

- [Para volver implementar el sitio \(CLI\)](#)
- [Para volver a implementar el sitio \(consola\)](#)

Para volver implementar el sitio (CLI)

Ejecute el comando `create-deployment` para crear una implementación basada en la revisión que ha cargado, utilizando también la aplicación **HelloWorld_App**, la configuración de implementación **CodeDeployDefault.OneAtATime**, el grupo de implementaciones **HelloWorld_DepGroup** y la revisión de la aplicación **HelloWorld_App.zip** del bucket **codedeploydemobucket**:

```
aws deploy create-deployment --application-name HelloWorld_App --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name HelloWorld_DepGroup --s3-location bucket=codedeploydemobucket,bundleType=zip,key=HelloWorld_App.zip
```

Puede comprobar el estado de la nueva implementación como se describe en [Monitorización y solución de problemas de la implementación](#).

Cuando CodeDeploy haya redistribuido el sitio, vuelva a visitarlo en su navegador web para comprobar que se han cambiado el color de fondo y el texto de la página web. (Puede que tenga que actualizar el navegador). Si el color de fondo y el texto se han modificado, ¡enhorabuena! Ha modificado y ha vuelto a implementar su sitio.

Para volver a implementar el sitio (consola)

1. [Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en https://console.aws.amazon.com/codedeploy](https://console.aws.amazon.com/codedeploy).

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, elija Applications (Aplicaciones).
3. En la lista de aplicaciones, elija HelloWorld_App.
4. En la pestaña Deployments (Implementaciones), elija Create deployment (Crear implementación).
 - a. En la lista de grupos de despliegues, elija HelloWorld_DepGroup.

- b. En Ubicación de la revisión, introduzca el enlace de Amazon S3 de la revisión.

Para encontrar el valor del enlace:

- i. Inicie sesión en la consola de Amazon S3 AWS Management Console y ábrala en <https://console.aws.amazon.com/s3/>.

Vaya al archivo `codedeploydemobucket`, ábralo y elija la revisión, **HelloWorld_App.zip**, en la consola de Amazon S3.

- ii. Si el panel Propiedades no se muestra en la consola de Amazon S3, elija el botón Propiedades.
 - iii. En el panel Properties (Propiedades), copie el valor del campo Link (Enlace).
 - iv. Regrese a la CodeDeploy consola y, a continuación, pegue el enlace en la ubicación de revisión.
 - v.
- c. En Revision file type (Tipo de archivo de revisión), si aparece un mensaje en el que se indica que no se ha podido detectar el tipo de archivo, elija `.zip`.
 - d. Deje Deployment description (Descripción de implementación) en blanco.
 - e. Amplíe Anulaciones de grupos de despliegues En la lista de configuraciones de despliegue, seleccione `CodeDeployDefault.OneAtATime` y, a continuación, seleccione Crear despliegue.

Puede comprobar el estado de la implementación como se describe en [Monitorización y solución de problemas de la implementación](#).

Cuando CodeDeploy haya vuelto a implementar el sitio, vuelva a visitarlo en su navegador web para comprobar que el color de fondo y el texto de la página web han cambiado. (Puede que tenga que actualizar el navegador). Si el color de fondo y el texto se han modificado, ¡enhorabuena! Ha modificado y ha vuelto a implementar su sitio.

Paso 6: Limpiar la aplicación de "Hello, World!" y sus recursos relacionados

Ha actualizado el código de "Hello, World!" correctamente e implementado de nuevo el sitio. Para evitar cargos recurrentes por los recursos que ha creado para completar este tutorial, debe eliminar:

- Cualquier AWS CloudFormation pila (o cierre cualquier instancia de Amazon EC2, si la creó fuera AWS CloudFormation de ella).

- Todos los buckets de Amazon S3.
- La aplicación HelloWorld_App en CodeDeploy.
- La asociación de administradores AWS Systems Manager estatales del CodeDeploy agente.

Puede utilizar Amazon S3 AWS CloudFormation, Amazon EC2 y CodeDeploy las consolas o las AWS API para realizar la limpieza. AWS CLI

Temas

- [Para limpiar los recursos \(CLI\)](#)
- [Para limpiar los recursos \(consola\)](#)
- [Siguiendo pasos](#)

Para limpiar los recursos (CLI)

1. Si usó la AWS CloudFormation pila para este tutorial, elimine la pila ejecutando el delete-stack comando correspondiente a la pila nombrada. **CodeDeployDemoStack** Esto termina todas las instancias de Amazon EC2 vinculadas y elimina todos los roles de IAM asociados creados originalmente por la pila.

```
aws cloudformation delete-stack --stack-name CodeDeployDemoStack
```

2. Para eliminar el bucket de Amazon S3, llame al comando rm con el modificador --recursive para el bucket denominado **codedeploydemobucket**. Así elimina el bucket y todos los objetos que contiene.

```
aws s3 rm s3://codedeploydemobucket --recursive --region region
```

3. Para eliminar la HelloWorld_App aplicación CodeDeploy, ejecuta el delete-application comando. Esto elimina todos los registros de grupo de implementaciones asociados y los registros de implementación de la aplicación.

```
aws deploy delete-application --application-name HelloWorld_App
```

4. Para eliminar la asociación de Systems Manager State Manager, llame al comando delete-association.

```
aws ssm delete-association --association-id association-id
```

Para obtener el valor de *id-asociación*, llame al comando describe-association.

```
aws ssm describe-association --name AWS-ConfigureAWSPackage --targets  
Key=tag:Name,Values=CodeDeployDemo
```

5. Si no usó la AWS CloudFormation pila para este tutorial, terminate-instances ejecute el comando para terminar las instancias de Amazon EC2 que creó manualmente. Indique el ID de la instancia de Amazon EC2 que desea terminar.

```
aws ec2 terminate-instances --instance-ids instanceId
```

Para limpiar los recursos (consola)

Si utilizó nuestra AWS CloudFormation plantilla para este tutorial, elimine la AWS CloudFormation pila asociada.

1. Inicie sesión AWS Management Console y abra la AWS CloudFormation consola en <https://console.aws.amazon.com/cloudformation>.
2. En el cuadro de búsqueda, escribe el nombre de la AWS CloudFormation pila (por ejemplo, **CodeDeployDemoStack**).
3. Seleccione la casilla junto al nombre de la pila.
4. En el menú Actions (Acciones), elija Delete Stack (Eliminar pila). Esto elimina la pila, termina todas las instancias de Amazon EC2 asociadas y elimina todos los roles de IAM asociados.

Para finalizar las instancias de Amazon EC2 que creó fuera de una AWS CloudFormation pila:

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. En el área Instances, elija Instances.
3. En el campo de búsqueda, escriba el nombre de la instancia de Amazon EC2 que desee eliminar y, a continuación, pulse Intro.
4. Seleccione la instancia de Amazon EC2.


5. Elija Acciones, seleccione Estado de la instancia y elija Terminar. Cuando se le pregunte, elija Yes, Terminate. Repita estos pasos para todas las instancias de Amazon EC2.

Para eliminar el bucket de Amazon S3:

1. Inicie sesión en la consola de Amazon S3 AWS Management Console y ábrala en <https://console.aws.amazon.com/s3/>.
2. En la lista de buckets, busque y seleccione el nombre del bucket de Amazon S3 (por ejemplo, **codedeploydemobucket**).
3. Para eliminar un bucket, primero debe eliminar su contenido. Seleccione todos los archivos del bucket, como **HelloWorld_App.zip**. En el menú Actions, elija Delete. Cuando se le pida confirmar la eliminación, elija OK.
4. Una vez vacío, puede eliminar el bucket. En la lista de buckets, elija la línea del bucket (pero no su nombre). Elija Delete bucket y cuando se le pida que confirme, seleccione OK.

Para eliminar la HelloWorld_App aplicación de CodeDeploy:

1. Inicie sesión AWS Management Console y abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. Elija **HelloWorld_App**.
4. Seleccione Eliminar aplicación.
5. Cuando se le pregunte, escriba **Delete** y, a continuación, elija Delete (Eliminar).

Para eliminar la asociación de Systems Manager State Manager:

1. Abra la AWS Systems Manager consola en <https://console.aws.amazon.com/systems-manager>.
2. En el panel de navegación, elija Administrador de estados.
3. Elija la asociación que ha creado y, después, elija Eliminar.

Siguientes pasos

Si ha llegado aquí, significa que ha completado correctamente una implementación con CodeDeploy. ¡Enhorabuena!

Tutorial: Implemente una aplicación en una instancia local con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux)

Este tutorial lo ayuda a adquirir experiencia CodeDeploy al guiarlo a través de la implementación de una revisión de aplicación de muestra en una única instancia local (es decir, un dispositivo físico que no es una instancia de Amazon EC2) que ejecute Windows Server, Ubuntu Server o Red Hat Enterprise Linux (RHEL). Para obtener información sobre las instancias locales y cómo funcionan con ellas, consulte. CodeDeploy [Working with On-Premises Instances](#)

¿No es lo que está buscando?

- Para practicar la implementación en una instancia de Amazon EC2 con Amazon Linux o RHEL, consulte [Tutorial: Implementación WordPress en una instancia de Amazon EC2 \(Amazon Linux o Red Hat Enterprise Linux y Linux, macOS o Unix\)](#).
- Para practicar la implementación en una instancia de Amazon EC2 con Windows Server, consulte [Tutorial: Implementación de una aplicación "Hello, World!" aplicación con CodeDeploy \(Windows Server\)](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Configurar la instancia en las instalaciones](#)
- [Paso 2: Crear una revisión de aplicación de ejemplo](#)
- [Paso 3: Empaquetado y carga de la revisión de aplicación en Amazon S3](#)
- [Paso 4: Implementar la revisión de la aplicación](#)
- [Paso 5: Comprobar la implementación](#)
- [Paso 6: Limpiar los recursos](#)

Requisitos previos

Antes de empezar este tutorial, debe cumplir los requisitos previos, entre los que se incluyen la configuración de un usuario [Empezar con CodeDeploy](#), la instalación o actualización del rol de servicio y la AWS CLI creación de un rol de servicio. No es necesario crear un perfil de instancia de IAM como se describe en los requisitos previos. Las instancias en las instalaciones no utilizan perfiles de instancia de IAM.

El dispositivo físico que configurará como instancia on-premises debe emplear uno de los sistemas operativos indicados en [Sistemas operativos compatibles con el agente CodeDeploy](#).

Paso 1: Configurar la instancia en las instalaciones

Antes de implementar en la instancia en las instalaciones, debe configurarla. Siga las instrucciones de [Working with On-Premises Instances](#) y, a continuación, vuelva a esta página.

Instale el agente CodeDeploy

Tras configurar la instancia local, siga los pasos para las instancias locales que se indican en [Instalar el CodeDeploy agente](#) y vuelva a esta página.

Paso 2: Crear una revisión de aplicación de ejemplo

En este paso, creará una revisión de aplicación de ejemplo para implementarla en la instancia en las instalaciones.

Dado que es difícil conocer el software y las características que ya están instalados -o que las políticas de su organización permiten instalar- en la instancia en las instalaciones, la revisión de aplicación de ejemplo que incluimos aquí usa simplemente scripts por lotes (en Windows Server) o scripts de intérpretes de comandos (en Ubuntu Server y RHEL) para escribir archivos de texto en una ubicación de la instancia en las instalaciones. Se escribe un archivo para cada uno de los diversos eventos del ciclo de vida de la CodeDeploy implementación, incluidos los eventos de instalación, `AfterInstallApplicationStart`, y `ValidateService`. Durante el evento del ciclo de vida de la `BeforeInstall` implementación, se ejecutará un script para eliminar los archivos antiguos escritos durante las implementaciones anteriores de este ejemplo y crear una ubicación en la instancia local en la que escribir los nuevos archivos.

Note

La implementación de la revisión de aplicación de ejemplo puede fallar si se da cualquiera de las siguientes circunstancias:

- El usuario que inicia el CodeDeploy agente en la instancia local no tiene permiso para ejecutar scripts.
- El usuario no tiene permiso para crear o eliminar carpetas en las ubicaciones indicadas en los scripts.
- El usuario no tiene permiso para crear archivos de texto en las ubicaciones indicadas en los scripts.

Note

Si ha configurado una instancia de Windows Server y desea implementar un ejemplo diferente, puede ser conveniente usar el indicado en [Paso 2: Configurar el contenido de origen para implementarlo en la instancia de Amazon EC2 de Windows Server](#) en el tutorial [Tutorial: Implementación de una aplicación "Hello, World!" aplicación con CodeDeploy \(Windows Server\)](#).

Si ha configurado una instancia de RHEL y desea implementar un ejemplo diferente, puede ser conveniente usar el indicado en [Paso 2: Configurar el contenido fuente para implementarlo en la instancia de Amazon EC2 de Amazon Linux o Red Hat Enterprise Linux](#) en el tutorial [Tutorial: Implementación WordPress en una instancia de Amazon EC2 \(Amazon Linux o Red Hat Enterprise Linux y Linux, macOS o Unix\)](#).

En la actualidad, no hay ningún ejemplo alternativo para Ubuntu Server.

1. En el equipo de desarrollo, cree un subdirectorio (subcarpeta) con el nombre `CodeDeployDemo-OnPrem` para almacenar los archivos de la revisión de aplicación de ejemplo y, a continuación, sitúese en esa subcarpeta. En este ejemplo, supondremos que va a utilizar la carpeta `c:\temp` como la carpeta raíz para Windows Server o la carpeta `/tmp` como la carpeta raíz para Ubuntu Server y RHEL. Si utiliza una carpeta diferente, asegúrese de indicarla en lugar de la anterior en todo el tutorial:

Para Windows:

```
mkdir c:\temp\CodeDeployDemo-OnPrem
cd c:\temp\CodeDeployDemo-OnPrem
```

Para Linux, macOS o Unix:

```
mkdir /tmp/CodeDeployDemo-OnPrem
cd /tmp/CodeDeployDemo-OnPrem
```

2. En la raíz de la subcarpeta `CodeDeployDemo-OnPrem`, utilice un editor de texto para crear dos archivos con los nombres `appspec.yml` e `install.txt`:

`appspec.yml` para Windows Server:

```
version: 0.0
os: windows
files:
  - source: .\install.txt
    destination: c:\temp\CodeDeployExample
hooks:
  BeforeInstall:
    - location: .\scripts\before-install.bat
      timeout: 900
  AfterInstall:
    - location: .\scripts\after-install.bat
      timeout: 900
  ApplicationStart:
    - location: .\scripts\application-start.bat
      timeout: 900
  ValidateService:
    - location: .\scripts\validate-service.bat
      timeout: 900
```

`appspec.yml` para Ubuntu Server y RHEL:

```
version: 0.0
os: linux
files:
  - source: ./install.txt
    destination: /tmp/CodeDeployExample
hooks:
  BeforeInstall:
```

```
- location: ./scripts/before-install.sh
  timeout: 900
AfterInstall:
- location: ./scripts/after-install.sh
  timeout: 900
ApplicationStart:
- location: ./scripts/application-start.sh
  timeout: 900
ValidateService:
- location: ./scripts/validate-service.sh
  timeout: 900
```

Para obtener más información sobre AppSpec los archivos, consulte [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#) y [CodeDeploy AppSpec referencia de archivo](#).

install.txt:

```
The Install deployment lifecycle event successfully completed.
```

3. En la raíz de la subcarpeta CodeDeployDemo-OnPrem, cree una subcarpeta scripts y sitúese en ella:

Para Windows:

```
mkdir c:\temp\CodeDeployDemo-OnPrem\scripts
cd c:\temp\CodeDeployDemo-OnPrem\scripts
```

Para Linux, macOS o Unix:

```
mkdir -p /tmp/CodeDeployDemo-OnPrem/scripts
cd /tmp/CodeDeployDemo-OnPrem/scripts
```

4. En la raíz de la subcarpeta scripts, utilice un editor de texto para crear cuatro archivos con los nombres before-install.bat, after-install.bat, application-start.bat y validate-service.bat para Windows Server, o before-install.sh, after-install.sh, application-start.sh y validate-service.sh para Ubuntu Server y RHEL:

Para Windows Server:

before-install.bat:

```
set FOLDER=%HOMEDRIVE%\temp\CodeDeployExample

if exist %FOLDER% (
  rd /s /q "%FOLDER%"
)

mkdir %FOLDER%
```

after-install.bat:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The AfterInstall deployment lifecycle event successfully completed. > after-
install.txt
```

application-start.bat:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The ApplicationStart deployment lifecycle event successfully completed. >
application-start.txt
```

validate-service.bat:

```
cd %HOMEDRIVE%\temp\CodeDeployExample

echo The ValidateService deployment lifecycle event successfully completed. >
validate-service.txt
```

Para Ubuntu Server y RHEL:**before-install.sh:**

```
#!/bin/bash
export FOLDER=/tmp/CodeDeployExample

if [ -d $FOLDER ]
then
```

```
rm -rf $FOLDER
fi

mkdir -p $FOLDER
```

after-install.sh:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The AfterInstall deployment lifecycle event successfully completed." > after-
install.txt
```

application-start.sh:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The ApplicationStart deployment lifecycle event successfully completed." >
application-start.txt
```

validate-service.sh:

```
#!/bin/bash
cd /tmp/CodeDeployExample

echo "The ValidateService deployment lifecycle event successfully completed." >
validate-service.txt

unset FOLDER
```

5. Solo para Ubuntu Server y RHEL, asegúrese de que los cuatro scripts de intérpretes de comandos tengan permisos de ejecución:

```
chmod +x ./scripts/*
```

Paso 3: Empaquetado y carga de la revisión de aplicación en Amazon S3

Para poder implementar la revisión de la aplicación, antes tiene que agrupar sus archivos y cargar el paquete en un bucket de Amazon S3. Siga las instrucciones en [Cree una aplicación con CodeDeploy](#) y [Enviar una revisión CodeDeploy a Amazon S3 \(solo para implementaciones de EC2/on-premise\)](#). (Aunque puede dar cualquier nombre a la aplicación y al grupo de implementaciones, le recomendamos especificar `CodeDeploy-OnPrem-App` como nombre para la aplicación y `CodeDeploy-OnPrem-DG` como nombre del grupo de implementaciones). Una vez que haya completado las instrucciones, vuelve a esta página.

Note

Como alternativa, puede cargar el paquete de archivos en un GitHub repositorio e implementarlo desde allí. Para obtener más información, consulte [Integración CodeDeploy con GitHub](#).

Paso 4: Implementar la revisión de la aplicación

Una vez cargada la revisión de la aplicación en un bucket de Amazon S3, intente implementarla en la instancia en las instalaciones. Siga las instrucciones de [Cree una implementación con CodeDeploy](#) y, a continuación, vuelva a esta página.

Paso 5: Comprobar la implementación

Para comprobar si la implementación se ha realizado correctamente, siga las instrucciones de [Ver detalles CodeDeploy de la implementación](#) y, a continuación, vuelva a esta página.

Si la implementación se ha realizado correctamente, verá cuatro archivos de texto en la carpeta `c:\temp\CodeDeployExample` (para Windows Server) o `/tmp/CodeDeployExample` (para Ubuntu Server y RHEL).

Si la implementación ha fallado, siga los pasos de resolución de problemas de [View Instance Details](#) y [Solución de problemas de instancias](#). Haga las correcciones necesarias, vuelva a empaquetar y cargar la revisión de la aplicación e intente implementarla de nuevo.

Paso 6: Limpiar los recursos

Para evitar cargos recurrentes por los recursos que ha creado para este tutorial, elimine el bucket de Amazon S3 si no va a utilizarlo. También puede limpiar los recursos asociados, como los registros del grupo de aplicaciones y despliegues CodeDeploy y la instancia local.

Puede utilizar las consolas Amazon S3 AWS CLI o una combinación de ellas AWS CLI para limpiar los recursos. CodeDeploy

Limpieza de los recursos (CLI)

Para eliminar el bucket de Amazon S3

- Ejecute el comando [rm](#) con el modificador `--recursive` para el bucket (por ejemplo, `codedeploydemobucket`). El bucket y todos los objetos que contiene se eliminarán.

```
aws s3 rm s3://your-bucket-name --recursive --region region
```

Para eliminar los registros del grupo de aplicaciones y despliegues en CodeDeploy

- Llame al comando [delete-application](#) para la aplicación (por ejemplo, `CodeDeploy-OnPrem-App`). Los registros de la implementación y del grupo de implementaciones se eliminarán.

```
aws deploy delete-application --application-name your-application-name
```

Para cancelar el registro de la instancia on-premises y eliminar el usuario de IAM

- Ejecute el comando [deregister](#) para la instancia en las instalaciones y la región:

```
aws deploy deregister --instance-name your-instance-name --delete-iam-user --  
region your-region
```

Note

Si no desea eliminar el usuario de IAM asociado a esta instancia en las instalaciones, especifique la opción `--no-delete-iam-user`.

Para desinstalar el CodeDeploy agente y eliminar el archivo de configuración de la instancia local

- Desde la instancia en las instalaciones, llame al comando [uninstall](#):

```
aws deploy uninstall
```

Ha completado todos los pasos para limpiar los recursos utilizados en este tutorial.

Limpieza de los recursos (consola)

Para eliminar el bucket de Amazon S3

1. Inicie sesión en la consola de Amazon S3 AWS Management Console y ábrala en <https://console.aws.amazon.com/s3/>.
2. Elija el icono junto al bucket que desea eliminar (por ejemplo, codedeploydemobucket), pero no elija el bucket en sí.
3. Elija Acciones y, a continuación, elija Eliminar.
4. Cuando se le pida confirmación de que desea eliminar el bucket, elija OK.

Para eliminar los registros del grupo de aplicaciones y despliegues en CodeDeploy

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, elija Applications (Aplicaciones).
3. Elija el nombre de la aplicación que desea eliminar (por ejemplo, CodeDeploy-OnPrem-App) y, a continuación, elija Delete application (Eliminar aplicación).
4. Cuando se le solicite, introduzca el nombre de la aplicación para confirmar que desea eliminarla y seleccione Delete (Eliminar).

No puede usar la AWS CodeDeploy consola para anular el registro de la instancia local o desinstalar el agente. CodeDeploy Siga las instrucciones en [Para cancelar el registro de la instancia on-premises y eliminar el usuario de IAM](#) .

Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling

En este tutorial, se utilizará CodeDeploy para implementar una revisión de la aplicación en un grupo de Auto Scaling. Amazon EC2 Auto Scaling lanza las instancias de Amazon EC2 con condiciones predefinidas y, a continuación, las termina cuando ya no se necesitan. Amazon EC2 Auto Scaling puede ayudar a CodeDeploy escalar al garantizar que siempre tenga el número correcto de instancias de Amazon EC2 disponibles para gestionar la carga de las implementaciones. Para obtener información sobre la integración de Auto Scaling de Amazon EC2 con CodeDeploy, consulte [Integración CodeDeploy con Amazon EC2 Auto Scaling](#)

Temas

- [Requisitos previos](#)
- [Paso 1: Crear y configurar el grupo de escalado automático](#)
- [Paso 2: Implementar la aplicación en el grupo de escalado automático](#)
- [Paso 3: Comprobar los resultados](#)
- [Paso 4: Aumentar el número de instancias de Amazon EC2 en el grupo de escalado automático](#)
- [Paso 5: Comprobar los resultados de nuevo](#)
- [Paso 6: limpiar](#)

Requisitos previos

Para seguir en este tutorial:

- Complete todos los pasos [Empezar con CodeDeploy](#), incluida la configuración AWS CLI y creación de un perfil de instancia de IAM (**CodeDeployDemo-EC2-Instance-Profile**) y un rol de servicio (**CodeDeployDemo**). Un rol de servicio es un tipo especial de rol de IAM que da a un servicio permiso para actuar en nombre de un usuario.
- Si crea un grupo de escalado automático con una plantilla de inicio, debe añadir los siguientes permisos:

- `ec2:RunInstances`
- `ec2:CreateTags`
- `iam:PassRole`

Para obtener más información, consulte [Paso 2: Crear un rol de servicio](#), [Creación de una plantilla de lanzamiento para un grupo de escalado automático](#) y [Compatibilidad con las plantillas de lanzamiento](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

- Cree y utilice una revisión que sea compatible con una instancia de Ubuntu Server y CodeDeploy. Para su revisión, puede realizar una de las siguientes acciones:
 - Cree y utilice la revisión de ejemplo en [Paso 2: Crear una revisión de aplicación de ejemplo](#) en el tutorial [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#).
 - Para crear su propia revisión, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).
- Cree un grupo de seguridad denominado **CodeDeployDemo-AS-SG** con la siguiente regla de entrada:
 - Tipo: HTTP
 - Origen: Cualquiera

Esto es necesario para ver la aplicación y comprobar que la implementación se ha realizado correctamente. Para obtener información sobre cómo crear un grupo de seguridad, consulte [Creación de un grupo de seguridad](#) en la Guía del usuario de Amazon EC2.

Paso 1: Crear y configurar el grupo de escalado automático

En este paso, creará un grupo de escalado automático que contenga una única instancia de Amazon EC2 de Amazon Linux, RHEL o Windows Server. En un paso posterior, indicará a Amazon EC2 Auto Scaling que añada una instancia más de Amazon EC2 e implementará su revisión CodeDeploy en ella.

Temas

- [Para crear y configurar el grupo de escalado automático \(CLI\)](#)
- [Para crear y configurar el grupo de escalado automático \(consola\)](#)

Para crear y configurar el grupo de escalado automático (CLI)

1. Llame al comando `create-launch-template` para crear una plantilla de lanzamiento de Amazon EC2.

Antes de invocar este comando, necesitará el ID de una AMI adecuada para este tutorial, representado por el marcador de posición *image-id*. También necesitará el nombre de un par de claves de instancia de Amazon EC2 para permitir el acceso a la instancia de Amazon EC2, representado por el marcador de posición *key-name*.

Para obtener el ID de una AMI adecuada para este tutorial:

- a. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
- b. En el panel de navegación, en Instances, elija Instances y seleccione Launch Instance.
- c. En la pestaña Inicio rápido de la página Elija una imagen de máquina de Amazon, anote el ID de la AMI junto a AMI de Amazon Linux 2, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS o Microsoft Windows Server 2012 R2.

Note

Si tiene una versión personalizada de una AMI que sea compatible CodeDeploy, selecciónela aquí en lugar de navegar por la pestaña Inicio rápido. Para obtener información sobre el uso de una AMI personalizada con CodeDeploy Amazon EC2 Auto Scaling, consulte. [Uso de una AMI personalizada con CodeDeploy Amazon EC2 Auto Scaling](#)

Como par de claves de la instancia de Amazon EC2, indique el nombre del par de claves de su instancia de Amazon EC2.

Llame al comando `create-launch-template`.

En máquinas Linux, macOS o Unix locales:

```
aws ec2 create-launch-template \  
  --launch-template-name CodeDeployDemo-AS-Launch-Template \  
  --launch-template-data file://config.json
```

El contenido del archivo `config.json`:

```
{
  "InstanceType": "t1.micro",
  "ImageId": "image-id",
  "IamInstanceProfile": {
    "Name": "CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName": "key-name"
}
```

En los equipos locales con Windows:

```
aws ec2 create-launch-template --launch-template-name CodeDeployDemo-AS-Launch-Template --launch-template-data file://config.json
```

El contenido del archivo `config.json`:

```
{
  "InstanceType": "t1.micro",
  "ImageId": "image-id",
  "IamInstanceProfile": {
    "Name": "CodeDeployDemo-EC2-Instance-Profile"
  },
  "KeyName": "key-name"
}
```

Estos comandos, junto con el `config.json` archivo, crean una plantilla de lanzamiento de Amazon EC2 denominada `CodeDeployDemo-as-Launch-template` para su grupo de Auto Scaling que se creará en un paso siguiente en función del tipo de instancia `t1.micro` de Amazon EC2. En función de los datos introducidos para `ImageId`, `IamInstanceProfile` y `KeyName`, la plantilla de lanzamiento también especifica el ID de la AMI, el nombre del perfil de instancia asociado al rol de IAM que se va a transferir a las instancias en el momento del lanzamiento y el par de claves de Amazon EC2 que se debe utilizar al conectarse a las instancias.

2. Llame al comando `create-auto-scaling-group` para crear un grupo de escalado automático. Necesitará el nombre de una de las zonas de disponibilidad de una de las regiones enumeradas en [Regiones y puntos de conexión](#) en la Referencia general de AWS, representada por el marcador de posición *availability-zone*.

Note

Para ver una lista de las zonas de disponibilidad de una región, ejecute este comando:

```
aws ec2 describe-availability-zones --region region-name
```

Por ejemplo, para ver una lista con las zonas de disponibilidad de la región Oeste de EE. UU. (Oregón), llame al comando siguiente:

```
aws ec2 describe-availability-zones --region us-west-2
```

Para obtener una lista de los identificadores de nombres de región, consulte [Nombres de bucket del kit de recursos por región](#).

En máquinas Linux, macOS o Unix locales:

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name CodeDeployDemo-AS-Group \  
  --launch-template CodeDeployDemo-AS-Launch-Template,Version='$Latest' \  
  --min-size 1 \  
  --max-size 1 \  
  --desired-capacity 1 \  
  --availability-zones availability-zone \  
  --tags Key=Name,Value=CodeDeployDemo,PropagateAtLaunch=true
```

En los equipos locales con Windows:

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name  
  CodeDeployDemo-AS-Group --launch-template LaunchTemplateName=CodeDeployDemo-  
  AS-Launch-Template,Version="$Latest" --min-size 1 --max-size 1 --  
  desired-capacity 1 --availability-zones availability-zone --tags  
  Key=Name,Value=CodeDeployDemo,PropagateAtLaunch=true
```

Estos comandos crean un grupo de escalado automático denominado **CodeDeployDemo-AS-Group** en función del nombre de la plantilla de lanzamiento de Amazon EC2 **CodeDeployDemo-AS-Launch-Template**. Este grupo de escalado automático tiene una única instancia de Amazon EC2 y se crea en la zona de disponibilidad especificada. Cada

instancia de este grupo de escalado automático tendrá la etiqueta Name=CodeDeployDemo. La etiqueta se utilizará al instalar el agente más adelante. CodeDeploy

3. Llame al comando describe-auto-scaling-groups con **CodeDeployDemo-AS-Group**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
LifecycleState]" --output text
```

No continúe hasta que los valores devueltos indiquen Healthy y InService.

4. Las instancias de su grupo de Auto Scaling deben tener el CodeDeploy agente instalado para poder usarse en CodeDeploy las implementaciones. Instale el CodeDeploy agente llamando al create-association comando desde AWS Systems Manager con las etiquetas que se agregaron al crear el grupo Auto Scaling.

```
aws ssm create-association \
--name AWS-ConfigureAWSPackage \
--targets Key=tag:Name,Values=CodeDeployDemo \


--parameters action=Install, name=AWSCodeDeployAgent \
--schedule-expression "cron(0 2 ? * SUN *)"
```

Este comando crea una asociación en Systems Manager State Manager que instalará el CodeDeploy agente en todas las instancias del grupo Auto Scaling y, a continuación, intentará actualizarlo a las 2:00 todos los domingos por la mañana. Para obtener más información sobre el CodeDeploy agente, consulte [Trabajar con el CodeDeploy agente](#). Para obtener más información acerca de Systems Manager, consulte [¿Qué es AWS Systems Manager?](#).

Para crear y configurar el grupo de escalado automático (consola)

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En la barra de navegación global, asegúrese de que esté seleccionada una de las regiones enumeradas en [Región y puntos de conexión](#) en Referencia general de AWS. Los recursos de Auto Scaling de Amazon EC2 están vinculados a la región que especifique y solo CodeDeploy se admiten en determinadas regiones.
3. En el panel de navegación, en Instancias, seleccione Plantillas de lanzamiento.
4. Elija Crear plantilla de lanzamiento.

5. En Nombre y descripción de la plantilla de lanzamiento, para Nombre de plantilla de lanzamiento, introduzca **CodeDeployDemo-AS-Launch-Template**. Deje los valores predeterminados para el resto de los campos.
6. En el cuadro de diálogo Imagen de máquina de Amazon (AMI), haga clic en el menú desplegable situado debajo de AMI y elija una AMI que funcione con este tutorial:
 - En la pestaña Inicio rápido del menú desplegable AMI, elija uno de los siguientes: AMI de Amazon Linux 2, Red Hat Enterprise Linux 7.1, Ubuntu Server 14.04 LTS o Microsoft Windows Server 2012 R2.

 Note

Si tiene una versión personalizada de una AMI que sea compatible CodeDeploy, selecciónela aquí en lugar de navegar por la pestaña Inicio rápido. Para obtener información sobre el uso de una AMI personalizada con CodeDeploy Amazon EC2 Auto Scaling, consulte. [Uso de una AMI personalizada con CodeDeploy Amazon EC2 Auto Scaling](#)

7. En Tipo de instancia, seleccione el menú desplegable y elija t1.micro. Puede utilizar la barra de búsqueda para encontrarla más rápidamente.
8. En el cuadro de diálogo Par de claves (inicio de sesión), seleccione Elegir un par de claves existente. En la lista desplegable Seleccionar un par de claves, elija el par de claves de la instancia de Amazon EC2 que ha creado o utilizado en los pasos anteriores.
9. En el cuadro de diálogo Configuración de red, elija Nube pública virtual (VPC).

En el menú desplegable Grupos de seguridad, elija el grupo de seguridad que creó en la [sección de requisitos previos del tutorial](#) (**CodeDeployDemo-AS-SG**).

10. Amplíe el cuadro de diálogo de Detalles avanzados. En el menú desplegable de Perfil de instancia de IAM, seleccione el rol de IAM que creó anteriormente (**CodeDeployDemo-EC2-Instance-Profile**) en Perfil de instancia de IAM.

Deje el resto de las opciones predeterminadas.

11. Elija Crear plantilla de inicialización.
12. En el cuadro de diálogo Próximos pasos, elija Crear grupo de escalado automático.
13. En la página Elegir una plantilla de lanzamiento o configuración, para Nombre de grupo de escalado automático, ingrese **CodeDeployDemo-AS-Group**.

14. En el cuadro de diálogo Plantilla de lanzamiento, debe ingresar su plantilla de lanzamiento (**CodeDeployDemo-AS-Launch-Template**); de lo contrario, selecciónela en el menú desplegable. Deje los valores predeterminados y elija Siguiente.
15. En la página Elegir opciones de lanzamiento de las instancias, en Red, en la opción VPC, elija la VPC predeterminada. A continuación, en Zonas de disponibilidad y subredes, elija una subred predeterminada. Debe crear una VPC si no puede elegir la predeterminada. Para obtener más información, consulte [Introducción a Amazon VPC](#).
16. En la sección Instance type requirements (Requisitos del tipo de instancia), utilice la configuración predeterminada para simplificar este paso. (No anule la plantilla de lanzamiento). En este tutorial, solo lanzará instancias bajo demanda con el tipo de instancia especificado en la plantilla de lanzamiento.
17. Elija Next (Siguiente) para ir a la página Configure advanced options (Configuración de opciones avanzadas).
18. Mantenga los valores predeterminados y elija Siguiente.
19. En la página Configurar el tamaño del grupo y las políticas de escalado, mantenga los valores predeterminados de Tamaño de grupo de 1. Elija Siguiente.
20. Omita el paso para configurar notificaciones y elija Siguiente.
21. En la página Añadir etiquetas, añada una etiqueta para utilizarla cuando instale el CodeDeploy agente más adelante. Seleccione Agregar etiqueta.
 - a. En Clave, introduzca **Name**.
 - b. En Valor, escriba **CodeDeployDemo**.

Elija Siguiente.

22. Revise la información del grupo de escalado automático en la página Revisión y luego elija Crear grupo de escalado automático.
23. En la barra de navegación, con Grupos de escalado automático seleccionado, elija **CodeDeployDemo-AS-Group** y luego seleccione la pestaña Administración de instancias. No continúe hasta que el valor de InServiceaparezca en la columna Ciclo de vida y el valor de Healthy aparezca en la columna Estado de salud.
24. Instale el CodeDeploy agente siguiendo los pasos que se indican en [Instalar el CodeDeploy agente](#) y utilizar las etiquetas de Name=CodeDeployDemo instancia.

Paso 2: Implementar la aplicación en el grupo de escalado automático

En este paso, implementará la revisión en la única instancia de Amazon EC2 del grupo de escalado automático.

Temas

- [Para crear la implementación \(CLI\)](#)
- [Para crear la implementación \(consola\)](#)

Para crear la implementación (CLI)

1. Ejecute el comando `create-application` para crear una aplicación denominada **SimpleDemoApp**:

```
aws deploy create-application --application-name SimpleDemoApp
```

2. Ya debe haber creado un rol de servicio siguiendo las instrucciones del [Paso 2: Crear un rol de servicio para CodeDeploy](#). El rol de servicio le dará CodeDeploy permiso para acceder a sus instancias de Amazon EC2 para expandir (leer) sus etiquetas. Necesitará el rol de servicio ARN. Para obtener el ARN del rol de servicio, siga las instrucciones de [Obtención del ARN del rol de servicio \(CLI\)](#).
3. Ahora que tiene un ARN de rol de servicio, ejecute el comando `create-deployment-group` para crear un grupo de implementación denominado **SimpleDemoDG** asociado a la aplicación **SimpleDemoApp**, utilizando el grupo de escalado automático **CodeDeployDemo-AS-Group** y la configuración de implementación **CodeDeployDefault.OneAtATime** con el ARN del rol de servicio especificado.

Note

El `create-deployment-group` comando permite crear activadores que provoquen el envío de notificaciones de Amazon SNS a los suscriptores del tema sobre eventos específicos en implementaciones e instancias. El comando también admite opciones para revertir automáticamente las implementaciones y configurar alarmas para detener las implementaciones cuando se cumplen los umbrales de monitoreo de CloudWatch las alarmas de Amazon. Los comandos para estas acciones no se incluyen en este tutorial.

En máquinas Linux, macOS o Unix locales:

```
aws deploy create-deployment-group \  
  --application-name SimpleDemoApp \  
  --auto-scaling-groups CodeDeployDemo-AS-Group \  
  --deployment-group-name SimpleDemoDG \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --service-role-arn service-role-arn
```

En los equipos locales con Windows:

```
aws deploy create-deployment-group --application-name SimpleDemoApp --auto-scaling-  
groups CodeDeployDemo-AS-Group --deployment-group-name SimpleDemoDG --deployment-  
config-name CodeDeployDefault.OneAtATime --service-role-arn service-role-arn
```

4. Ejecute el comando `create-deployment` para crear una implementación asociada a la aplicación **SimpleDemoApp**, la configuración de implementación **CodeDeployDefault.OneAtATime** y el grupo de implementaciones **SimpleDemoDG**, utilizando la revisión que se encuentra en el lugar especificado.

Para instancias Amazon EC2 de Amazon Linux y RHEL, llamadas desde máquinas locales Linux, macOS o Unix

```
aws deploy create-deployment \  
  --application-name SimpleDemoApp \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name SimpleDemoDG \  
  --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/  
SampleApp_Linux.zip
```

bucket-name es el nombre del depósito de Amazon S3 que contiene los archivos del kit CodeDeploy de recursos de su región. Por ejemplo, en el caso de la región Este de EE. UU. (Ohio), sustituya *bucket-name* por `aws-codedeploy-us-east-2`. Para obtener una lista de los nombres de los buckets, consulte [Nombres de bucket del kit de recursos por región](#).

Para instancias Amazon EC2 de Amazon Linux y RHEL, llamadas desde máquinas locales Windows

```
aws deploy create-deployment --application-name SimpleDemoApp --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name SimpleDemoDG --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/SampleApp_Linux.zip
```

bucket-name es el nombre del depósito de Amazon S3 que contiene los archivos del kit CodeDeploy de recursos de su región. Por ejemplo, en el caso de la región Este de EE. UU. (Ohio), sustituya *bucket-name* por `aws-codedeploy-us-east-2`. Para obtener una lista de los nombres de los buckets, consulte [Nombres de bucket del kit de recursos por región](#).

Para instancias Amazon EC2 de Windows Server, llamadas desde máquinas locales Linux, macOS o Unix

```
aws deploy create-deployment \  
  --application-name SimpleDemoApp \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name SimpleDemoDG \  
  --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/  
SampleApp_Windows.zip
```

bucket-name es el nombre del depósito de Amazon S3 que contiene los archivos del kit CodeDeploy de recursos de su región. Por ejemplo, en el caso de la región Este de EE. UU. (Ohio), sustituya *bucket-name* por `aws-codedeploy-us-east-2`. Para obtener una lista de los nombres de los buckets, consulte [Nombres de bucket del kit de recursos por región](#).

Para instancias Amazon EC2 de Windows Server, llamadas desde máquinas locales Windows

```
aws deploy create-deployment --application-name SimpleDemoApp --deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name SimpleDemoDG --s3-location bucket=bucket-name,bundleType=zip,key=samples/latest/SampleApp_Windows.zip
```

bucket-name es el nombre del depósito de Amazon S3 que contiene los archivos del kit CodeDeploy de recursos de su región. Por ejemplo, en el caso de la región Este de EE. UU. (Ohio), sustituya *bucket-name* por `aws-codedeploy-us-east-2`. Para obtener una lista de los nombres de los buckets, consulte [Nombres de bucket del kit de recursos por región](#).

Note

Actualmente, CodeDeploy no proporciona una revisión de muestra para implementarla en instancias Amazon EC2 de Ubuntu Server. Para crear su propia revisión, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).

5. Ejecute el comando `get-deployment` para asegurarse de que la implementación se ha realizado correctamente.

Antes de ejecutar este comando, necesitará el ID de la implementación, que debe haber obtenido tras ejecutar el comando `create-deployment`. Si necesita obtener de nuevo el ID de la implementación, ejecute el comando `list-deployments` para la aplicación **SimpleDemoApp** y el grupo de implementaciones **SimpleDemoDG**:

```
aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-name SimpleDemoDG --query "deployments" --output text
```

Llame ahora al comando `get-deployment` utilizando el ID de la implementación:

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.status" --output text
```

No continúe hasta que el valor devuelto sea `Succeeded`.

Para crear la implementación (consola)

1. Ya debe haber creado un rol de servicio siguiendo las instrucciones del [Paso 2: Crear un rol de servicio para CodeDeploy](#). El rol de servicio le dará CodeDeploy permiso para acceder a sus instancias para expandir (leer) sus etiquetas. Antes de usar la CodeDeploy consola para implementar la revisión de la aplicación, necesitará el ARN del rol de servicio. Para obtener el ARN del rol de servicio, siga las instrucciones de [Obtención del ARN del rol de servicio \(consola\)](#).
2. Ahora que tiene el ARN del rol de servicio, puede usar la CodeDeploy consola para implementar la revisión de la aplicación.

Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

3. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
4. Elija Crear aplicación.
5. Elija Custom application (Aplicación personalizada).
6. En Application name (Nombre de aplicación), escriba **SimpleDemoApp**.
7. En Plataforma de informática, elija EC2/en las instalaciones.
8. Elija Crear aplicación.
9. En la pestaña Deployment groups (Grupos de implementaciones), elija Create deployment group (Crear grupo de implementaciones).
10. En Nombre de grupo de implementación, escriba **SimpleDemoDG**.
11. En Service Role (Rol de servicio), elija el nombre del rol de servicio.
12. En Tipo de implementación, elija Local.
13. En Configuración del entorno, seleccione Grupos de escalado automático y, a continuación, elija **CodeDeployDemo-AS-Group**.
14. En Configuración de implementación, elija CodeDeployDefault. OneAtA Time.
15. Desactive Enable load balancing (Habilitar balanceo de carga).
16. Elija Crear grupo de implementación.
17. En la página del grupo de implementaciones, elija Create deployment (Crear implementación).
18. En Tipo de revisión, elija Mi aplicación está almacenada en Amazon S3.
19. En Revision location (Ubicación de la revisión), introduzca la ubicación de la aplicación de ejemplo para el sistema operativo y la región.

Para instancias Amazon EC2 de Amazon Linux y RHEL

Región	Ubicación de aplicación de ejemplo
Región del Este de EE. UU. (Ohio)	<code>http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip</code>

Región	Ubicación de aplicación de ejemplo
Región del Este de EE. UU (Norte de Virginia)	<code>http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip</code>
Región del Oeste de EE. UU (Norte de California)	<code>http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip</code>
Región del Oeste de EE. UU (Oregón)	<code>http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip</code>
Región de Canadá (centro)	<code>http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip</code>
Región de Europa (Irlanda)	<code>http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip</code>
Región de Europa (Londres)	<code>http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip</code>
Región Europa (París)	<code>http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip</code>

Región	Ubicación de aplicación de ejemplo
Región de Europa (Fráncfort)	http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip
Región Israel (Tel Aviv)	https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Linux.zip
Región de Asia-Pacífico (Hong Kong)	https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Linux.zip
Asia Pacífico (Tokio)	http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip
Región de Asia-Pacífico (Seúl)	http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip
Región de Asia-Pacífico (Singapur)	http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip
Región de Asia-Pacífico (Sídney)	http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip

Región	Ubicación de aplicación de ejemplo
Región de Asia-Pacífico (Melbourne)	https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Linux.zip
Región de Asia-Pacífico (Bombay)	http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip
Región de América del Sur (São Paulo)	http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip

Instancias de Amazon EC2 con Windows Server

Región	Ubicación de aplicación de ejemplo
Región del Este de EE. UU. (Ohio)	http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip
Región del Este de EE. UU. (Norte de Virginia)	http://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip
Región del Oeste de EE. UU. (Norte de California)	http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip

Región	Ubicación de aplicación de ejemplo
Región del Oeste de EE. UU (Oregón)	<code>http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip</code>
Región de Canadá (centro)	<code>http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip</code>
Región de Europa (Irlanda)	<code>http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip</code>
Región de Europa (Londres)	<code>http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip</code>
Región Europa (París)	<code>http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip</code>
Región de Europa (Fráncfort)	<code>http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip</code>
Región Israel (Tel Aviv)	<code>https://aws-codedeploy-il-central-1.s3.il-central-1.amazonaws.com/samples/latest/SampleApp_Windows.zip</code>

Región	Ubicación de aplicación de ejemplo
Región de Asia-Pacífico (Hong Kong)	<code>https://aws-codedeploy-ap-east-1.s3.ap-east-1.amazonaws.com/samples/latest/SampleApp_Windows.zip</code>
Región de Asia-Pacífico (Seúl)	<code>http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip</code>
Región de Asia-Pacífico (Singapur)	<code>http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip</code>
Región de Asia-Pacífico (Sídney)	<code>http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip</code>
Región de Asia-Pacífico (Melbourne)	<code>https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/samples/latest/SampleApp_Windows.zip</code>
Región de Asia-Pacífico (Bombay)	<code>http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip</code>
Región de América del Sur (São Paulo)	<code>http://s3-sa-east-1.amazonaws.com/aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip</code>

Para instancias de Amazon EC2 de Ubuntu Server

Escriba la ubicación de la revisión de la aplicación personalizada almacenada en Amazon S3.

20. Deje Deployment description (Descripción de implementación) en blanco.
21. Expanda Advanced (Avanzadas).
22. Elija Crear implementación.

Note

Si aparece Failed (Error) en lugar de Succeeded (Correcto), puede ser conveniente probar alguna de las técnicas descritas en [Monitorización y solución de problemas de la implementación](#) (utilizando **SimpleDemoApp** como nombre de la aplicación y **SimpleDemoDG** como nombre del grupo de implementaciones).

Paso 3: Comprobar los resultados

En este paso, comprobará si la **SimpleDemoApp** revisión CodeDeploy está instalada en la única instancia de Amazon EC2 del grupo Auto Scaling.

Temas

- [Para comprobar los resultados \(CLI\)](#)
- [Para comprobar los resultados \(consola\)](#)

Para comprobar los resultados (CLI)

En primer lugar, necesitará el DNS público de la instancia de Amazon EC2.

Utilice el AWS CLI para obtener el DNS público de la instancia de Amazon EC2 del grupo Auto Scaling mediante una llamada al describe-instances comando.

Antes de llamar a este comando, necesitará el ID de la instancia de Amazon EC2. Para obtener el ID, ejecute el comando describe-auto-scaling-groups con **CodeDeployDemo-AS-Group** igual que hizo anteriormente:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Llame ahora al comando describe-instances:

```
aws ec2 describe-instances --instance-id instance-id --query "Reservations[0].Instances[0].PublicDnsName" --output text
```

El valor devuelto es el DNS público de la instancia de Amazon EC2.

Con un navegador web, muestre la SimpleDemoApp revisión implementada en esa instancia de Amazon EC2 mediante una URL como la siguiente:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si ve la página de felicitaciones, ¡ha utilizado CodeDeploy correctamente la implementación de una revisión en una sola instancia de Amazon EC2 en un grupo de Auto Scaling!

A continuación, agregará una instancia de Amazon EC2 al grupo de escalado automático. Una vez que Amazon EC2 Auto Scaling añada la instancia de Amazon EC2 CodeDeploy , implementará la revisión en la nueva instancia.

Para comprobar los resultados (consola)

En primer lugar, necesitará el DNS público de la instancia de Amazon EC2.

Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.

En el panel de navegación de Amazon EC2, en Escalado automático, elija Grupos de escalado automático y, a continuación, elija la entrada **CodeDeployDemo-AS-Group**.

En la pestaña Instancias, elija el ID de la instancia de Amazon EC2 en la lista.

En la página Instances, en la pestaña Description, tome nota del valor de Public DNS. Debe tener un aspecto similar al siguiente: **ec2-01-234-567-890.compute-1.amazonaws.com**.

Con un navegador web, muestre la SimpleDemoApp revisión implementada en esa instancia de Amazon EC2 mediante una URL como la siguiente:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si ve la página de felicitaciones, ¡ha utilizado CodeDeploy correctamente la implementación de una revisión en una sola instancia de Amazon EC2 en un grupo de Auto Scaling!

A continuación, agregará una instancia de Amazon EC2 al grupo de escalado automático. Una vez que Amazon EC2 Auto Scaling añada la instancia de Amazon EC2 CodeDeploy, implementará la revisión en la nueva instancia de Amazon EC2.

Paso 4: Aumentar el número de instancias de Amazon EC2 en el grupo de escalado automático

En este paso, indica al grupo de escalado automático que cree una instancia Amazon EC2 adicional. Después de que Amazon EC2 Auto Scaling cree la instancia, CodeDeploy implementa la revisión en ella.

Temas

- [Para escalar horizontalmente la cantidad de instancias de Amazon EC2 en el grupo de escalado automático \(CLI\)](#)
- [Para aumentar el número de instancias de Amazon EC2 del grupo de implementación \(consola\)](#)

Para escalar horizontalmente la cantidad de instancias de Amazon EC2 en el grupo de escalado automático (CLI)

1. Llame al comando `update-auto-scaling-group` para aumentar las instancias de Amazon EC2 del grupo de escalado automático denominado **CodeDeployDemo-AS-Group** de una a dos.

En máquinas Linux, macOS o Unix locales:

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name CodeDeployDemo-AS-Group \  
  --min-size 2 \  
  --max-size 2 \  
  --desired-capacity 2
```

En los equipos locales con Windows:

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-AS-Group --min-size 2 --max-size 2 --desired-capacity 2
```

2. Asegúrese de que el grupo de escalado automático tenga ahora dos instancias de Amazon EC2. Llame al comando `describe-auto-scaling-groups` con **CodeDeployDemo-AS-Group**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].[HealthStatus,
LifecycleState]" --output text
```

No continúe hasta que los dos valores devueltos indiquen `Healthy` y `InService`.

Para aumentar el número de instancias de Amazon EC2 del grupo de implementación (consola)

1. En el panel de navegación de Amazon EC2, en Escalado automático, elija Grupos de escalado automático y, a continuación, elija **CodeDeployDemo-AS-Group**.
2. Seleccione Acciones y, a continuación, Editar.
3. En la pestaña Details (Detalles), en los cuadros Desired (Objetivos), Min (Mínimo) y Max (Máximo), escriba **2** y luego elija Save (Guardar).
4. Seleccione la pestaña Instances. La nueva instancia de Amazon EC2 debe aparecer en la lista. (Si la instancia no aparece, puede que tenga que pulsar el botón Refresh varias veces). No continúe hasta que el valor de `InService` aparezca en la columna Ciclo de vida y el valor de `Healthy` aparezca en la columna Estado de salud.

Paso 5: Comprobar los resultados de nuevo

En este paso, comprobará si está CodeDeploy instalada la SimpleDemoApp revisión en la nueva instancia del grupo Auto Scaling.

Temas

- [Para comprobar los resultados de la implementación automática \(CLI\)](#)
- [Para comprobar los resultados de la implementación automática \(consola\)](#)

Para comprobar los resultados de la implementación automática (CLI)

1. Antes de ejecutar el comando `get-deployment`, necesitará el ID de la implementación automática. Para obtener el ID, ejecute el comando `list-deployments` con la aplicación **SimpleDemoApp** y el grupo de implementaciones **SimpleDemoDG**:

```
aws deploy list-deployments --application-name SimpleDemoApp --deployment-group-name SimpleDemoDG --query "deployments" --output text
```

Debe haber dos ID de implementación. Utilice el que aún no haya empleado en una llamada al comando `get-deployment`:

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.[status, creator]" --output text
```

Además del estado de la implementación, en la salida del comando debe aparecer `autoScaling`. (`autoScaling` significa que Amazon EC2 Auto Scaling ha creado la implementación).

No continúe hasta que el estado de la implementación aparezca como `Succeeded`.

2. Antes de llamar al comando `describe-instances`, necesitará el ID de la instancia de Amazon EC2 nueva. Para obtener este ID, ejecute de nuevo el comando `describe-auto-scaling-groups` con **CodeDeployDemo-AS-Group**:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names CodeDeployDemo-AS-Group --query "AutoScalingGroups[0].Instances[*].InstanceId" --output text
```

Ahora ejecute el comando `describe-instances`:

```
aws ec2 describe-instances --instance-id instance-id --query "Reservations[0].Instances[0].PublicDnsName" --output text
```

En la salida del comando `describe-instances`, anote el DNS público de la instancia de Amazon EC2 nueva.

3. En un navegador web, muestre la revisión `SimpleDemoApp` implementada en la instancia de Amazon EC2 mediante una URL como la siguiente:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si aparece la página de felicitaciones, ¡ CodeDeploy solía implementar una revisión en una instancia de Amazon EC2 ampliada en un grupo de Auto Scaling!

Para comprobar los resultados de la implementación automática (consola)

1. [Inicie sesión AWS Management Console y abra la CodeDeploy consola en https://console.aws.amazon.com/codedeploy.](https://console.aws.amazon.com/codedeploy)

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y después elija Implementaciones.
3. Elija el ID de la implementación que ha creado Amazon EC2 Auto Scaling.
4. La página Deployment (Implementación) muestra información sobre la implementación. Normalmente, debería crear una implementación por su cuenta, pero Amazon EC2 Auto Scaling ya ha creado una por usted para implementar la revisión en la instancia de Amazon EC2 nueva.
5. Cuando aparezca Succeeded (Correcto) en la parte superior de la página, compruebe los resultados en la instancia. Primero debe obtener el DNS público de la instancia:
6. En el panel de navegación de Amazon EC2, en Escalado automático, elija Grupos de escalado automático y, a continuación, elija la entrada **CodeDeployDemo-AS-Group**.
7. En la pestaña Instances, elija el ID de la nueva instancia de Amazon EC2.
8. En la página Instances, en la pestaña Description, tome nota del valor de Public DNS. Debe tener un aspecto similar al siguiente: **ec2-01-234-567-890.compute-1.amazonaws.com**.

Muestre la SimpleDemoApp revisión implementada en la instancia utilizando una URL como la siguiente:

```
http://ec2-01-234-567-890.compute-1.amazonaws.com
```

Si aparece la página de felicitaciones, ¡ CodeDeploy solía implementar una revisión en una instancia de Amazon EC2 ampliada en un grupo de Auto Scaling!

Paso 6: limpiar

En este paso eliminará el grupo de escalado automático para evitar cargos recurrentes por los recursos utilizados durante este tutorial. Si lo desea, puede eliminar los registros de los componentes de configuración y CodeDeploy despliegue de Auto Scaling.

Temas

- [Para limpiar los recursos \(CLI\)](#)
- [Para limpiar los recursos \(consola\)](#)

Para limpiar los recursos (CLI)

1. Elimine el grupo de escalado automático llamando al comando `delete-auto-scaling-group` con **CodeDeployDemo-AS-Group**. Esto también terminará las instancias de Amazon EC2.

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name CodeDeployDemo-AS-Group --force-delete
```

2. Si lo desea, puede eliminar la plantilla de lanzamiento de escalado automático llamando al comando `delete-launch-template` con la configuración de lanzamiento **CodeDeployDemo-AS-Launch-Template**:

```
aws ec2 delete-launch-template --launch-template-name CodeDeployDemo-AS-Launch-Template
```

3. Si lo desea, elimine la aplicación CodeDeploy ejecutando el `delete-application` comando correspondiente a la aplicación nombrada **SimpleDemoApp**. Esto también eliminará los registros de implementación, grupo de implementaciones y revisión asociados.

```
aws deploy delete-application --application-name SimpleDemoApp
```

4. Para eliminar la asociación de Systems Manager State Manager, llame al comando `delete-association`.

```
aws ssm delete-association --association-id association-id
```

Para obtener el valor de *id-asociación*, llame al comando describe-association.

```
aws ssm describe-association --name AWS-ConfigureAWSPackage --targets
Key=tag:Name,Values=CodeDeployDemo
```

Para limpiar los recursos (consola)

Para eliminar el grupo de escalado automático, con lo que también se terminan las instancias de Amazon EC2:

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. En el panel de navegación de Amazon EC2, en Escalado automático, elija Grupos de escalado automático y, a continuación, elija la entrada **CodeDeployDemo-AS-Group**.
3. Elija Actions, seleccione Delete y luego seleccione Yes, Delete.

(Opcional) Para eliminar la plantilla de lanzamiento:

1. En la barra de navegación, en Escalado automático, elija Configuraciones de lanzamiento y, a continuación, elija **CodeDeployDemo-AS-Launch-Template**.
 2. Elija Actions, seleccione Delete launch configuration y luego seleccione Yes, Delete.
-
1. Si lo desea, elimine la aplicación de CodeDeploy. Esto también eliminará los registros de implementación, grupo de implementaciones y revisión asociados. Abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy>.
 2. Inicie sesión AWS Management Console y abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy>.

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.

3. En la lista de aplicaciones, elija SimpleDemoApp.
4. En la página Application details (Detalles de la aplicación), elija Delete application (Eliminar aplicación).
5. Cuando se le pregunte, escriba **Delete** y, a continuación, elija Delete (Eliminar).

Para eliminar la asociación de Systems Manager State Manager:

1. Abra la AWS Systems Manager consola en <https://console.aws.amazon.com/systems-manager>.
2. En el panel de navegación, elija Administrador de estados.
3. Elija la asociación que ha creado y, después, elija Eliminar.

Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub

En este tutorial, se utiliza CodeDeploy para implementar una revisión de aplicación de GitHub muestra en una sola instancia de Amazon EC2 que ejecute Amazon Linux, una sola instancia de Red Hat Enterprise Linux (RHEL) o una sola instancia de Windows Server. Para obtener información sobre la GitHub integración con CodeDeploy, consulte [Integración CodeDeploy con GitHub](#)

Note

También se puede utilizar CodeDeploy para implementar una revisión de una aplicación desde GitHub una instancia de Ubuntu Server. Puede utilizar el ejemplo de revisión descrito [Paso 2: Crear una revisión de aplicación de ejemplo](#) en [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#) o puede crear una revisión compatible con una instancia de Ubuntu Server y CodeDeploy. Para crear su propia revisión, consulte [Planifique una revisión para CodeDeploy](#) y [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Configurar una GitHub cuenta](#)
- [Paso 2: Crea un GitHub repositorio](#)

- [Paso 3: Sube una aplicación de muestra a tu repositorio GitHub](#)
- [Paso 4: Aprovisionar una instancia](#)
- [Paso 5: Crear una aplicación y un grupo de implementación](#)
- [Paso 6: Implementar la aplicación en la instancia](#)
- [Paso 7: Monitorizar y comprobar la implementación](#)
- [Paso 8: Eliminación](#)

Requisitos previos

Antes de empezar este tutorial, haga lo siguiente:

- Instale Git en su equipo local. Para instalar Git, consulte [Git Downloads](#).
- Complete los pasos indicados en [Empezar con CodeDeploy](#), incluida la instalación y configuración de la AWS CLI. Esto es especialmente importante si quieres usarlo AWS CLI para implementar una revisión desde GitHub la instancia.

Paso 1: Configurar una GitHub cuenta

Necesitará una GitHub cuenta para crear un GitHub repositorio en el que se almacenará la revisión. Si ya tiene una GitHub cuenta, pase a [Paso 2: Crea un GitHub repositorio](#).

1. Vaya a <https://github.com/join>.
2. Escriba un nombre de usuario, su dirección de correo electrónico y una contraseña.
3. Seleccione Registrarse y GitHub, a continuación, sigue las instrucciones.

Paso 2: Crea un GitHub repositorio

Necesitará un GitHub repositorio para almacenar la revisión.

Si ya tiene un GitHub repositorio, asegúrese de sustituirlo por su nombre a **CodeDeployGitHubDemo** lo largo de este tutorial y, a continuación, vaya directamente a [Paso 3: Sube una aplicación de muestra a tu repositorio GitHub](#) .

1. En la [página de GitHub inicio](#), realice una de las siguientes acciones:
 - En Your repositories, elija New repository.

- En la barra de navegación, seleccione Create new (+) y luego elija New repository.
2. En la página Create a new repository, haga lo siguiente:
 - En el cuadro Repository name (Nombre del repositorio), indique **CodeDeployGitHubDemo**.
 - Seleccione Public.

Note

La opción predeterminada Public implica que cualquiera puede ver el repositorio. Puede seleccionar la opción Private (Privado) para limitar quién puede ver el repositorio y confirmarlo.

- Desactive la casilla Initialize this repository with a README. En su lugar creará un archivo README .md manualmente en el siguiente paso.
 - Elija Create repository.
3. Siga las instrucciones correspondientes al tipo de su equipo local para crear el repositorio con la línea de comandos.

Note

Si ha activado la autenticación de dos factores GitHub, asegúrese de introducir su token de acceso personal en lugar de su contraseña de inicio de GitHub sesión si se le solicita una contraseña. Para obtener información, consulte [Proporcionar su código de autenticación 2FA](#).

En máquinas Linux, macOS o Unix locales:

1. Desde el terminal, ejecuta los siguientes comandos, uno *por uno*, donde *username es tu nombre* de GitHub usuario:

```
mkdir /tmp/CodeDeployGitHubDemo
```

```
cd /tmp/CodeDeployGitHubDemo
```

```
touch README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "My first commit"
```

```
git remote add origin https://github.com/user-name/CodeDeployGitHubDemo.git
```

```
git push -u origin master
```

2. Deje el terminal abierto en la ubicación `/tmp/CodeDeployGitHubDemo`.

En los equipos locales con Windows:

1. Desde el símbolo del sistema abierto como administrador, ejecute sucesivamente los comandos siguientes:

```
mkdir c:\temp\CodeDeployGitHubDemo
```

```
cd c:\temp\CodeDeployGitHubDemo
```

```
notepad README.md
```

2. En el Bloc de notas, guarde el archivo `README.md`. Cierre el Bloc de notas. Ejecute los siguientes comandos, uno a la vez, donde *username es su nombre* de GitHub usuario:

```
git init
```

```
git add README.md
```

```
git commit -m "My first commit"
```

```
git remote add origin https://github.com/user-name/CodeDeployGitHubDemo.git
```



```
git push -u origin master
```

3. Deje el símbolo del sistema abierto en la ubicación `c:\temp\CodeDeployGitHubDemo`.

Paso 3: Sube una aplicación de muestra a tu repositorio GitHub

En este paso, copiará un ejemplo de revisión de un bucket público de Amazon S3 a su GitHub repositorio. (Para simplificar, las revisiones de ejemplo incluidas para este tutorial son páginas web individuales).

Note

Si en lugar de nuestro ejemplo utiliza una revisión propia, esta deberá:

- Seguir las directrices de [Planifique una revisión para CodeDeploy](#) y [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).
- Funcionar con el tipo de instancia correspondiente.
- Sea accesible desde su GitHub panel de control.

Si su revisión cumple estos requisitos, continúe en [Paso 5: Crear una aplicación y un grupo de implementación](#).

Si vas a realizar el despliegue en una instancia de Ubuntu Server, tendrás que cargar en tu GitHub repositorio una revisión compatible con una instancia de Ubuntu Server y CodeDeploy. Para obtener más información, consulte [Planifique una revisión para CodeDeploy](#) y [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).

Temas

- [Enviar una revisión de ejemplo desde una máquina local Linux, macOS o Unix](#)
- [Envío de una revisión de ejemplo desde un equipo local con Windows](#)

Enviar una revisión de ejemplo desde una máquina local Linux, macOS o Unix

Con el terminal aún abierto, por ejemplo en la ubicación `/tmp/CodeDeployGitHubDemo`, ejecute sucesivamente los comandos siguientes:

Note

Si tiene previsto implementar en una instancia de Windows Server, sustituya `SampleApp_Windows.zip` por `SampleApp_Linux.zip` en los comandos.

(Amazon S3 copy command)

```
unzip SampleApp_Linux.zip
```

```
rm SampleApp_Linux.zip
```

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Donde *(comando copy de Amazon S3)* es uno de los siguientes:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2` para la región Este de EE. UU. (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1` para la región Este de EE. UU. (Norte de Virginia)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1` para la región Oeste de EE. UU. (Norte de California)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2` para la región Oeste de EE. UU. (Oregón)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1` para la región Canadá (centro)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1` para la región Europa (Irlanda)

- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2` para la región Europa (Londres)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3` para la región Europa (París)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1` para la región Europa (Fráncfort)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1` para la región Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1` para la región Asia Pacífico (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1` para la región Asia-Pacífico (Tokio)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2` para la región Asia-Pacífico (Seúl)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1` para la región Asia-Pacífico (Singapur)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2` para la región Asia-Pacífico (Sídney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4` para la región Asia-Pacífico (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1` para la región Asia-Pacífico (Bombay)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1` para la región América del Sur (São Paulo)

Envío de una revisión de ejemplo desde un equipo local con Windows

Con el símbolo del sistema aún abierto, por ejemplo en la ubicación `c:\temp\CodeDeployGitHubDemo`, ejecute sucesivamente los comandos siguientes:

Note

Si tiene previsto realizar la implementación en una instancia de Amazon Linux o RHEL, sustituya `SampleApp_Linux.zip` por `SampleApp_Windows.zip` en los comandos.

(Amazon S3 copy command)

Descomprima el contenido the del archivo ZIP directamente en el directorio local (por ejemplo `c:\temp\CodeDeployGitHubDemo`), no en un nuevo subdirectorio.

```
git add .
```

```
git commit -m "Added sample app"
```

```
git push
```

Donde *(comando copy de Amazon S3)* es uno de los siguientes:

- `aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Windows.zip . --region us-east-2` para la región Este de EE. UU. (Ohio)
- `aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Windows.zip . --region us-east-1` para la región Este de EE. UU. (Norte de Virginia)
- `aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Windows.zip . --region us-west-1` para la región Oeste de EE. UU. (Norte de California)
- `aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Windows.zip . --region us-west-2` para la región Oeste de EE. UU. (Oregón)
- `aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Windows.zip . --region ca-central-1` para la región Canadá (centro)
- `aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Windows.zip . --region eu-west-1` para la región Europa (Irlanda)

- `aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Windows.zip . --region eu-west-2` para la región Europa (Londres)
- `aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Windows.zip . --region eu-west-3` para la región Europa (París)
- `aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Windows.zip . --region eu-central-1` para la región Europa (Fráncfort)
- `aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Windows.zip . --region il-central-1` para la región Israel (Tel Aviv)
- `aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Windows.zip . --region ap-east-1` para la región Asia Pacífico (Hong Kong)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Windows.zip . --region ap-northeast-1` para la región Asia-Pacífico (Tokio)
- `aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Windows.zip . --region ap-northeast-2` para la región Asia-Pacífico (Seúl)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Windows.zip . --region ap-southeast-1` para la región Asia-Pacífico (Singapur)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Windows.zip . --region ap-southeast-2` para la región Asia-Pacífico (Sídney)
- `aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Windows.zip . --region ap-southeast-4` para la región Asia-Pacífico (Melbourne)
- `aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Windows.zip . --region ap-south-1` para la región Asia-Pacífico (Bombay)
- `aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Windows.zip . --region sa-east-1` para la región América del Sur (São Paulo)

Para enviar su propia revisión a una instancia de Ubuntu Server, copie la revisión en el repositorio local y, a continuación, ejecute lo siguiente:

```
git add .
```

```
git commit -m "Added Ubuntu app"  
git push
```

Paso 4: Aprovisionar una instancia

En este paso creará o configurará la instancia en la que va a implementar la aplicación de ejemplo. Puede realizar la implementación en una instancia de Amazon EC2 o en una instancia local que ejecute uno de los sistemas operativos compatibles con CodeDeploy. Para obtener información, consulte [Sistemas operativos compatibles con el agente CodeDeploy](#). (Si ya tiene una instancia configurada para su uso en CodeDeploy las implementaciones, vaya al paso siguiente).

Para aprovisionar una instancia

1. Siga las instrucciones de [Lanzamiento de una instancia de Amazon EC2 \(consola\)](#) para aprovisionar una instancia.
2. Al lanzar la instancia, no olvide especificar una etiqueta en la página Añadir etiquetas. Para obtener información detallada acerca de la forma de especificar la etiqueta, consulte [Lanzamiento de una instancia de Amazon EC2 \(consola\)](#).

Para comprobar que el CodeDeploy agente se está ejecutando en la instancia

- Siga las instrucciones de [Compruebe que el CodeDeploy agente se esté ejecutando](#) para comprobar que el agente se está ejecutando.

Una vez que hayas aprovisionado correctamente la instancia y hayas comprobado que el CodeDeploy agente se está ejecutando, continúa con el siguiente paso.

Paso 5: Crear una aplicación y un grupo de implementación

En este paso, utilizará la CodeDeploy consola o la AWS CLI para crear un grupo de aplicaciones y despliegues que utilizará para implementar la revisión de muestra desde su GitHub repositorio.

Creación de una aplicación y un grupo de implementación (consola)

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. Seleccione Create application (Crear aplicación) y, a continuación, seleccione Custom application (Aplicación personalizada).
4. En Application name (Nombre de aplicación), escriba **CodeDeployGitHubDemo-App**.
5. En Plataforma de informática, elija EC2/On-Premises.
6. Elija Crear aplicación.
7. En la pestaña Deployment groups (Grupos de implementaciones), elija Create deployment group (Crear grupo de implementaciones).
8. En Nombre de grupo de implementación, escriba **CodeDeployGitHubDemo-DepGrp**.
9. En Función de servicio, elija el nombre de la función de CodeDeploy servicio que creó en [Crear una función de servicio para CodeDeploy](#).
10. En Tipo de implementación, elija Local.
11. En Configuración de entorno, en función del tipo de instancia que utiliza, seleccione Instancias de Amazon EC2 o Instancias en las instalaciones. En Key (Clave) y Value (Valor), escriba la clave y el valor de la etiqueta de instancia que se aplicó a la instancia como parte de [Paso 4: Aprovisionar una instancia](#).
12. En Configuración de despliegue, elija CodeDeployDefault. AllatOnce.
13. En Equilibrador de carga, anule la selección de Habilitar equilibración de carga.
14. Expanda Advanced (Avanzadas).
15. En Alarms (Alarmas), seleccione Ignore alarm configuration (Ignorar las alarmas).
16. Elija Create deployment group (Crear grupo de implementaciones) y vaya al paso siguiente.

Creación de una aplicación y un grupo de implementaciones (CLI)

1. Llame al create-application comando para crear una aplicación con el CodeDeploy nombreCodeDeployGitHubDemo-App:

```
aws deploy create-application --application-name CodeDeployGitHubDemo-App
```

2. Llame al comando `create-deployment-group` para crear un grupo de implementación denominado `CodeDeployGitHubDemo-DepGrp`:
 - Si va a realizar la implementación en una instancia de Amazon EC2, *ec2-tag-key* es la clave de la etiqueta de instancia de Amazon EC2 que se aplicó a la instancia de Amazon EC2 en el [Paso 4: Aprovisionar una instancia](#).
 - Si va a realizar la implementación en una instancia de Amazon EC2, *ec2-tag-value* es el valor de la etiqueta de instancia de Amazon EC2 que se aplicó a la instancia de Amazon EC2 en el [Paso 4: Aprovisionar una instancia](#).
 - Si vas a realizar la implementación en una instancia local, *on-premises-tag-keyes* la clave de etiqueta de la instancia local que se aplicó a la instancia local como parte de ella. [Paso 4: Aprovisionar una instancia](#)
 - Si vas a realizar la implementación en una instancia local, *on-premises-tag-valuees* el valor de etiqueta de la instancia local que se aplicó a la instancia local como parte de ella. [Paso 4: Aprovisionar una instancia](#)
 - *service-role-arnes* el ARN del rol de servicio para el que creó en [Crear un rol de servicio](#). CodeDeploy (Siga las instrucciones de [Obtención del ARN del rol de servicio \(CLI\)](#) para determinar el ARN del rol de servicio).

```
aws deploy create-deployment-group --application-name CodeDeployGitHubDemo-App
--ec2-tag-filters Key=ec2-tag-key,Type=KEY_AND_VALUE,Value=ec2-tag-value --on-
premises-tag-filters Key=on-premises-tag-key,Type=KEY_AND_VALUE,Value=on-premises-
tag-value --deployment-group-name CodeDeployGitHubDemo-DepGrp --service-role-
arn service-role-arn
```

Note

El [create-deployment-group](#) comando permite crear activadores que provoquen el envío de notificaciones de Amazon SNS a los suscriptores del tema sobre eventos específicos en implementaciones e instancias. El comando también admite opciones para revertir automáticamente las implementaciones y configurar alarmas para detener las implementaciones cuando se cumplen los umbrales de monitoreo de CloudWatch las alarmas de Amazon. Los comandos para estas acciones no se incluyen en este tutorial.

Paso 6: Implementar la aplicación en la instancia

En este paso, utiliza la CodeDeploy consola o la AWS CLI para implementar la revisión de muestra del GitHub repositorio en la instancia.


Para implementar la revisión (consola)

1. En la página Deployment group details (Detalles del grupo de implementaciones), elija Create deployment (Crear implementación).
2. En Deployment group (Grupo de implementación), elija **CodeDeployGitHubDemo-DepGrp**.
3. En Tipo de revisión, elija GitHub.
4. En Conectarse a GitHub, realice una de las siguientes acciones:
 - Para crear una conexión de CodeDeploy aplicaciones a una GitHub cuenta, cierre sesión GitHub en una pestaña independiente del navegador web. En la GitHub cuenta, introduce un nombre para identificar esta conexión y, a continuación, selecciona Conectar a GitHub. La página web le pide que autorice CodeDeploy la interacción con GitHub la aplicación nombrada CodeDeployGitHubDemo-App. Continúe en el paso 5.
 - Para usar una conexión que ya haya creado, en la GitHub cuenta, seleccione su nombre y, a continuación, elija Conectar a GitHub. Continúe en el paso 7.
 - Para crear una conexión a otra GitHub cuenta, cierra sesión GitHub en otra pestaña del navegador web. Selecciona Conectarse a una GitHub cuenta diferente y, a continuación, selecciona Conectar a GitHub. Continúe en el paso 5.
5. Sigue las instrucciones de la página de inicio de sesión para iniciar sesión con tu GitHub cuenta.
6. En la página Authorize application, elija Authorize application.
7. En la página CodeDeploy Crear despliegue, en Nombre del repositorio, introduzca el nombre de GitHub usuario que utilizó para iniciar sesión, seguido de una barra inclinada (/), seguido del nombre del repositorio en el que insertó la revisión de la aplicación (por ejemplo, **my-github-user-name/CodeDeployGitHubDemo**).

Si no está seguro del valor que debe introducir o si desea especificar un repositorio diferente:

- a. En una pestaña independiente del navegador web, dirígete al [GitHub panel](#) de control.
- b. En Your repositories, pase el cursor sobre el nombre del repositorio de destino. Aparece una descripción emergente en la que se muestra el nombre del GitHub usuario o la

organización, seguido de una barra diagonal (/) seguida del nombre del repositorio. Escriba este valor en Repository name (Nombre del repositorio).

 Note

Si el nombre del repositorio de destino no aparece en Tus repositorios, usa el GitHub cuadro de búsqueda para buscar el repositorio de destino y el nombre del GitHub usuario u organización.

8. En el cuadro ID de confirmación, introduce el ID de la confirmación asociada a la transferencia de la revisión de tu aplicación a GitHub.


Si no está seguro del valor que debe introducir:

- a. En otra pestaña del navegador web, ve a tu [GitHub panel de control](#).
 - b. En Your repositories (Sus repositorios), elija CodeDeployGitHubDemo.
 - c. En la lista de confirmaciones, busca y copia el ID de confirmación asociado al envío de la revisión de tu aplicación a GitHub. Este ID suele tener 40 caracteres y se compone de letras y números. (No utilice la versión corta del ID de confirmación, que suele ser los primeros 10 caracteres de la versión larga).
 - d. Pegue el ID de confirmación en el cuadro Commit ID.
9. Elija Deploy y continúe con el paso siguiente.

Para implementar la revisión (CLI)


Antes de poder llamar a cualquier AWS CLI comando con el que interactúes GitHub (por ejemplo, el create-deployment comando al que llamarás a continuación), debes dar CodeDeploy permiso para usar tu cuenta de GitHub usuario con GitHub la que interactuar en la CodeDeployGitHubDemo-App aplicación. Actualmente, debe usar la CodeDeploy consola para hacerlo.

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. Elija CodeDeployGitHubDemo-App.
4. En la pestaña Implementaciones, elija Crear implementación.


 Note

No va a crear una nueva implementación. Actualmente, esta es la única forma de dar CodeDeploy permiso para interactuar GitHub en nombre de tu cuenta de GitHub usuario.

5. En el grupo de implementación, seleccione CodeDeployGitHubDemo- DepGrp.
6. En Tipo de revisión, elija GitHub.
7. En Conectarse a GitHub, realice una de las siguientes acciones:
 - Para crear una conexión de CodeDeploy aplicaciones a una GitHub cuenta, cierre sesión GitHub en una pestaña independiente del navegador web. En GitHub cuenta, escriba un nombre para identificar esta conexión y, a continuación, seleccione Conectar a GitHub. La página web le pide que autorice CodeDeploy la interacción con GitHub la aplicación nombradaCodeDeployGitHubDemo-App. Continúe en el paso 8.
 - Para usar una conexión que ya haya creado, en la GitHubcuenta, seleccione su nombre y, a continuación, elija Conectar a GitHub. Continúe en el paso 10.
 - Para crear una conexión a otra GitHub cuenta, cierra sesión GitHub en otra pestaña del navegador web. Selecciona Conectarse a una GitHub cuenta diferente y, a continuación, selecciona Conectar a GitHub. Continúe en el paso 8.
8. Sigue las instrucciones de la página de inicio de sesión para iniciar sesión con tu nombre GitHub de usuario o correo electrónico y contraseña.
9. En la página Authorize application, elija Authorize application.
10. En la página CodeDeploy Crear despliegue, selecciona Cancelar.
11. Ejecuta el create-deployment comando para implementar la revisión desde tu GitHub repositorio a la instancia, donde:
 - *repositorio* es el nombre de tu GitHub cuenta, seguido de una barra diagonal (/), seguido del nombre de tu repositorio (CodeDeployGitHubDemo), por ejemplo, MyGitHubUserName/CodeDeployGitHubDemo

Si no está seguro del valor que debe indicar o si desea especificar un repositorio diferente:

1. [En otra pestaña del navegador web, ve a tu panel de control. GitHub](#)
2. En Your repositories, pase el cursor sobre el nombre del repositorio de destino. Aparece una descripción emergente en la que se muestra el nombre del GitHub usuario o la organización, seguido de una barra diagonal (/) seguida del nombre del repositorio. Este es el valor que debe usar.

 Note

Si el nombre del repositorio de destino no aparece en Tus repositorios, usa el GitHub cuadro de búsqueda para buscar el repositorio de destino y el nombre de GitHub usuario u organización correspondiente.

- *commit-id* es la confirmación asociada al almacenamiento de la versión de la revisión de la aplicación en el repositorio (por ejemplo, f835159a . . . 528eb76f).

Si no está seguro del valor que debe usar:

1. En una pestaña independiente del navegador web, ve a tu [GitHub panel](#) de control.
2. En Your repositories (Sus repositorios), elija CodeDeployGitHubDemo.
3. En la lista de confirmaciones, busca el ID de confirmación asociado al envío de la revisión de tu aplicación a GitHub. Este ID suele tener 40 caracteres y se compone de letras y números. (No utilice la versión corta del ID de confirmación, que suele ser los primeros 10 caracteres de la versión larga). Utilice este valor.

Si está trabajando en una máquina Linux, macOS o Unix local:

```
aws deploy create-deployment \  
  --application-name CodeDeployGitHubDemo-App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name CodeDeployGitHubDemo-DepGrip \  
  --description "My GitHub deployment demo" \  
  --github-location repository=repository,commitId=commit-id
```

Si se encuentra en un equipo local con Windows:

```
aws deploy create-deployment --application-name CodeDeployGitHubDemo-App --  
deployment-config-name CodeDeployDefault.OneAtATime --deployment-group-name
```

```
CodeDeployGitHubDemo-DepGrp --description "My GitHub deployment demo" --github-  
location repository=repository,commitId=commit-id
```

Paso 7: Monitorizar y comprobar la implementación

En este paso, utilizará la CodeDeploy consola o la AWS CLI para comprobar el éxito de la implementación. Utilizará el navegador web para ver la página implementada en la instancia que ha creado o configurado.

Note

Si se trata de una implementación en una instancia de Ubuntu Server, utilice su propia estrategia de pruebas para determinar si la revisión implementada funciona según lo previsto en la instancia y, a continuación, continúe en el paso siguiente.

Para monitorizar y comprobar la implementación (consola)

1. En el panel de navegación, expanda Implementar y después elija Implementaciones.
2. En la lista de despliegues, busque la fila con un valor de aplicación de CodeDeployGitHubDemo-App y un valor de grupo de despliegues de CodeDeployGitHubDemo -. DepGrp Si no aparece Succeeded o Failed en la columna Status, pulse el botón Refresh periódicamente.
3. Si aparece Failed en la columna Status, siga las instrucciones de [Visualización de los detalles de la instancia \(consola\)](#) para resolver los problemas de la implementación.
4. Si aparece Succeeded en la columna Status, ya puede comprobar la implementación en el navegador web. Nuestra revisión de ejemplo implementa una única página web en la instancia. Si la implementación es en una instancia de Amazon EC2, en el navegador web, vaya a `http://public-dns` para la instancia (por ejemplo, `http://ec2-01-234-567-890.compute-1.amazonaws.com`).
5. Si puede ver la página web, ¡enhorabuena! Ahora que ha utilizado correctamente AWS CodeDeploy la implementación de una revisión desde GitHub, puede pasar a [Paso 8: Eliminación](#)

Para monitorizar y comprobar la implementación (CLI)

1. Para obtener el ID de implementación, ejecute el comando `list-deployments` con la aplicación `CodeDeployGitHubDemo-App` y el grupo de implementaciones `CodeDeployGitHubDemo-DepGrp`:

```
aws deploy list-deployments --application-name CodeDeployGitHubDemo-App --
deployment-group-name CodeDeployGitHubDemo-DepGrp --query "deployments" --output
text
```

2. Ejecute el comando `get-deployment` indicando el ID de la implementación obtenido en la salida del comando `list-deployments`:

```
aws deploy get-deployment --deployment-id deployment-id --query "deploymentInfo.
[status, creator]" --output text
```

3. Si el resultado es `Failed`, siga las instrucciones de [Visualización de los detalles de la instancia \(consola\)](#) para resolver los problemas de la implementación.
4. Si el resultado es `Succeeded`, ya puede comprobar la implementación en el navegador web. Nuestra revisión de ejemplo implementa una única página web en la instancia. Si la implementación es en una instancia de Amazon EC2, puede ver la página en el navegador web en la dirección `http://public-dns` para la instancia de Amazon EC2 (por ejemplo, `http://ec2-01-234-567-890.compute-1.amazonaws.com`).
5. Si puede ver la página web, ¡enhorabuena! La has utilizado correctamente AWS CodeDeploy para realizar la implementación desde tu GitHub repositorio.

Paso 8: Eliminación

Para evitar cargos adicionales por los recursos que ha utilizado en este tutorial, debe terminar la instancia de Amazon EC2 y sus recursos asociados. Si lo desea, puede eliminar los registros de los componentes de CodeDeploy despliegue asociados a este tutorial. Si estaba utilizando un GitHub repositorio solo para este tutorial, también puede eliminarlo ahora.

Para eliminar una AWS CloudFormation pila (si utilizó la AWS CloudFormation plantilla para crear una instancia de Amazon EC2)

1. Inicie sesión en la AWS CloudFormation consola AWS Management Console y ábrala en <https://console.aws.amazon.com/cloudformation>.

2. En la columna Stacks (Pilas) elija la pila que comience por CodeDeploySampleStack.
3. Elija Eliminar.
4. Cuando se le solicite, seleccione Delete stack (Eliminar pila). Se eliminará la instancia de Amazon EC2 y el perfil de instancia de IAM y el rol de servicio asociados.

Para anular el registro y limpiar manualmente una instancia on-premises (si ha provisionado una instancia on-premises)

1. *Usa el AWS CLI comando [deregister](#) en la instancia local representada aquí por tu región `your-instance-name` la región asociada en tu región:*

```
aws deploy deregister --instance-name your-instance-name --no-delete-iam-user --  
region your-region
```

2. Desde la instancia en las instalaciones, llame al comando [uninstall](#):

```
aws deploy uninstall
```

Para terminar manualmente una instancia de Amazon EC2 (si ha iniciado manualmente una instancia de Amazon EC2)

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. En el panel de navegación, bajo Instances, elija Instances.
3. Seleccione la casilla junto a la instancia de Amazon EC2 que desea terminar. En el menú Actions, seleccione Instance State y elija Terminate.
4. Cuando se le pregunte, elija Yes, Terminate.

Para eliminar los registros de los componentes de implementación en CodeDeploy

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. Elija CodeDeployGitHubDemo-App.
4. Seleccione Eliminar aplicación.
5. Cuando se le pregunte, escriba **Delete** y, a continuación, elija Delete (Eliminar).

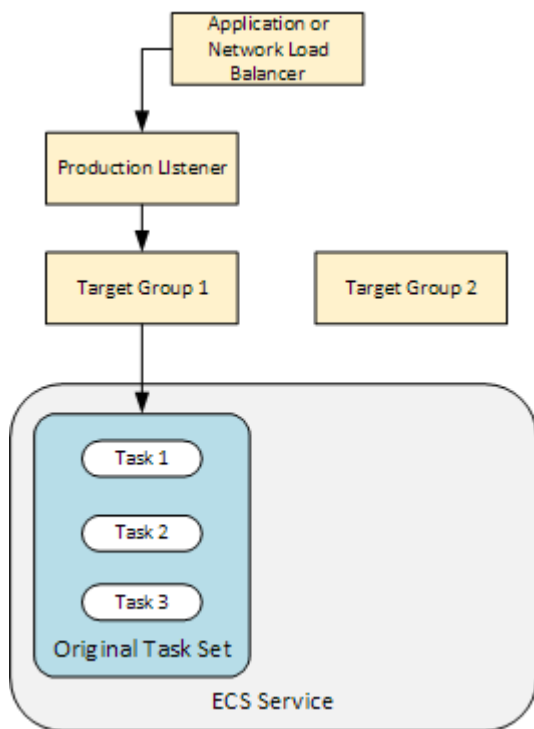
Para eliminar tu GitHub repositorio

Consulte [Eliminar un repositorio](#) en la [GitHub ayuda](#).

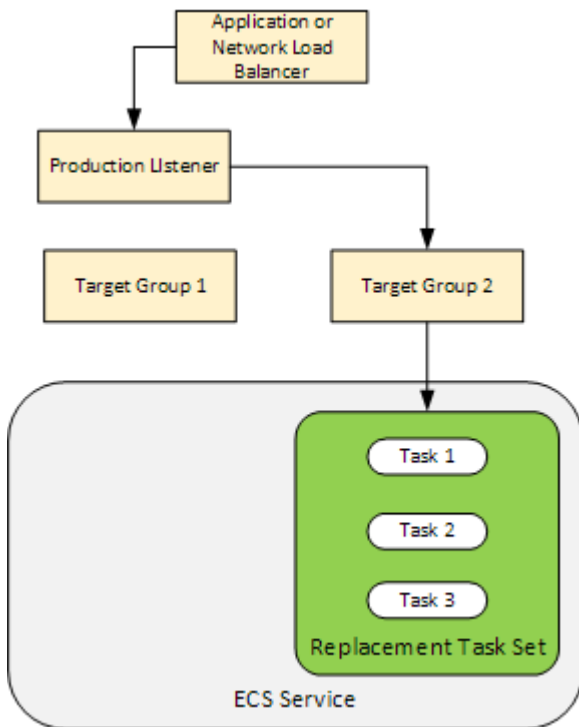
Tutorial: Implementación de una aplicación en Amazon ECS

En este tutorial, aprenderá a implementar una aplicación en Amazon ECS mediante CodeDeploy. Comience con una aplicación que ya creó e implementó en Amazon ECS. El primer paso consiste en actualizar la aplicación modificando su archivo de definición de tareas con una nueva etiqueta. A continuación, debe CodeDeploy implementar la actualización. Durante la implementación, CodeDeploy instala la actualización en un nuevo conjunto de tareas de reemplazo. A continuación, desvía el tráfico de producción de la versión original del servicio de Amazon ECS, que se encuentra en su conjunto de tareas original, a la versión actualizada en el conjunto de tareas de sustitución.

Durante una implementación de Amazon ECS, CodeDeploy utiliza un balanceador de carga que está configurado con dos grupos de destino y un detector de tráfico de producción. En el siguiente diagrama se muestra cómo están relacionados el equilibrador de carga, el oyente de producción, los grupos de destino y la aplicación de Amazon ECS antes de que comience la implementación. En este tutorial, se utiliza un equilibrador de carga de aplicación. También puede usar un equilibrador de carga de red.



Después de una implementación correcta, el oyente de tráfico de producción ofrece el tráfico al nuevo conjunto de tareas de sustitución y se termina el conjunto de tareas original. En el siguiente diagrama se muestra cómo están relacionados los recursos después de una implementación correcta. Para obtener más información, consulte [¿Qué sucede durante una implementación de Amazon ECS?](#).



Para obtener información sobre cómo utilizarla AWS CLI para implementar una aplicación en Amazon ECS, consulte el [Tutorial: Creación de un servicio mediante una implementación azul/verde](#). Para obtener información sobre cómo detectar e implementar automáticamente los cambios en un servicio de Amazon ECS CodeDeploy, consulte el [tutorial: Crear una canalización con una fuente de Amazon ECR y una implementación de ECS a implementación](#). CodePipeline CodeDeploy

Tras completar este tutorial, puede utilizar el grupo de CodeDeploy aplicaciones y despliegues que creó para añadir una prueba de validación de despliegues. [Tutorial: Implementación de un servicio de Amazon ECS con una prueba de validación](#)

Temas

- [Requisitos previos](#)
- [Paso 2: Actualizar la aplicación de Amazon ECS](#)
- [Paso 2: Crea el AppSpec archivo](#)
- [Paso 3: Utilice la CodeDeploy consola para implementar la aplicación](#)
- [Paso 4: Limpiar](#)

Requisitos previos

Para completar este tutorial, primero debe:

- Completar los pasos 2 y 3 de [Empezar con CodeDeploy](#).
- Crear un equilibrador de carga de aplicación configurado con dos grupos de destino y un oyente. Para obtener información sobre la creación un balanceador de carga mediante la consola, consulte [Configure un balanceador de carga, grupos objetivo y agentes de escucha para las implementaciones de CodeDeploy Amazon ECS](#). Para obtener información sobre cómo crear un balanceador de carga mediante el AWS CLI, consulte el [paso 1: Crear un balanceador de carga de aplicaciones](#) en la Guía del usuario de Amazon Elastic Container Service. Cuando cree el equilibrador de carga, anote lo siguiente para este tutorial:
 - Nombre del balanceador de carga.
 - Los nombres de los grupos de destino.
 - El puerto utilizado por el agente de escucha del balanceador de carga.
- Cree un clúster y servicio de Amazon ECS. Para obtener más información, consulte los pasos 2, 3 y 4 de [Tutorial: Creación de un servicio mediante una implementación azul/verde](#) en la Guía de usuario de Amazon Elastic Container Service. Anote lo siguiente para este tutorial:
 - El nombre de su clúster de Amazon ECS.
 - El ARN de la definición de tarea utilizada por el servicio de Amazon ECS.
 - El nombre del contenedor utilizado por el servicio de Amazon ECS.
- Cree un bucket de Amazon S3 para su AppSpec archivo.

Paso 2: Actualizar la aplicación de Amazon ECS

En esta sección, actualizará la aplicación de Amazon ECS con una nueva revisión de su definición de tarea. La revisión actualizada añade un nuevo par de claves y etiquetas. En [Paso 3: Utilice la CodeDeploy consola para implementar la aplicación](#), implemente la versión actualizada de la aplicación de Amazon ECS.

Para actualizar la definición de tarea, realice el siguiente procedimiento:

1. Abra la consola en <https://console.aws.amazon.com/ecs/v2>.
2. En el panel de navegación, elija Task Definitions.
3. Elija la definición de tarea utilizada por el servicio de Amazon ECS.
4. Seleccione la revisión de la definición de tarea y, a continuación, elija Crear nueva revisión y Crear nueva revisión.

5. En este tutorial, realice una pequeña actualización de la definición de la tarea añadiendo una etiqueta. En la parte inferior de la página, en Etiquetas, cree una nueva etiqueta escribiendo un nuevo par de clave y valor.
6. Seleccione Crear.

Debería ver que el número de revisión de la definición de la tarea se ha incrementado en uno.

7. Seleccione la pestaña JSON. Anote lo siguiente, ya que necesitará esa información en el siguiente paso.
 - El valor de `taskDefinitionArn`. Su formato es el siguiente: `arn:aws:ecs:aws-region:account-id:task-definition/task-definition-family:task-definition-revision`. Este es el ARN de la definición de tarea actualizada.
 - En el elemento `containerDefinitions`, el valor de `name`. Este es el nombre del contenedor.
 - En el elemento `portMappings`, el valor de `containerPort`. Este es el puerto del contenedor.

Paso 2: Crea el AppSpec archivo

En esta sección, crea el AppSpec archivo y lo carga en el bucket de Amazon S3 que creó en la [Requisitos previos](#) sección. El AppSpec archivo de una implementación de Amazon ECS especifica la definición de la tarea, el nombre del contenedor y el puerto del contenedor. Para obtener más información, consulte [AppSpec Ejemplo de archivo para una implementación de Amazon ECS](#) y [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#).

Para crear su AppSpec archivo

1. Si quieres crear tu AppSpec archivo con YAML, crea un archivo llamado `appspec.yaml`. Si quieres crear tu AppSpec archivo con JSON, crea un archivo con el nombre `appspec.json`.
2. Elige la pestaña adecuada, dependiendo de si usas YAML o JSON para tu AppSpec archivo, y copia su contenido en el AppSpec archivo que acabas de crear. Para la propiedad `TaskDefinition`, utilice el ARN de definición de tarea que anotó en la sección [Paso 2: Actualizar la aplicación de Amazon ECS](#).

JSON AppSpec

```
{
```

```

"version": 0.0,
"Resources": [
  {
    "TargetService": {
      "Type": "AWS::ECS::Service",
      "Properties": {
        "TaskDefinition": "arn:aws:ecs:aws-region-id:aws-account-id:task-
definition/ecs-demo-task-definition:revision-number",
        "LoadBalancerInfo": {
          "ContainerName": "your-container-name",
          "ContainerPort": your-container-port
        }
      }
    }
  }
]
}

```

YAML AppSpec

```

version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:aws-region-id:aws-account-id:task-
definition/ecs-demo-task-definition:revision-number"
        LoadBalancerInfo:
          ContainerName: "your-container-name"
          ContainerPort: your-container-port

```

Note

El conjunto de tareas de sustitución hereda la subred, el grupo de seguridad, la versión de la plataforma y los valores de IP públicas asignados del conjunto de tareas original. Puedes anular estos valores para tu conjunto de tareas de reemplazo configurando sus propiedades opcionales en el archivo. AppSpec Para obtener más información, consulte [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#) y [AppSpec Ejemplo de archivo para una implementación de Amazon ECS](#).

3. Cargue el AppSpec archivo en el depósito de S3 que creó como requisito previo para este tutorial.

Paso 3: Utilice la CodeDeploy consola para implementar la aplicación

En esta sección, creará un grupo de CodeDeploy aplicaciones y despliegues para implementar la aplicación actualizada en Amazon ECS. Durante la implementación, CodeDeploy traslada el tráfico de producción de la aplicación a su nueva versión en un nuevo conjunto de tareas de reemplazo.

Para completar este paso, necesita lo siguiente:

- El nombre de su clúster de Amazon ECS.
- El nombre de servicio de Amazon ECS.
- El nombre del equilibrador de carga de aplicación
- El puerto del agente de escucha de producción.
- Los nombres de grupo de destino.
- El nombre del bucket de S3 que creó.

Para crear una CodeDeploy aplicación

1. Inicie sesión AWS Management Console y abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy/>.
2. Elija Crear aplicación.
3. En Application name (Nombre de aplicación), escriba **ecs-demo-codedeploy-app**.
4. En Compute platform (Plataforma de computación), elija Amazon ECS.
5. Elija Crear aplicación.

Para crear un grupo CodeDeploy de implementación

1. En la pestaña Grupos de implementación de la página de la aplicación, elija Crear grupo de implementación.
2. En Nombre de grupo de implementación, escriba **ecs-demo-dg**.
3. En Función de servicio, elija una función de servicio que conceda CodeDeploy acceso a Amazon ECS. Para obtener más información, consulte [Administración de identidades y accesos para AWS CodeDeploy](#).

4. En Configuración de entorno, elija el nombre del clúster y el nombre del servicio de Amazon ECS.
5. En Equilibradores de carga, seleccione el nombre del equilibrador de carga que envía tráfico a su servicio de Amazon ECS.
6. En Puerto del oyente de producción, elija el puerto y el protocolo del oyente que envía tráfico de producción a su servicio de Amazon ECS (por ejemplo, HTTP: 80). Este tutorial no incluye un oyente de prueba opcional, por lo que no elija un puerto en Puerto de oyente de prueba.
7. En Target group 1 name (Nombre de grupo de destino 1) y Target group 2 name (Nombre de grupo de destino 2), elija dos grupos de destino diferentes para dirigir el tráfico durante su implementación. Asegúrese de que se trata de los grupos de destino que ha creado para el equilibrador de carga. No importa el que se utilice para el grupo de destino 1 y el que se utilice para el grupo de destino 2.
8. Elija Reroute traffic immediately (Redirigir el tráfico de forma inmediata).
9. En Original revision termination (Terminación de revisión original), elija 0 days (0 días), 0 hours (0 horas) y 5 minutes (5 minutos). Esto le permite ver que la implementación se completa más rápido que si utiliza el valor predeterminado (1 hora).

Environment configuration

Choose an ECS cluster name

ecs-tutorial-cluster

Choose an ECS service name

ecs-demo-service

Load balancers

Choose a load balancer

ecs-demo-alb

Production listener port

HTTP: 80

Test listener port - *optional*

A test listener is required if you want to test your replacement version before traffic reroutes to it

Target group 1 name

ecs-demo-tg-1

Target group 2 name

ecs-demo-tg-2

Deployment settings

Traffic rerouting

Choose whether traffic reroutes to the replacement environment immediately or waits for you to start the rerouting process

Reroute traffic immediately

Specify when to reroute traffic

Deployment Configuration

CodeDeployDefault.ECSALLAtOnce

Original revision termination

Specify how long CodeDeploy waits before it terminates the original task set. After termination starts, you cannot rollback manually or automatically

Days

0

Hours

0

Minutes

5


10. Elija Crear grupo de implementación.


Para implementar la aplicación de Amazon ECS


1. En la página de la consola del grupo de implementaciones, elija Create deployment (Crear implementación).
2. En Grupo de implementación, elija ecs-demo-dg.
3. En Tipo de revisión, elija Mi aplicación está almacenada en Amazon S3. En Revision location (Ubicación de la revisión), escriba el nombre de su bucket de S3.
4. En Revision file type (Tipo de archivo de revisión), elija .json o .yaml, según corresponda.
5. En el cuadro Deployment description (Descripción de la implementación), especifique una descripción para la implementación (opcional).
6. Elija Crear implementación.
7. En Deployment status (Estado de implementación), puede monitorizar la implementación. Después de que el 100 % del tráfico de producción se dirija al conjunto de tareas de sustitución y antes de que transcurra el tiempo de espera de cinco minutos, puede elegir Terminar conjunto de tareas original para terminar inmediatamente el conjunto de tareas original. Si no elige Terminate original task set (Terminar conjunto de tareas original), el conjunto de tareas original termina después de que transcurra el tiempo de espera de cinco minutos especificado.

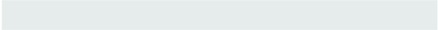
d-MVGEP9PSM Stop deployment Stop and roll back deployment Terminate original task set

Deployment status



Step 1:
Deploying replacement task set **Completed**
 ✔ Succeeded

Step 2:
Retrouting production traffic to replacement task set **100% traffic shifted**
 ✔ Succeeded

Step 3:
Wait 5 minutes 0 seconds **Waiting**
 ↻ In progress

Step 4:
Terminate original task set **Not started**
 ↻ In progress

Traffic shifting progress

Original	Replacement
 Original task set not serving traffic	 Replacement task set serving traffic

Paso 4: Limpiar

El siguiente tutorial se [Tutorial: Implementación de un servicio de Amazon ECS con una prueba de validación](#) basa en este tutorial y utiliza el grupo de CodeDeploy aplicaciones y despliegues que creó. Si desea seguir los pasos de ese tutorial, omita este paso y no elimine los recursos que ha creado.

Note

Su AWS cuenta no incurre en cargos por los CodeDeploy recursos que creó.

Los nombres de los recursos en estos pasos son los nombres sugeridos en este tutorial (por ejemplo, **ecs-demo-codedeploy-app** para el nombre de la CodeDeploy aplicación). Si utilizó nombres diferentes, asegúrese de usarlos en el borrado.

1. Use el [delete-deployment-group](#) comando para eliminar el grupo CodeDeploy de implementación.

```
aws deploy delete-deployment-group --application-name ecs-demo-codedeploy-app --  
deployment-group-name ecs-demo-dg --region aws-region-id
```

2. Utilice el comando [delete-application](#) para eliminar la CodeDeploy aplicación.

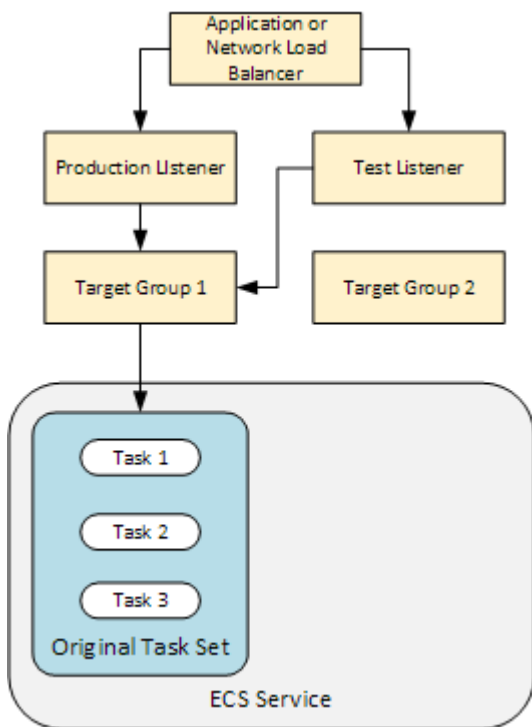
```
aws deploy delete-application --application-name ecs-demo-codedeploy-app --  
region aws-region-id
```

Tutorial: Implementación de un servicio de Amazon ECS con una prueba de validación

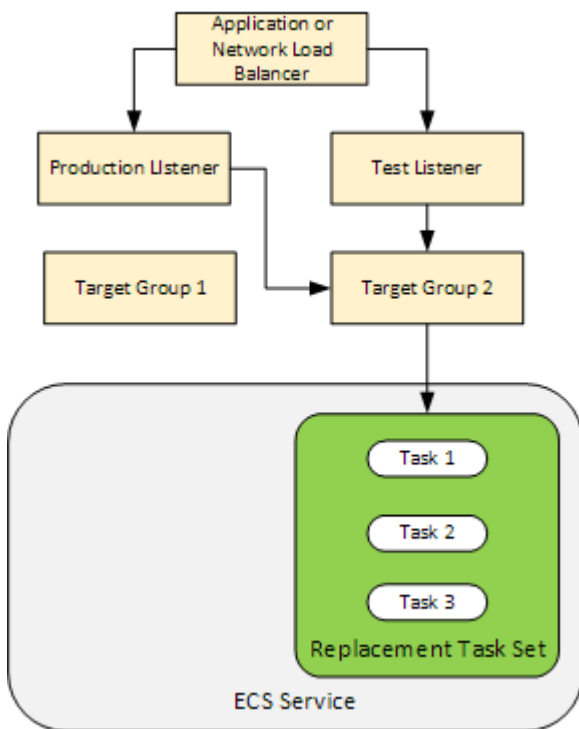
En este tutorial, aprenderá a utilizar una función de Lambda para validar parte de la implementación de una aplicación de Amazon ECS actualizada. En este tutorial se utilizan la CodeDeploy aplicación, el grupo de CodeDeploy implementación y la aplicación Amazon ECS en la que utilizó [Tutorial: Implementación de una aplicación en Amazon ECS](#). Complete ese tutorial antes de comenzar este.

Para añadir una prueba de validación, primero debe implementar la prueba en una función de Lambda. A continuación, en el AppSpec archivo de despliegue, especifique la función Lambda para el enlace del ciclo de vida que desee probar. Si se produce un error en una prueba de validación, la implementación se detiene, se revisa y se marca como errónea. Si la prueba se realiza correctamente, la implementación continúa con el siguiente evento o enlace del ciclo de vida de la implementación.

Durante una implementación de Amazon ECS con pruebas de validación, CodeDeploy utiliza un balanceador de carga que está configurado con dos grupos objetivo: un detector de tráfico de producción y un detector de tráfico de prueba. En el siguiente diagrama se muestra cómo están relacionados el equilibrador de carga, los oyentes de producción y prueba, los grupos de destino y la aplicación de Amazon ECS antes de que comience la implementación. En este tutorial, se utiliza un equilibrador de carga de aplicación. También puede usar un equilibrador de carga de red.



Durante una implementación de Amazon ECS, existen cinco enlaces de ciclo de vida para pruebas. En este tutorial se implementa una prueba durante el tercer enlace de implementación de ciclo de vida, `AfterAllowTestTraffic`. Para obtener más información, consulte [Lista de enlaces de eventos de ciclo de vida para una implementación de Amazon ECS](#). Después de una implementación correcta, el agente de escucha de tráfico de producción ofrece el tráfico al nuevo conjunto de tareas de sustitución y se termina el conjunto de tareas original. En el siguiente diagrama se muestra cómo están relacionados los recursos después de una implementación correcta. Para obtener más información, consulte [¿Qué sucede durante una implementación de Amazon ECS?](#).



i Note

Si completa este tutorial, es posible que se le cobren cargos a su cuenta. AWS Estos incluyen posibles cargos por CodeDeploy, AWS Lambda, y CloudWatch. Para obtener más información, consulta [AWS CodeDeploy precios](#), [AWS Lambda precios](#) y [CloudWatch precios de Amazon](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Creación de un oyente de prueba](#)
- [Paso 2: Actualizar la aplicación de Amazon ECS](#)
- [Paso 3: Creación de una función de Lambda de enlace de ciclo de vida](#)
- [Paso 4: actualice AppSpec el archivo](#)
- [Paso 5: Utilice la CodeDeploy consola para implementar el servicio Amazon ECS](#)
- [Paso 6: Ver el resultado de la función de enlace Lambda en los registros CloudWatch](#)
- [Paso 7: limpiar](#)

Requisitos previos

Para completar correctamente este tutorial, primero debe realizar el siguiente procedimiento:

- Cumpla los requisitos previos de [Requisitos previos](#) para [Tutorial: Implementación de una aplicación en Amazon ECS](#).
- Realice los pasos que se indican en [Tutorial: Implementación de una aplicación en Amazon ECS](#). Anote lo siguiente:
 - Nombre del balanceador de carga.
 - Los nombres de los grupos de destino.
 - El puerto utilizado por el agente de escucha del balanceador de carga.
 - El ARN del balanceador de carga. Puede utilizarlo para crear un nuevo agente de escucha.
 - El ARN de uno de los grupos de destino. Puede utilizarlo para crear un nuevo agente de escucha.
 - El grupo de CodeDeploy aplicaciones y despliegues que cree.
 - El AppSpec archivo que crea y que utiliza la CodeDeploy implementación. Puede editar este archivo en este tutorial.

Paso 1: Creación de un oyente de prueba

Una implementación de Amazon ECS con pruebas de validación requiere un segundo oyente. Este oyente se utiliza para ofrecer tráfico de prueba a la aplicación de Amazon ECS actualizada en un conjunto de tareas de sustitución. Las pruebas de validación se ejecutan en el tráfico de prueba.

El agente de escucha del tráfico de prueba puede utilizar cualquiera de los grupos de destino. Utilice el AWS CLI comando [create-listener](#) para crear un segundo detector con una regla predeterminada que reenvíe el tráfico de prueba al puerto 8080. Utilice el ARN del balanceador de carga y el ARN de uno de los grupos de destino.

```
aws elbv2 create-listener --load-balancer-arn your-load-balancer-arn \  
--protocol HTTP --port 8080 \  
--default-actions Type=forward,TargetGroupArn=your-target-group-arn --region your-aws-  
region
```

Paso 2: Actualizar la aplicación de Amazon ECS

En esta sección, actualizará la aplicación de Amazon ECS para que utilice una nueva revisión de su definición de tarea. Puede crear la nueva revisión y añadirle una actualización secundaria añadiendo una etiqueta.

Para actualizar la definición de tarea, realice el siguiente procedimiento:

1. Abra la consola clásica de Amazon ECS en <https://console.aws.amazon.com/ecs/>.
2. En el panel de navegación, elija Task Definitions.
3. Seleccione la casilla de verificación de la definición de tarea utilizada por el servicio de Amazon ECS.
4. Elija Create new revision (Crear nueva revisión).
5. Realice una pequeña actualización de la definición de la tarea añadiendo una etiqueta. En la parte inferior de la página, en Tags (Etiquetas), cree una nueva etiqueta escribiendo un nuevo par de clave y valor.
6. Seleccione Crear. Debería ver que el número de revisión de la definición de la tarea se ha incrementado en uno.
7. Seleccione la pestaña JSON. Anote el valor de taskDefinitionArn. Su formato es el siguiente: `arn:aws:ecs:aws-region:account-id:task-definition/task-definition-family:task-definition-revision`. Este es el ARN de la definición de tarea actualizada.

Paso 3: Creación de una función de Lambda de enlace de ciclo de vida

En esta sección, implementará una función de Lambda para el enlace de `AfterAllowTestTraffic` de la implementación de Amazon ECS. La función de Lambda ejecuta una prueba de validación antes de instalar la aplicación de Amazon ECS actualizada. En este tutorial, la función de Lambda devuelve `Succeeded`. Durante una implementación real, las pruebas de validación devuelven `Succeeded` o `Failed`, en función del resultado de la prueba de validación. También durante una implementación real, puede implementar una función de Lambda de prueba para uno o varios de los demás enlaces de eventos del ciclo de vida de la implementación de Amazon ECS (`BeforeInstall`, `AfterInstall`, `BeforeAllowTraffic`, y `AfterAllowTraffic`). Para obtener más información, consulte [Lista de enlaces de eventos de ciclo de vida para una implementación de Amazon ECS](#).

Se requiere un rol de IAM para crear la función de Lambda. El rol otorga a la función Lambda permiso para escribir en los CloudWatch registros y establecer el estado de un enlace de CodeDeploy ciclo de vida.

Cómo crear un rol de IAM

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, seleccione Roles (Roles) y, a continuación, seleccione Create role (Crear rol).
3. Cree un rol con las propiedades siguientes:
 - Trusted entity (Entidad de confianza): AWS Lambda.
 - Permisos: AWSLambdaBasicExecutionRole. Esto le otorga a la función Lambda permiso para escribir en los registros. CloudWatch
 - Role name (Nombre de rol): **lambda-cli-hook-role**.

Para obtener más información, consulte [Crear un rol AWS Lambda de ejecución](#).

4. Asocie el permiso `codedeploy:PutLifecycleEventHookExecutionStatus` al rol que ha creado. Esto otorga a sus funciones de Lambda permiso para establecer el estado de un enlace de CodeDeploy ciclo de vida durante una implementación. Para obtener más información, consulte [Añadir permisos de identidad de IAM](#) en la Guía del AWS Identity and Access Management usuario y [PutLifecycleEventHookExecutionStatus](#) en la Referencia de la CodeDeploy API.

Para crear una función de Lambda del enlace de **AfterAllowTestTraffic**

1. Cree un archivo denominado `AfterAllowTestTraffic.js` con el siguiente contenido.

```
'use strict';

const AWS = require('aws-sdk');
const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {

  console.log("Entering AfterAllowTestTraffic hook.");

  // Read the DeploymentId and LifecycleEventHookExecutionId from the event payload
```



```
var deploymentId = event.DeploymentId;
var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;
var validationTestResult = "Failed";

// Perform AfterAllowTestTraffic validation tests here. Set the test result
// to "Succeeded" for this tutorial.
console.log("This is where AfterAllowTestTraffic validation tests happen.")
validationTestResult = "Succeeded";

// Complete the AfterAllowTestTraffic hook by sending CodeDeploy the validation
status
var params = {
  deploymentId: deploymentId,
  lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
  status: validationTestResult // status can be 'Succeeded' or 'Failed'
};

// Pass CodeDeploy the prepared validation test results.
codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
  if (err) {
    // Validation failed.
    console.log('AfterAllowTestTraffic validation tests failed');
    console.log(err, err.stack);
    callback("CodeDeploy Status update failed");
  } else {
    // Validation succeeded.
    console.log("AfterAllowTestTraffic validation tests succeeded");
    callback(null, "AfterAllowTestTraffic validation tests succeeded");
  }
});
}
```

2. Cree un paquete de implementación de Lambda.

```
zip AfterAllowTestTraffic.zip AfterAllowTestTraffic.js
```

3. Utilice el comando `create-function` para crear una función de Lambda para el enlace de `AfterAllowTestTraffic`.

```
aws lambda create-function --function-name AfterAllowTestTraffic \
  --zip-file fileb://AfterAllowTestTraffic.zip \
  --handler AfterAllowTestTraffic.handler \
```

```
--runtime nodejs10.x \  
--role arn:aws:iam::aws-account-id:role/lambda-cli-hook-role
```

4. Anote el ARN de la función de Lambda en la respuesta de `create-function`. Este ARN se utiliza cuando se actualiza el AppSpec archivo de la CodeDeploy implementación en el siguiente paso.

Paso 4: actualice AppSpec el archivo

En esta sección, actualizas tu AppSpec archivo con una Hooks sección. En la sección Hooks, debe especificar una función de Lambda para el enlace de ciclo de vida `AfterAllowTestTraffic`.

Para actualizar el AppSpec archivo

1. Abra el AppSpec archivo [Paso 2: Crea el AppSpec archivo](#) de archivo que creó en [Tutorial: Implementación de una aplicación en Amazon ECS](#).
2. Actualice la propiedad `TaskDefinition` con el ARN de definición de tarea que anotó en [Paso 2: Actualizar la aplicación de Amazon ECS](#).
3. Copia y pega la Hooks sección en tu AppSpec archivo de archivo. Actualice el ARN después de `AfterAllowTestTraffic` con el ARN de la función de Lambda que anotó en [Paso 3: Creación de una función de Lambda de enlace de ciclo de vida](#).

JSON AppSpec

```
{  
  "version": 0.0,  
  "Resources": [  
    {  
      "TargetService": {  
        "Type": "AWS::ECS::Service",  
        "Properties": {  
          "TaskDefinition": "arn:aws:ecs:aws-region-id:aws-account-id::task-definition/ecs-demo-task-definition:revision-number",  
          "LoadBalancerInfo": {  
            "ContainerName": "sample-website",  
            "ContainerPort": 80  
          }  
        }  
      }  
    ]  
  }
```

```

    }
  ],
  "Hooks": [
    {
      "AfterAllowTestTraffic": "arn:aws:lambda:aws-region-id:aws-account-id:function:AfterAllowTestTraffic"
    }
  ]
}

```

YAML AppSpec

```

version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:aws-region-id:aws-account-id::task-definition/ecs-demo-task-definition:revision-number"
        LoadBalancerInfo:
          ContainerName: "sample-website"
          ContainerPort: 80
Hooks:
  - AfterAllowTestTraffic: "arn:aws:lambda:aws-region-id:aws-account-id:function:AfterAllowTestTraffic"

```

4. Guarde el AppSpec archivo y cárguelo en su bucket de S3.

Paso 5: Utilice la CodeDeploy consola para implementar el servicio Amazon ECS

En esta sección, debe actualizar el grupo de implementaciones especificando el puerto para el agente de escucha de prueba. Este es el agente de escucha que creó en [Paso 1: Creación de un oyente de prueba](#). Durante la implementación, CodeDeploy ejecuta la prueba de validación durante el ciclo de vida de la `AfterAllowTestTraffic` implementación, utilizando el tráfico de prueba enviado a tu conjunto de tareas de reemplazo mediante el detector de pruebas. La prueba de validación devuelve el resultado `Succeeded`, por lo que la implementación avanza con el siguiente evento del ciclo de vida de la implementación. En un escenario real, la función de prueba devuelve `Succeeded` o `Failed`.

Para añadir un agente de escucha de prueba a su grupo de implementaciones, realice el siguiente procedimiento:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy/>.
2. En el panel de navegación, elija Applications (Aplicaciones).
3. Elija la aplicación que ha creado en [Tutorial: Implementación de una aplicación en Amazon ECS](#). Si usaste el nombre sugerido, es ecs-demo-codedeploy-app.
4. En Deployment groups (Grupos de implementaciones), elija el grupo de implementaciones que creó en [Tutorial: Implementación de una aplicación en Amazon ECS](#). Si usó el nombre sugerido, lo es ecs-demo-dg.
5. Elija Editar.
6. En Test listener port (Puerto de agente de escucha de prueba), elija el puerto y el protocolo del agente de escucha de prueba que creó anteriormente en este tutorial. Debe ser HTTP: 8080.
7. Elija Guardar cambios.

Para implementar la aplicación de Amazon ECS

1. En la página de la consola del grupo de implementaciones, elija Create deployment (Crear implementación).
2. En Grupo de implementación, elija ecs-demo-dg.
3. En Tipo de revisión, elija Mi aplicación está almacenada en Amazon S3. En la ubicación de la revisión, introduzca el nombre del depósito y el AppSpec archivo de S3 (por ejemplo, **s3://my-s3-bucket/appspec.json**).
4. En Revision file type (Tipo de archivo de revisión), elija .json o .yaml, según corresponda.
5. En el cuadro Deployment description (Descripción de la implementación), especifique una descripción para la implementación (opcional).
6. Elija Crear implementación.

Puede monitorizar la implementación en Deployment status (Estado de implementación). Después de que el 100 % del tráfico de producción se dirija al conjunto de tareas de sustitución, puede elegir Terminar conjunto de tareas original para terminar inmediatamente el conjunto de tareas original. Si no elige Terminate original task set (Terminar conjunto de tareas original), el conjunto de tareas original termina después de la duración especificada al crear el grupo de implementaciones.

The screenshot displays the AWS CodeDeploy console interface. At the top, there are three buttons: "Stop deployment", "Stop and roll back deployment", and "Terminate original task set". Below these, the "Deployment status" section shows five steps:

- Step 1:** Deploying replacement task set. Status: Completed. Progress bar is green with a checkmark and "Succeeded".
- Step 2:** Test traffic route setup. Status: Completed. Progress bar is green with a checkmark and "Succeeded".
- Step 3:** Rerouting production traffic to replacement task set. Status: 100% traffic shifted. Progress bar is green with a checkmark and "Succeeded".
- Step 4:** Wait 5 minutes 0 seconds. Status: Waiting. Progress bar is blue with a right arrow and "In progress".
- Step 5:** Terminate original task set. Status: Not started. Progress bar is grey with a right arrow and "In progress".

The "Traffic shifting progress" section shows two progress indicators:

- Original:** 0% progress. Status: Original task set not serving traffic.
- Replacement:** 100% progress. Status: Replacement task set serving traffic.

Paso 6: Ver el resultado de la función de enlace Lambda en los registros CloudWatch

Si la CodeDeploy implementación se realiza correctamente, las pruebas de validación de las funciones de enlace de Lambda también lo son. Puede confirmarlo consultando el registro para ver la función de enlace en CloudWatch los registros.

1. Abre la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, elija Logs (Registros). Debería ver un grupo de registros nuevo para la función de enlace de Lambda que especificó en el archivo AppSpec .

Log Groups	Insights	Expire Events After	Metric Filters	Subscriptions
<input type="radio"/> /aws/lambda/AfterAllowTestTraffic	Explore	Never Expire	0 filters	None

- Elija el nuevo grupo de registros. Debe ser /aws/lambda/ AfterAllowTestTrafficHook.
- Elija el flujo de registros. Si ve más de un flujo de registros, elija el que tenga la fecha y la hora más recientes en Last Event Time (Hora del último evento).
- Expanda los eventos del flujo de registro para confirmar que la función de enlace de Lambda escribió mensajes de éxito en el registro. A continuación se muestra cómo la función de enlace de Lambda se ha realizado correctamente.

Time (UTC +00:00)	Message
2019-09-11	
	<i>No older ev</i>
20:11:20	START RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916 Version: \$LATEST
	START RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916 Version: \$LATEST
20:11:21	2019-09-11T20:11:21.033Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO Entering AfterAllowTestTraffic hook.
	2019-09-11T20:11:21.033Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO Entering AfterAllowTestTraffic hook.
20:11:21	2019-09-11T20:11:21.034Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO This is where AfterAllowTestTraffic validation tests happen.
	2019-09-11T20:11:21.034Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO This is where AfterAllowTestTraffic validation tests happen.
20:11:21	2019-09-11T20:11:21.789Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO AfterAllowTestTraffic validation tests succeeded
	2019-09-11T20:11:21.789Z e875485b-cdb2-4e1e-b4e8-8054e7b7f916 INFO AfterAllowTestTraffic validation tests succeeded
20:11:21	END RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916
20:11:21	REPORT RequestId: e875485b-cdb2-4e1e-b4e8-8054e7b7f916 Duration: 977.80 ms Billed Duration: 1000 ms Memory Size: 128 MB Ma

Paso 7: limpiar

Cuando termine este tutorial, debe limpiar los recursos asociados para evitar incurrir en cargos por recursos que no está utilizando. Los nombres de los recursos de este paso son los nombres sugeridos en este tutorial (por ejemplo, **ecs-demo-codedeploy-app** para el nombre de la CodeDeploy aplicación). Si utilizó nombres diferentes, asegúrese de usarlos en el borrado.

Para borrar los recursos del tutorial, realice el siguiente procedimiento:

- Use el [delete-deployment-group](#) comando para eliminar el grupo CodeDeploy de implementación.

```
aws deploy delete-deployment-group --application-name ecs-demo-deployment-group --
deployment-group-name ecs-demo-dg --region aws-region-id
```

- Utilice el comando [delete-application](#) para eliminar la CodeDeploy aplicación.

```
aws deploy delete-application --application-name ecs-demo-deployment-group --  
region aws-region-id
```

- Utilice el comando [delete-function](#) para eliminar la función de enlace de Lambda.

```
aws lambda delete-function --function-name AfterAllowTestTraffic
```

- Utilice el [delete-log-group](#) comando para eliminar el grupo de CloudWatch registros.

```
aws logs delete-log-group --log-group-name /aws/lambda/AfterAllowTestTraffic
```

Tutorial: Implemente una función Lambda actualizada con CodeDeploy y el modelo de aplicaciones AWS sin servidor

AWS SAM es un marco de código abierto para crear aplicaciones sin servidor. Transforma y expande la sintaxis YAML de una AWS SAM plantilla en AWS CloudFormation sintaxis para crear aplicaciones sin servidor, como una función Lambda. Para obtener más información, consulte [¿Qué es AWS Serverless Application Model?](#).

En este tutorial, usará AWS SAM para crear una solución que haga lo siguiente:

- Crea una función de Lambda.
- Crea el grupo CodeDeploy de aplicaciones y despliegues.
- Crea dos funciones Lambda que ejecutan pruebas de validación de despliegue durante los enlaces CodeDeploy del ciclo de vida.
- Detecta cuándo se actualiza la función de Lambda. La actualización de la función Lambda desencadena un despliegue CodeDeploy que desplaza gradualmente el tráfico de producción de la versión original de la función Lambda a la versión actualizada.

Note

Este tutorial requiere que cree recursos que podrían dar lugar a cargos en su cuenta de AWS . Estos incluyen posibles cargos para CodeDeploy Amazon CloudWatch y AWS Lambda. Para obtener más información, consulta [CodeDeploy precios](#), [CloudWatch precios de Amazon](#) y [AWS Lambda precios](#).

Temas

- [Requisitos previos](#)
- [Paso 1: Configurar la infraestructura](#)
- [Paso 2: Actualizar la función de Lambda](#)
- [Paso 3: Implementar la función de Lambda actualizada](#)
- [Paso 4: Visualizar los resultados de la implementación](#)
- [Paso 5: Eliminar](#)

Requisitos previos

Para completar este tutorial, primero debe:

- Realice los pasos que se indican en [Empezar con CodeDeploy](#).
- Instale la AWS Serverless Application Model CLI. Para obtener más información, consulte [Instalación de la CLI de AWS SAM](#).
- Cree un bucket de S3. AWS SAM carga los artefactos a los que se hace referencia en la [plantilla de AWS SAM](#) en este depósito.

Paso 1: Configurar la infraestructura

En este tema se muestra cómo crear AWS SAM archivos para la AWS SAM plantilla y las funciones de Lambda. A continuación, utilice los `deploy` comandos AWS SAM `package` y para generar los componentes de su infraestructura. Cuando la infraestructura esté lista, dispondrá de un grupo de CodeDeploy aplicaciones y despliegues, la función Lambda para actualizar e implementar y dos funciones Lambda que contienen pruebas de validación que se ejecutan al implementar la función Lambda. Cuando haya terminado, puede utilizarlos AWS CloudFormation para ver los componentes en la consola de Lambda o AWS CLI para probar la función Lambda.

Temas

- [Creación de archivos](#)
- [Package la aplicación AWS SAM](#)
- [Implemente la aplicación AWS SAM](#)
- [Inspección y prueba de la infraestructura \(opcional\)](#)

Creación de archivos

Para crear su infraestructura, debe crear los siguientes archivos:

- `template.yml`
- `myDateTimeFunction.js`
- `beforeAllowTraffic.js`
- `afterAllowTraffic.js`

Temas

- [Cree su plantilla SAM AWS](#)
- [Creación de un archivo para la función de Lambda](#)
- [Cree un archivo para su función BeforeAllowTraffic Lambda](#)
- [Cree un archivo para su función AfterAllowTraffic Lambda](#)

Cree su plantilla SAM AWS

Cree un archivo de plantilla AWS SAM que especifique los componentes de su infraestructura.

Para crear su plantilla AWS SAM

1. Cree un directorio llamado `SAM-Tutorial`.
2. En el directorio `SAM-Tutorial`, cree un archivo denominado `template.yml`.
3. Copie el código YAML siguiente en `template.yml`. Esta es la plantilla de AWS SAM .

```
AWSTemplateFormatVersion : '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: A sample SAM template for deploying Lambda functions.

Resources:
# Details about the myDateTimeFunction Lambda function
myDateTimeFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: myDateTimeFunction.handler
    Runtime: nodejs18.x
# Instructs your myDateTimeFunction is published to an alias named "live".
  AutoPublishAlias: live
```

```
# Grants this function permission to call lambda:InvokeFunction
Policies:
  - Version: "2012-10-17"
    Statement:
      - Effect: "Allow"
        Action:
          - "lambda:InvokeFunction"
        Resource: '*'
    DeploymentPreference:
# Specifies the deployment configuration
  Type: Linear10PercentEvery1Minute
# Specifies Lambda functions for deployment lifecycle hooks
  Hooks:
    PreTraffic: !Ref beforeAllowTraffic
    PostTraffic: !Ref afterAllowTraffic

# Specifies the BeforeAllowTraffic lifecycle hook Lambda function
beforeAllowTraffic:
  Type: AWS::Serverless::Function
  Properties:
    Handler: beforeAllowTraffic.handler
    Policies:
      - Version: "2012-10-17"
# Grants this function permission to call
  codedeploy:PutLifecycleEventHookExecutionStatus
  Statement:
    - Effect: "Allow"
      Action:
        - "codedeploy:PutLifecycleEventHookExecutionStatus"
      Resource:
        !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
      - Version: "2012-10-17"
# Grants this function permission to call lambda:InvokeFunction
  Statement:
    - Effect: "Allow"
      Action:
        - "lambda:InvokeFunction"
      Resource: !Ref myDateTimeFunction.Version
  Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
  FunctionName: 'CodeDeployHook_beforeAllowTraffic'
  DeploymentPreference:
    Enabled: false
```

```
    Timeout: 5
    Environment:
      Variables:
        NewVersion: !Ref myDateTimeFunction.Version

# Specifies the AfterAllowTraffic lifecycle hook Lambda function
afterAllowTraffic:
  Type: AWS::Serverless::Function
  Properties:
    Handler: afterAllowTraffic.handler
    Policies:
      - Version: "2012-10-17"
        Statement:
# Grants this function permission to call
      codedeploy:PutLifecycleEventHookExecutionStatus
        - Effect: "Allow"
          Action:
            - "codedeploy:PutLifecycleEventHookExecutionStatus"
          Resource:
            !Sub 'arn:aws:codedeploy:${AWS::Region}:
${AWS::AccountId}:deploymentgroup:${ServerlessDeploymentApplication}/*'
        - Version: "2012-10-17"
          Statement:
# Grants this function permission to call lambda:InvokeFunction
        - Effect: "Allow"
          Action:
            - "lambda:InvokeFunction"
          Resource: !Ref myDateTimeFunction.Version
    Runtime: nodejs18.x
# Specifies the name of the Lambda hook function
    FunctionName: 'CodeDeployHook_afterAllowTraffic'
    DeploymentPreference:
      Enabled: false
    Timeout: 5
    Environment:
      Variables:
        NewVersion: !Ref myDateTimeFunction.Version
```

Esta plantilla especifica lo siguiente. Para obtener más información, consulte [Conceptos de la plantilla de AWS SAM](#).

Una función de Lambda llamada `myDateTimeFunction`

Cuando se publica esta función de Lambda, la línea `AutoPublishAlias` de la plantilla la vincula a un alias denominado `live`. Más adelante en este tutorial, una actualización de esta función desencadena una implementación AWS CodeDeploy que desplaza gradualmente el tráfico de producción de la versión original a la versión actualizada.

Dos funciones de validación de implementación de Lambda

Las siguientes funciones de Lambda se ejecutan durante los enlaces CodeDeploy del ciclo de vida. Las funciones contienen código que valida la implementación de `myDateTimeFunction` actualizado. El resultado de las pruebas de validación se transfiere CodeDeploy mediante su método de `PutLifecycleEventHookExecutionStatus` API. Si la prueba de validación falla, se produce un error en la implementación y se revisa.

- `CodeDeployHook_beforeAllowTraffic` se ejecuta durante el enlace de `BeforeAllowTraffic`.
- `CodeDeployHook_afterAllowTraffic` se ejecuta durante el enlace de `AfterAllowTraffic`.

El nombre de ambas funciones comienza por `CodeDeployHook_`. El rol de `CodeDeployRoleForLambda` permite llamadas al método de `invoke` de Lambda solo en funciones de Lambda con nombres que comienzan con este prefijo. Para obtener más información, consulte [AppSpec sección de «ganchos» para una implementación de AWS Lambda](#) y [PutLifecycleEventHookExecutionStatus](#) en la Referencia de la CodeDeploy API.

Detección automática de una función de Lambda actualizada

El término `AutoPublishAlias` indica al marco de trabajo que detecte cuándo cambia la función de `myDateTimeFunction` y, a continuación, la implementa con el alias `live`.

Una configuración de implementación

La configuración de despliegue determina la velocidad a la que la CodeDeploy aplicación desplaza el tráfico de la versión original de la función Lambda a la nueva versión. Esta plantilla especifica la configuración de implementación predefinida `Linear10PercentEvery1Minute`.

Note

No puede especificar una configuración de despliegue personalizada en una plantilla de AWS SAM. Para obtener más información, consulte [Create a Deployment Configuration](#).

Funciones de enlace de ciclo de vida de implementación

La sección Hooks especifica las funciones que se van a ejecutar durante los enlaces de eventos del ciclo de vida. `PreTraffic` especifica la función que se ejecuta durante el enlace `BeforeAllowTraffic`. `PostTraffic` especifica la función que se ejecuta durante el enlace `AfterAllowTraffic`.

Permisos para que Lambda invoque otra función de Lambda

El `lambda:InvokeFunction` permiso especificado otorga permiso al rol utilizado por la aplicación AWS SAM para invocar una función Lambda. Esto es necesario cuando las funciones `CodeDeployHook_beforeAllowTraffic` y `CodeDeployHook_afterAllowTraffic` invocan la función de Lambda implementada durante las pruebas de validación.

Creación de un archivo para la función de Lambda

Cree el archivo para la función que actualizará e implementará más adelante en este tutorial.

Note

Una función de Lambda puede utilizar cualquier tiempo de ejecución admitido por AWS Lambda. Para obtener más información, consulte [Tiempos de ejecución de AWS Lambda](#).

Para crear una función de Lambda

1. Cree un archivo de texto y guárdelo como `myDateTimeFunction.js` en el directorio `SAM-Tutorial`.
2. Copie el siguiente código Node.js en `myDateTimeFunction.js`.

```
'use strict';

exports.handler = function(event, context, callback) {

  if (event.body) {
    event = JSON.parse(event.body);
  }

  var sc; // Status code
  var result = ""; // Response payload
```

```
switch(event.option) {
  case "date":
    switch(event.period) {
      case "yesterday":
        result = setDateResult("yesterday");
        sc = 200;
        break;
      case "today":
        result = setDateResult();
        sc = 200;
        break;
      case "tomorrow":
        result = setDateResult("tomorrow");
        sc = 200;
        break;
      default:
        result = {
          "error": "Must specify 'yesterday', 'today', or 'tomorrow'."
        };
        sc = 400;
        break;
    }
  break;

/* Later in this tutorial, you update this function by uncommenting
this section. The framework created by AWS SAM detects the update
and triggers a deployment by CodeDeploy. The deployment shifts
production traffic to the updated version of this function.

  case "time":
    var d = new Date();
    var h = d.getHours();
    var mi = d.getMinutes();
    var s = d.getSeconds();

    result = {
      "hour": h,
      "minute": mi,
      "second": s
    };
    sc = 200;
    break;

*/
```

```
        default:
            result = {
                "error": "Must specify 'date' or 'time'."
            };
            sc = 400;
            break;
    }

    const response = {
        statusCode: sc,
        headers: { "Content-type": "application/json" },
        body: JSON.stringify( result )
    };

    callback(null, response);

    function setDateResult(option) {

        var d = new Date(); // Today
        var mo; // Month
        var da; // Day
        var y; // Year

        switch(option) {
            case "yesterday":
                d.setDate(d.getDate() - 1);
                break;
            case "tomorrow":
                d.setDate(d.getDate() + 1);
            default:
                break;
        }

        mo = d.getMonth() + 1; // Months are zero offset (0-11)
        da = d.getDate();
        y = d.getFullYear();

        result = {
            "month": mo,
            "day": da,
            "year": y
        };

        return result;
    }
}
```

```
}  
};
```

La función de Lambda devuelve el día, el mes y el año de ayer, hoy o mañana. Más adelante en este tutorial, quite el comentario del código que actualiza la función para devolver información sobre el día o la hora que especifique (por ejemplo, el día, el mes y el año, o la hora, minuto y segundo actuales). El marco creado por AWS SAM detecta e implementa la versión actualizada de la función.

Note

Esta función Lambda también se utiliza en un AWS Cloud9 tutorial. AWS Cloud9 es un entorno de desarrollo integrado basado en la nube. Para obtener información sobre cómo crear, ejecutar, actualizar y depurar esta función en AWS Cloud9, consulte el [AWS Lambda tutorial de AWS Cloud9](#).

Cree un archivo para su función BeforeAllowTraffic Lambda

Cree el archivo para la función de Lambda del enlace `beforeAllowTraffic`.

1. Cree un archivo de texto y guárdelo como `beforeAllowTraffic.js` en el directorio `SAM-Tutorial`.
2. Copie el siguiente código Node.js en `beforeAllowTraffic.js`. Esta función se ejecuta durante el enlace `BeforeAllowTraffic` de la implementación.

```
'use strict';  
  
const AWS = require('aws-sdk');  
const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});  
var lambda = new AWS.Lambda();  
  
exports.handler = (event, context, callback) => {  
  
    console.log("Entering PreTraffic Hook!");  
  
    // Read the DeploymentId and LifecycleEventHookExecutionId from the event  
    payload  
    var deploymentId = event.DeploymentId;  
    var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;
```



```
var functionToTest = process.env.NewVersion;
console.log("BeforeAllowTraffic hook tests started");
console.log("Testing new function version: " + functionToTest);

// Create parameters to pass to the updated Lambda function that
// include the newly added "time" option. If the function did not
// update, then the "time" option is invalid and function returns
// a statusCode of 400 indicating it failed.
var lambdaParams = {
  FunctionName: functionToTest,
  Payload: "{\"option\": \"time\"}",
  InvocationType: "RequestResponse"
};

var lambdaResult = "Failed";
// Invoke the updated Lambda function.
lambda.invoke(lambdaParams, function(err, data) {
  if (err){ // an error occurred
    console.log(err, err.stack);
    lambdaResult = "Failed";
  }
  else{ // successful response
    var result = JSON.parse(data.Payload);
    console.log("Result: " + JSON.stringify(result));
    console.log("statusCode: " + result.statusCode);

    // Check if the status code returned by the updated
    // function is 400. If it is, then it failed. If
    // is not, then it succeeded.
    if (result.statusCode !== "400"){
      console.log("Validation succeeded");
      lambdaResult = "Succeeded";
    }
    else {
      console.log("Validation failed");
    }
  }

  // Complete the PreTraffic Hook by sending CodeDeploy the validation status
  var params = {
    deploymentId: deploymentId,
    lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
    status: lambdaResult // status can be 'Succeeded' or 'Failed'
  };
};
```

```
// Pass CodeDeploy the prepared validation test results.
codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data)
{
  if (err) {
    // Validation failed.
    console.log("CodeDeploy Status update failed");
    console.log(err, err.stack);
    callback("CodeDeploy Status update failed");
  } else {
    // Validation succeeded.
    console.log("CodeDeploy status updated successfully");
    callback(null, "CodeDeploy status updated successfully");
  }
});
}
});
}
```

Cree un archivo para su función AfterAllowTraffic Lambda

Cree el archivo para la función de Lambda del enlace afterAllowTraffic.

1. Cree un archivo de texto y guárdelo como `afterAllowTraffic.js` en el directorio SAM-Tutorial.
2. Copie el siguiente código Node.js en `afterAllowTraffic.js`. Esta función se ejecuta durante el enlace AfterAllowTraffic de la implementación.

```
'use strict';

const AWS = require('aws-sdk');
const codedeploy = new AWS.CodeDeploy({apiVersion: '2014-10-06'});
var lambda = new AWS.Lambda();

exports.handler = (event, context, callback) => {

  console.log("Entering PostTraffic Hook!");

  // Read the DeploymentId and LifecycleEventHookExecutionId from the event
  payload
  var deploymentId = event.DeploymentId;
  var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;
```

```
var functionToTest = process.env.NewVersion;
console.log("AfterAllowTraffic hook tests started");
console.log("Testing new function version: " + functionToTest);

// Create parameters to pass to the updated Lambda function that
// include the original "date" parameter. If the function did not
// update as expected, then the "date" option might be invalid. If
// the parameter is invalid, the function returns
// a statusCode of 400 indicating it failed.
var lambdaParams = {
  FunctionName: functionToTest,
  Payload: "{\"option\": \"date\", \"period\": \"today\"}",
  InvocationType: "RequestResponse"
};

var lambdaResult = "Failed";
// Invoke the updated Lambda function.
lambda.invoke(lambdaParams, function(err, data) {
  if (err){ // an error occurred
    console.log(err, err.stack);
    lambdaResult = "Failed";
  }
  else{ // successful response
    var result = JSON.parse(data.Payload);
    console.log("Result: " + JSON.stringify(result));
    console.log("statusCode: " + result.statusCode);

    // Check if the status code returned by the updated
    // function is 400. If it is, then it failed. If
    // is not, then it succeeded.
    if (result.statusCode != "400"){
      console.log("Validation of time parameter succeeded");
      lambdaResult = "Succeeded";
    }
    else {
      console.log("Validation failed");
    }
  }

  // Complete the PostTraffic Hook by sending CodeDeploy the validation status
  var params = {
    deploymentId: deploymentId,
    lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
    status: lambdaResult // status can be 'Succeeded' or 'Failed'
  };
};
```

```
// Pass CodeDeploy the prepared validation test results.
codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data)
{
  if (err) {
    // Validation failed.
    console.log("CodeDeploy Status update failed");
    console.log(err, err.stack);
    callback("CodeDeploy Status update failed");
  } else {
    // Validation succeeded.
    console.log("CodeDeploy status updated successfully");
    callback(null, "CodeDeploy status updated successfully");
  }
});
}
});
}
```

Package la aplicación AWS SAM

Ahora debería tener cuatro archivos en su directorio SAM-Tutorial:

- beforeAllowTraffic.js
- afterAllowTraffic.js
- myDateTimeFunction.js
- template.yml

Ahora puede usar el `sam package` comando AWS SAM para crear y empaquetar artefactos para sus funciones y CodeDeploy aplicaciones de Lambda. Los artefactos se cargan en un bucket de S3. El resultado del comando es un nuevo archivo llamado `package.yml`. El `sam deploy` comando AWS SAM utilizará este archivo en el siguiente paso.

Note

Para obtener más información sobre el comando `sam package`, consulte la [Referencia de la CLI de AWS SAM](#) en la Guía para desarrolladores de AWS Serverless Application Model .

En el directorio `SAM-Tutorial`, ejecute el siguiente elemento.

```
sam package \  
  --template-file template.yml \  
  --output-template-file package.yml \  
  --s3-bucket your-S3-bucket
```

Para el parámetro `s3-bucket`, especifique el bucket de Amazon S3 que ha creado como requisito previo para este tutorial. `output-template-file` Especifica el nombre del nuevo archivo que utiliza el `sam deploy` comando AWS SAM.

Implemente la aplicación AWS SAM

Utilice el `sam deploy` comando AWS SAM con el `package.yml` archivo para crear sus funciones Lambda y su grupo de CodeDeploy aplicaciones y despliegues mediante AWS CloudFormation

Note

Para obtener más información sobre el comando `sam deploy`, consulte la [Referencia de la CLI de AWS SAM](#) en la Guía para desarrolladores de AWS Serverless Application Model .

En el directorio `SAM-Tutorial`, ejecute el siguiente comando.

```
sam deploy \  
  --template-file package.yml \  
  --stack-name my-date-time-app \  
  --capabilities CAPABILITY_IAM
```

El parámetro `--capabilities CAPABILITY_IAM` es necesario para autorizar a AWS CloudFormation a crear roles de IAM.


Inspección y prueba de la infraestructura (opcional)

En este tema se muestra cómo ver los componentes de la infraestructura y probar la función de Lambda.

Para ver el resultado de la pila después de ejecutar **sam deploy**, realice el siguiente procedimiento:

1. Abra la AWS CloudFormation consola en <https://console.aws.amazon.com/cloudformation>.

2. En el panel de navegación, seleccione Stacks (Pilas). La pila `my-date-time-app` aparece en la parte superior.
3. Elija la pestaña Events (Eventos) para ver qué eventos se han completado. Puede ver los eventos mientras la creación de la pila está en curso. Cuando se haya completado la creación de la pila, podrá ver todos los eventos de creación esta.
4. Con la pila seleccionada, elija Resources (Recursos). En la columna Tipo, puede ver las funciones de Lambda `myDateTimeFunction`, `CodeDeployHook_beforeAllowTraffic` y `CodeDeployHook_afterAllowTraffic`. La columna ID físico de cada una de las funciones de Lambda contiene un enlace para ver las funciones en la consola de Lambda.

 Note

El nombre de la función `myDateTimeFunction` Lambda va precedido del nombre de la AWS CloudFormation pila y se le ha añadido un identificador, por lo que parece `my-date-time-app-myDateTimeFunction-123456ABCDEF`

5. [Abra la CodeDeploy consola en https://console.aws.amazon.com/codedeploy/](https://console.aws.amazon.com/codedeploy/).
6. En el panel de navegación, expanda Deploy (Implementar) y después elija Applications (Aplicaciones).
7. Debería ver una nueva CodeDeploy aplicación creada por AWS CloudFormation con un nombre que empiece por `my-date-time-app-ServerlessDeploymentApplication`. Elija esta aplicación.
8. Debería ver un grupo de implementaciones con un nombre que comienza por `my-date-time-app-myDateTimeFunctionDeploymentGroup`. Elija este grupo de implementaciones.

En Configuración de despliegue, deberías ver `CodeDeployDefault.LambdaLinear10PercentEvery 1 minuto`.

Para probar la función (consola) (opcional)

1. Abra la AWS Lambda consola en <https://console.aws.amazon.com/lambda/>.
2. En el panel de navegación, elija la función `my-date-time-app-myDateTimeFunction`. En la consola, su nombre contiene un identificador, por lo que tendrá un aspecto similar a `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Seleccione Probar.
4. En Event name (Nombre del evento), escriba un nombre para el evento de prueba.

5. Escriba lo siguiente para el evento de prueba y, a continuación, elija Create (Crear).

```
{
  "option": "date",
  "period": "today"
}
```

6. Seleccione Probar. Solo debería ver el evento de prueba en la lista de eventos de prueba.

En Execution result (Resultado de ejecución), debería ver succeeded (correcto).

7. En Execution result (Resultado de ejecución), expanda Details (Detalles) para ver los resultados. Debería ver el mes, día y año actuales.

Para probar la función (AWS CLI) (opcional):

1. Localice el ARN de la función de Lambda. Aparece en la parte superior de la consola de Lambda cuando se visualiza la función.
2. Ejecute el siguiente comando de la `.your-function-arn` Sustitúyala por la función ARN.

```
aws lambda invoke \
--function your-function-arn \
--cli-binary-format raw-in-base64-out \
--payload "{\"option\": \"date\", \"period\": \"today\"}" out.txt
```

3. Abra `out.txt` para confirmar que el resultado contiene el mes, el día y el año actuales.

Paso 2: Actualizar la función de Lambda

En este tema, actualizará el archivo `myDateTimeFunction.js`. En el siguiente paso, utilizará el archivo para implementar la función actualizada. Esto desencadena CodeDeploy su implementación al cambiar el tráfico de producción de la versión actual de la función Lambda a la versión actualizada.

Para actualizar la función de Lambda

1. Abra `myDateTimeFunction.js`.
2. Elimine los dos marcadores de comentario (`"/**` y `*/`) y el texto explicativo al principio y al final del case denominado `time` en el bloque `switch`.

El código sin comentarios le permite transferir un nuevo parámetro, `time`, a la función. Si pasa `time` a la función actualizada, devuelve el valor actual `hour`, `minute` y `second`.

3. Guarde `myDateTimeFunction.js`. Debe parecerse a lo siguiente:

```
'use strict';

exports.handler = function(event, context, callback) {

  if (event.body) {
    event = JSON.parse(event.body);
  }

  var sc; // Status code
  var result = ""; // Response payload

  switch(event.option) {
    case "date":
      switch(event.period) {
        case "yesterday":
          result = setDateResult("yesterday");
          sc = 200;
          break;
        case "today":
          result = setDateResult();
          sc = 200;
          break;
        case "tomorrow":
          result = setDateResult("tomorrow");
          sc = 200;
          break;
        default:
          result = {
            "error": "Must specify 'yesterday', 'today', or 'tomorrow'."
          };
          sc = 400;
          break;
      }
      break;
    case "time":
      var d = new Date();
      var h = d.getHours();
      var mi = d.getMinutes();
```



```
    var s = d.getSeconds();

    result = {
      "hour": h,
      "minute": mi,
      "second": s
    };
    sc = 200;
    break;

  default:
    result = {
      "error": "Must specify 'date' or 'time'."
    };
    sc = 400;
    break;
}

const response = {
  statusCode: sc,
  headers: { "Content-type": "application/json" },
  body: JSON.stringify( result )
};

callback(null, response);

function setDateResult(option) {

  var d = new Date(); // Today
  var mo; // Month
  var da; // Day
  var y; // Year

  switch(option) {
    case "yesterday":
      d.setDate(d.getDate() - 1);
      break;
    case "tomorrow":
      d.setDate(d.getDate() + 1);
    default:
      break;
  }

  mo = d.getMonth() + 1; // Months are zero offset (0-11)
```

```
    da = d.getDate();
    y = d.getFullYear();

    result = {
      "month": mo,
      "day": da,
      "year": y
    };

    return result;
  }
};
```

Paso 3: Implementar la función de Lambda actualizada

En este paso, utilizará el `myDateTimeFunction.js` actualizado para actualizar e iniciar la implementación de la función de Lambda. Puede supervisar el progreso de la implementación en la AWS Lambda consola CodeDeploy o.

La `AutoPublishAlias`: `live` línea de la AWS SAM plantilla hace que la infraestructura detecte las actualizaciones de las funciones que utilizan el `live` alias. Una actualización de la función desencadena una implementación CodeDeploy que desplaza el tráfico de producción de la versión original de la función a la versión actualizada.

Los comandos `sam package` y `sam deploy` se utilizan para actualizar y activar la implementación de la función de Lambda. Ha ejecutado estos comandos en [Package la aplicación AWS SAM](#) y [Implemente la aplicación AWS SAM](#).

Para implementar la función de Lambda actualizada

1. En el directorio `SAM-Tutorial`, ejecute el siguiente comando.

```
sam package \
  --template-file template.yml \
  --output-template-file package.yml \
  --s3-bucket your-S3-bucket
```

Esto crea un nuevo conjunto de artefactos que hacen referencia a la función de Lambda actualizada en el bucket de S3.

2. En el directorio `SAM-Tutorial`, ejecute el siguiente comando.

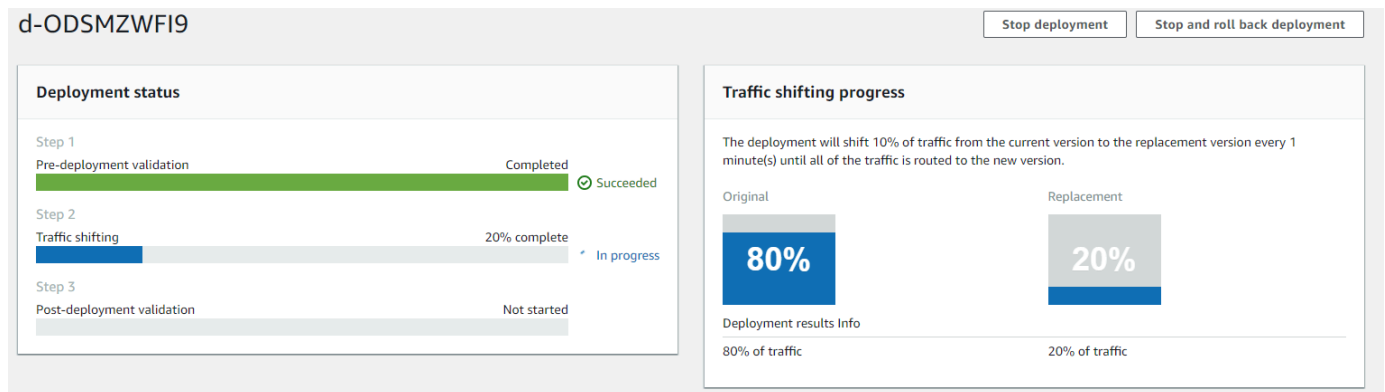
```
sam deploy \  
  --template-file package.yml \  
  --stack-name my-date-time-app \  
  --capabilities CAPABILITY_IAM
```

Como el nombre de la pila sigue siendo `my-date-time-app`, AWS CloudFormation reconoce que se trata de una actualización de la pila. Para ver la pila actualizada, regresa a la AWS CloudFormation consola y, en el panel de navegación, selecciona Pilas.

(Opcional) para ver el tráfico durante una implementación (CodeDeploy consola)

1. Abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy/>.
2. En el panel de navegación, expanda Aplicaciones y, a continuación, elija su ServerlessDeploymentApplication aplicación `my-date-time-app`.
3. En Deployment groups (Grupos de implementaciones), elija el grupo de implementaciones de la aplicación. Su estado debe ser In progress (En curso).
4. En Deployment group history (Historial de grupos de implementaciones), elija la implementación que está en curso.

La barra de progreso Traffic shifting (Desvío de tráfico) y los porcentajes de los cuadros Original y Replacement (Sustitución) de esta página muestran su progreso.



Para ver el tráfico durante una implementación (consola de Lambda) (opcional)

1. Abra la AWS Lambda consola en <https://console.aws.amazon.com/lambda/>.

2. En el panel de navegación, elija la función `my-date-time-app-myDateTimeFunction`. En la consola, su nombre contiene un identificador, por lo que tendrá un aspecto similar a `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Elija Alias y, a continuación, directo.

Las ponderaciones situadas junto a la versión original de la función (versión 1) y la versión actualizada de la función (versión 2) muestran la cantidad de tráfico que se ofrece a cada versión en el momento en que se cargó esta página de la consola de AWS Lambda . La página no actualiza las ponderaciones con el paso del tiempo. Si actualiza la página una vez al minuto, la ponderación de la versión 1 disminuye en un 10 por ciento y la ponderación de la versión 2 aumenta en un 10 por ciento hasta que la ponderación de la versión 2 es 100.

Aliases

You are viewing the configuration for alias `live`.
[Manage the configuration](#) for the underlying version 1.
[Manage the configuration](#) for the underlying version 2.

Name
live

Description

Version*
 ▼ Weight: 80%

You can shift traffic between two versions, based on weights (%) that you assign. Click [here](#) to learn more.

Additional version
 ▼ Weight
 %

Paso 4: Visualizar los resultados de la implementación

En este paso, podrá ver los resultados de la implementación. Si la implementación se realiza correctamente, puede confirmar que la función de Lambda actualizada recibe tráfico de producción. Si su implementación falla, puede usar CloudWatch los registros para ver el resultado de las pruebas de validación en la función Lambda que se ejecutan durante los ciclos de vida de la implementación.

Temas

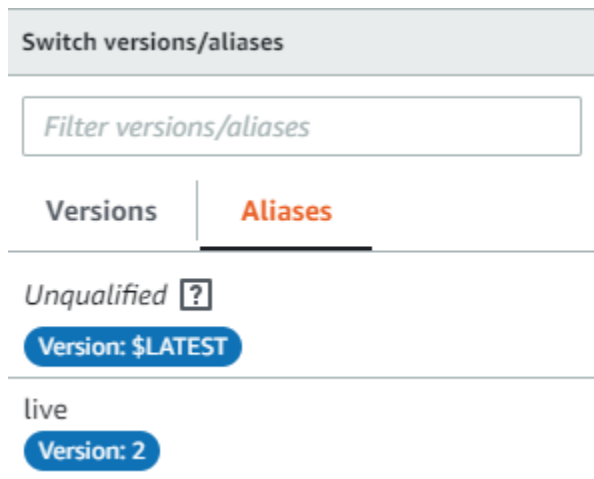
- [Prueba de la función implementada](#)
- [Vea los eventos de los enganches en los registros CloudWatch](#)

Prueba de la función implementada

El comando `aws deploy` actualiza la función de Lambda `my-date-time-app-myDateTimeFunction`. La versión de la función se actualiza a 2 y se añade al alias `live`.

Para ver la actualización en la consola de Lambda

1. [Abra la AWS Lambda consola en https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/).
2. En el panel de navegación, elija la función `my-date-time-app-myDateTimeFunction`. En la consola, su nombre contiene un identificador, por lo que tendrá un aspecto similar a `my-date-time-app-myDateTimeFunction-123456ABCDEF`.
3. Elija **Qualifiers (Cualificadores)** y, a continuación, **Aliases (Alias)**. Una vez completada la implementación (aproximadamente 10 minutos), para el alias `live` alias, debería ver **Version: 2 (Versión: 2)**.



4. En **Function code (Código de función)**, consulte el código fuente de la función. Deben aparecer los cambios.
5. Puede utilizar las instrucciones de prueba de [Paso 2: Actualizar la función de Lambda](#) para probar la función actualizada (opcional). Cree un nuevo evento de prueba con la siguiente carga y, a continuación, confirme que el resultado contiene la hora, el minuto y el segundo actuales.

```
{
  "option": "time"
}
```

Para usar la AWS CLI función actualizada, ejecute el siguiente comando y, a continuación, ábralo `out.txt` para confirmar que el resultado contiene la hora, el minuto y el segundo actuales.

```
aws lambda invoke --function your-function-arn --payload "{\"option\": \"time\"}"
out.txt
```

Note

Si lo utiliza AWS CLI para probar la función antes de que se complete la implementación, es posible que reciba resultados inesperados. Esto se debe a que transfiere CodeDeploy gradualmente el 10 por ciento del tráfico a la versión actualizada cada minuto. Durante la implementación, parte del tráfico sigue apuntando a la versión original, por lo que `aws lambda invoke` podría utilizar la versión original. Después de 10 minutos, la implementación se ha completado y todo el tráfico apunta a la nueva versión de la función.

Vea los eventos de los enganches en los registros CloudWatch

Durante el `BeforeAllowTraffic` enlace, CodeDeploy ejecuta la función `CodeDeployHook_beforeAllowTraffic Lambda`. Durante el `AfterAllowTraffic` enlace, CodeDeploy ejecuta la función `CodeDeployHook_afterAllowTraffic Lambda`. Cada función ejecuta una prueba de validación que invoca la versión actualizada de la función mediante el nuevo parámetro `time`. Si la actualización de la función de Lambda se realiza correctamente, la opción `time` no provoca un error y la validación se realiza correctamente. Si la función no se ha actualizado, el parámetro no reconocido provoca un error y la validación falla. Estas pruebas de validación son solo para fines de demostración. Puede escribir sus propias pruebas para validar la implementación. Puede utilizar la consola de CloudWatch registros para ver las pruebas de validación.

Para ver los eventos de tus CodeDeploy ganchos

1. Abre la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, elija Logs (Registros).
3. En la lista de grupos de registros, elija `/aws/lambda/ _` o `CodeDeployHook /aws/lambda/ _`.
`beforeAllowTraffic CodeDeployHook afterAllowTraffic`
4. Elija el flujo de registros. Debe ver solo uno.
5. Expanda los eventos para ver sus detalles.

Time (UTC +00:00)	Message
2019-07-12	No older events found at the moment. Re
▶ 22:08:56	START RequestId: 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Version: \$LATEST
▶ 22:08:56	2019-07-12T22:08:56.834Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Entering PreTraffic Hook!
▶ 22:08:56	2019-07-12T22:08:56.834Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Testing new function version: arn:aws:lambda:ca-
▼ 22:08:58	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Result: {"statusCode":200,"headers":{"Content-t
2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Result:	<pre>{ "statusCode": 200, "headers": { "Content-type": "application/json" }, "body": "{\"hour\":22,\"minute\":8,\"second\":57}" }</pre>
▼ 22:08:58	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 statusCode: 200
2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 statusCode: 200	
▼ 22:08:58	2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Validation succeeded
2019-07-12T22:08:58.084Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Validation succeeded	
▼ 22:08:58	2019-07-12T22:08:58.302Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Codedeploy status updated successfully
2019-07-12T22:08:58.302Z 9f1d8158-5acc-4618-bf5f-5c1c99d8fa49 Codedeploy status updated successfully	

Paso 5: Eliminar

Para evitar cargos adicionales por los recursos que utilizó durante este tutorial, elimine los recursos creados por la AWS SAM plantilla y los CloudWatch registros creados por las funciones de validación de Lambda.

Para eliminar tu pila AWS CloudFormation

1. Inicia sesión en la AWS CloudFormation consola AWS Management Console y ábrela en <https://console.aws.amazon.com/cloudformation>.
2. En la columna Stacks (Pilas), elija la pila de my-date-time-app y, a continuación, elija Delete (Eliminar).
3. Cuando se le solicite, seleccione Delete stack (Eliminar pila). Se eliminan las funciones Lambda, el grupo de CodeDeploy aplicaciones y despliegues y las funciones de IAM creadas por AWS SAM .

Para eliminar sus registros en Logs CloudWatch

1. Abre la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.

2. En el panel de navegación, elija Logs (Registros).
3. En la lista de grupos de registros, seleccione el botón situado junto a CodeDeployHook/aws/lambda/ `_`. beforeAllowTraffic
4. En Actions (Acciones), elija Delete log group (Eliminar grupo de registros), y, a continuación, Yes, Delete (Sí, eliminar).
5. En la lista de grupos de registros, seleccione el botón situado junto a /aws/lambda/ `_`. CodeDeployHook afterAllowTraffic
6. En Actions (Acciones), elija Delete log group (Eliminar grupo de registros), y, a continuación, Yes, Delete (Sí, eliminar).

Trabajando con el CodeDeploy agente

El AWS CodeDeploy agente es un paquete de software que, cuando se instala y configura en una instancia, permite que esa instancia se utilice en CodeDeploy las implementaciones.

AWS es compatible con la última versión secundaria del CodeDeploy agente. Actualmente, la última versión secundaria es la 1.7.x.

Note

El CodeDeploy agente solo es necesario si se implementa en una plataforma informática local o de EC2. El agente no es necesario para las implementaciones que utilizan la plataforma de informática de Amazon ECS o AWS Lambda.

Un archivo de configuración se coloca en la instancia cuando se instala el agente. Este archivo se utiliza para especificar cómo funciona el agente. Este archivo de configuración especifica las rutas de los directorios y otros ajustes AWS CodeDeploy que se utilizarán cuando interactúe con la instancia. Puede cambiar algunas de las opciones de configuración en el archivo. Para obtener información sobre cómo trabajar con el archivo de configuración del CodeDeploy agente, consulte [CodeDeploy referencia de configuración del agente](#).

Para obtener más información sobre cómo trabajar con el CodeDeploy agente, como los pasos para instalar, actualizar y verificar las versiones, consulte [Gestión de las operaciones CodeDeploy de los agentes](#).

Temas

- [Sistemas operativos compatibles con el agente CodeDeploy](#)
- [Protocolo de comunicación y puerto para el CodeDeploy agente](#)
- [Historial de versiones del agente CodeDeploy](#)
- [Administrar el CodeDeploy proceso](#)
- [Limpieza de revisiones de aplicación y archivos de registro](#)
- [Archivos instalados por el agente CodeDeploy](#)
- [Gestión de las operaciones CodeDeploy de los agentes](#)

Sistemas operativos compatibles con el agente CodeDeploy

Sistemas operativos de AMI de Amazon EC2 compatibles

El CodeDeploy agente se ha probado en los siguientes sistemas operativos AMI de Amazon EC2:

- Amazon Linux 2023 (ARM, x86)
- Amazon Linux 2 (ARM, x86)
- Microsoft Windows Server 2022, 2019
- Red Hat Enterprise Linux (RHEL) 9.x, 8.x, 7.x
- Ubuntu Server 22.04 LTS, 20.04 LTS, 18.04 LTS, 16.04 LTS

El CodeDeploy agente está disponible como código abierto para que pueda adaptarlo a sus necesidades. Se puede utilizar con otros sistemas operativos de AMI de Amazon EC2. Para obtener más información, vaya al repositorio de [CodeDeploy agentes](#) en GitHub.

Sistemas operativos en las instalaciones compatibles

El CodeDeploy agente se ha probado en los siguientes sistemas operativos locales:

- Microsoft Windows Server 2022, 2019
- Red Hat Enterprise Linux (RHEL) 9.x, 8.x, 7.x
- Servidor Ubuntu 22.04 LTS, 20.04 LTS

El CodeDeploy agente está disponible como código abierto para que lo adapte a sus necesidades. Se puede utilizar con otros sistemas operativos de instancias on-premises. Para obtener más información, vaya al repositorio de [CodeDeploy agentes](#) en GitHub.

Protocolo de comunicación y puerto para el CodeDeploy agente

El CodeDeploy agente se comunica de forma saliente mediante HTTPS a través del puerto 443.

Cuando el CodeDeploy agente se ejecute en una instancia de EC2, utilizará el extremo de [metadatos de EC2](#) para recuperar la información relacionada con la instancia. Obtenga más información sobre [la limitación y la concesión de acceso al servicio de metadatos de instancia](#).

Historial de versiones del agente CodeDeploy

Sus instancias deben ejecutar una versión compatible del CodeDeploy agente. La versión mínima admitida actualmente es la 1.7.x.

Note

Recomendamos utilizar la última versión del CodeDeploy agente. Si tienes problemas, actualiza a la última versión antes de ponerte en contacto con AWS Support. Para obtener información acerca de cómo actualizar, consulte [Actualice el CodeDeploy agente](#).

En la siguiente tabla se enumeran todas las versiones del CodeDeploy agente y las funciones y mejoras incluidas en cada versión.


Versión	Fecha de lanzamiento	Detalles
1.7.0	6 de marzo de 2024	<p>Se agregó: un ajuste <code>:disable_imds_v1:</code> de configuración al archivo de configuración del CodeDeploy agente. Use esta configuración para deshabilitar la posibilidad de recurrir a IMDSv1 cuando se produzcan errores en el IMDSv2. El valor predeterminado es (habilita la alternativa). <code>false</code> Para obtener más información, consulte la referencia de configuración CodeDeploy del agente.</p> <p>Se agregó: Support para el sistema operativo Red Hat Enterprise Linux 9 (RHEL 9).</p> <p>Añadido: Support para las versiones 3.1 y 3.2 de Ruby en Ubuntu Server.</p> <p>Corregido: el CodeDeploy agente ahora genera un error fácil de usar si el archivo de configuración del CodeDeploy agente no se carga.</p> <p>Modificado: se actualizó Ruby a la versión 2.7.8-1 en el CodeDeploy agente para Windows.</p>


Versión	Fecha de lanzamiento	Detalles
1.6.0	30 de marzo de 2023	<p>Se ha añadido: compatibilidad con Ruby 3.1, 3.2.</p> <p>Se ha añadido: compatibilidad con Amazon Linux 2023.</p> <p>Se ha añadido: compatibilidad con Windows Server 2022.</p> <p>Se ha modificado: la configuración predeterminada de <code>verbose</code> es ahora <code>false</code> para las instancias de Windows Server. Para seguir imprimiendo los mensajes de depuración en los archivos de registro de Windows, debe configurar <code>verbose</code> en <code>true</code>.</p> <p>Se ha eliminado: compatibilidad con Windows Server 2016 y Windows Server 2012 R2.</p> <p>Se ha eliminado: compatibilidad con Amazon Linux 2018.03.x.</p>

Versión	Fecha de lanzamiento	Detalles
1.5.0	3 de marzo de 2023	<p>Se ha añadido: compatibilidad con Ruby 3.</p> <p>Se ha añadido: compatibilidad con Ubuntu 22.04.</p> <p>Se ha corregido un error que provocaba que, al reiniciar el CodeDeploy agente poco después del arranque, el agente quedara bloqueado.</p> <p>Modificado: el CodeDeploy agente ahora falla en el despliegue de un host al iniciar el agente si el servicio del agente se reinicia inesperadamente mientras se ejecuta un script de enlace. Esta solución permite evitar el tiempo de espera de 70 minutos antes de volver a intentar una implementación.</p> <p>Aviso de obsolescencia: el CodeDeploy agente 1.5.0 es la última versión compatible con Windows Server 2016 y Windows Server 2012 R2.</p> <p>Eliminado: Support para el CodeDeploy agente en Ubuntu 14.04 LTS, Windows Server 2008 R2 y Windows Server 2008 R2 de 32 bits.</p>
1.4.1	6 de diciembre de 2022	<p>Se ha corregido: vulnerabilidad de seguridad relacionada con el registro.</p> <p>Mejora: se mejoró el registro al sondear el comando host.</p>

Versión	Fecha de lanzamiento	Detalles
1.4.0	31 de agosto de 2021	<p>Se ha añadido: compatibilidad con Red Hat Enterprise Linux 8.</p> <p>Añadido: Support para rutas de archivo largas en el CodeDeploy agente para Windows. Para habilitar las rutas de archivo largas, tendrá que configurar la clave de registro de Windows adecuada y, a continuación, reiniciar el agente. Para obtener más información, consulte Las rutas de archivo largas provocan el error “No existe tal archivo o directorio”.</p> <p>Se ha corregido: un problema con la operación de descompresión cuando el disco estaba lleno. El CodeDeploy agente ahora detecta el código de salida 50 de la descompresión, que indica que el disco está lleno, elimina los archivos extraídos parcialmente y crea una excepción si se produce un error en el CodeDeploy servidor. El mensaje de error aparece como un mensaje de error de un evento del ciclo de vida y la implementación a nivel de host se detendrá sin bloquearse y sin agotar el tiempo de espera.</p> <p>Se ha corregido: un problema que provocaba un error en el agente.</p> <p>Se ha corregido: un problema que provocaba que se agotara el tiempo de espera de los enlaces durante una carrera al límite. Los enlaces sin scripts ahora continuarán y ya no causarían errores ni se agotarían los tiempos de espera.</p> <p>Modificado: el update script del bin directorio del CodeDeploy agente se ha eliminado porque ya no se utiliza.</p> <p>Modificado: el CodeDeploy agente para Windows Server ahora incluye Ruby 2.7.</p> <p>Modificado: se agregaron nuevas variables de entorno para que las utilizaran los scripts de enlace en función del origen</p>

Versión	Fecha de lanzamiento	Detalles
		<p data-bbox="625 258 1455 338">del paquete de implementación (Amazon S3 o Amazon S3 GitHub).</p> <p data-bbox="625 386 1490 466">Para obtener más información, consulte Disponibilidad de las variables de entorno para los enlaces.</p> <div data-bbox="625 510 1507 1062" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p data-bbox="657 548 841 583">⚠ Important</p><p data-bbox="704 606 1451 730">Aviso de obsolescencia: el CodeDeploy agente 1.4.0 es la última versión que incluirá instaladores para Windows Server de 32 bits.</p><p data-bbox="704 747 1451 871">Aviso de obsolescencia: el CodeDeploy agente 1.4.0 es la última versión compatible con Windows Server 2008 R2.</p><p data-bbox="704 890 1425 1014">Eliminado: Support para el CodeDeploy agente en las siguientes AMI de Amazon EC2: Amazon Linux 2014.09, 2016.03, 2016.09 y 2017.03.</p></div>

Versión	Fecha de lanzamiento	Detalles
1.3.2	6 de mayo de 2021	<p> Important</p> <p>CodeDeploy El agente 1.3.2 soluciona el problema CVE-2018-1000201, lo que afecta a los hosts Windows que ejecutan el agente. El CVE cita ruby-ffi, que es una dependencia del agente. CodeDeploy y Si su agente se instaló con Amazon EC2 Systems Manager (SSM) y está configurado para actualizarse automáticamente, no es necesario realizar ninguna acción. De lo contrario, es necesario tomar medidas para actualizar manualmente el agente. Para actualizar el agente, siga las instrucciones de Actualizar el CodeDeploy agente en Windows Server.</p> <p>Solucionado: se produjo un problema al instalar el CodeDeploy y agente en Ubuntu 20.04 y versiones posteriores.</p> <p>Se ha corregido: un problema intermitente que se producía al extraer archivos comprimidos porque las rutas relativas no se gestionaban correctamente.</p> <p>Se ha añadido: compatibilidad con AWS PrivateLink y puntos de conexión de VPC para instancias de Windows.</p> <p>Se han añadido mejoras en los AppSpec archivos, tal y como se describe a continuación.</p> <ul style="list-style-type: none">• Ahora puede especificar un nombre de archivo personalizado para el AppSpec archivo al crear una implementación local. Para obtener más información, consulte Creación de una implementación local.• El AppSpec archivo ahora puede tener una extensión de .yaml archivo.

Versión	Fecha de lanzamiento	Detalles
		<ul style="list-style-type: none">Ahora puede sobrescribir los archivos desplegados mediante una nueva <code>file_exists_behavior</code> configuración opcional en el AppSpec archivo. Para obtener más información, consulte AppSpec Sección de «archivos» (solo para despliegues de EC2/on-premise). <p>Actualizado: CodeDeploy ahora usa el AWS SDK para Ruby 3.0.</p>
1.3.1	22 de diciembre de 2020	Se ha corregido: un problema de la versión 1.3.0 que impedía el inicio de las instancias en las instalaciones.
1.3.0	10 de noviembre de 2020	<div data-bbox="630 814 1507 982" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> Important Esta versión es obsoleta.</p></div> <p>Se ha corregido: se ha eliminado un certificado caducado que ya no se utilizaba.</p> <p>Se corrigió: se eliminó el mensaje de aviso del script de desinstalación del agente utilizado por el agente AWS Systems Manager, lo que facilitó la degradación de un host o una flota a una versión anterior del agente.</p>

Versión	Fecha de lanzamiento	Detalles
1.2.1	23 de septiembre de 2020	<p>Se ha modificado: se ha actualizado la dependencia de AWS SDK for Ruby de la versión 2 a la versión 3.</p> <p>Se ha añadido: compatibilidad con IMDSv2. Incluye una alternativa silenciosa a IMDSv1 si las solicitudes HTTP de IMDSv2 fallan.</p> <p>Se ha modificado: se han actualizado las dependencias de Rake y Rubyzip para los parches de seguridad.</p> <p>Se ha corregido: se ha asegurado que un archivo PID vacío devuelva el estado de <code>No CodeDeploy Agent Running</code> y se limpie el archivo PID al iniciar el agente.</p>

Versión	Fecha de lanzamiento	Detalles
1.1.2	4 de agosto de 2020	<p>Se ha añadido: compatibilidad con Ubuntu Server 19.10 y 20.04.</p> <p>Nota: La versión 19.10 llegó a su end-of-life fecha y ya no es compatible con Ubuntu o. CodeDeploy</p> <p>Se ha añadido: mejoras en la eficiencia de la memoria para Linux y Ubuntu para liberar la memoria reservada de manera más oportuna.</p> <p>Se ha añadido: compatibilidad con la “limpieza silenciosa” de Windows Server, que provocaba que el agente no respondiera en algunos casos.</p> <p>Se ha añadido: ignorar los directorios que no están vacíos durante la limpieza para evitar errores durante la implementación.</p> <p>Se agregó: Support for AWS Local Zone en Los Ángeles (LA).</p> <p>Agregado: extraiga la AZ de los metadatos de la instancia para proporcionar compatibilidad con las Zonas AWS Locales.</p> <p>Se ha añadido: los usuarios ahora pueden proporcionar su archivo en subdirectorios y no es necesario que lo almacenen en el directorio raíz.</p> <p>Se ha añadido: se ha detectado un problema con Rubyzip que podría provocar pérdidas de memoria. Se actualizó el comando unzip para intentar usar primero una utilidad de descompresión instalada en el sistema antes de usar Rubyzip.</p> <p>Se ha añadido: <code>:enable_auth_policy:</code> como ajuste de configuración del agente.</p>

Versión	Fecha de lanzamiento	Detalles
		Se ha modificado: las advertencias de descompresión ahora se ignoran para que las implementaciones continúen.
1.1.1.0	30 de junio de 2020	<p>Modificado: el control de versiones del CodeDeploy agente ahora sigue la convención de control de versiones estándar de Ruby.</p> <p>Se ha añadido: nuevo parámetro del comando de instalación y actualización para permitir la instalación de una versión específica del agente desde la línea de comandos.</p> <p>Eliminado: se ha eliminado el CodeDeploy agente Auto Updater para Linux y Ubuntu. Para configurar las actualizaciones automáticas del CodeDeploy agente, consulte Instalar el CodeDeploy agente mediante AWS Systems Manager.</p>
1.0.1.1597	15 de noviembre de 2018	<p>Mejora: CodeDeploy es compatible con Ubuntu 18.04.</p> <p>Mejora: CodeDeploy es compatible con Ruby 2.5.</p> <p>Mejora: CodeDeploy admite puntos finales FIPS. Para obtener más información sobre los puntos de conexión de FIPS, consulte Información general sobre FIPS 140-2. Para ver los puntos finales con los que se puede utilizar CodeBuild, consulte CodeDeployregiones y puntos finales.</p>
1.0.1.1518	12 de junio de 2018	<p>Mejora: se ha corregido un error que provocaba un error cuando el CodeDeploy agente estaba cerrado mientras aceptaba solicitudes de sondeo.</p> <p>Mejora: se ha añadido una función de seguimiento del despliegue que evita que el CodeDeploy agente se cierre cuando hay un despliegue en curso.</p> <p>Mejora: se ha mejorado el desempeño al eliminar archivos.</p>

Versión	Fecha de lanzamiento	Detalles
1.0.1.1458	6 de marzo de 2018	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Mejora: se han mejorado las validaciones de certificado para admitir más autoridades de confianza.</p> <p>Mejora: se ha corregido un problema que provocaba que la CLI local fallara durante una implementación que incluía un evento BeforeInstall del ciclo de vida.</p> <p>Mejora: se ha corregido un problema que podía provocar un error en una implementación activa cuando se actualizaba el CodeDeploy agente.</p>
1.0.1.1352	16 de noviembre de 2017	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Característica: Se introdujo una nueva función para probar y depurar una implementación local de EC2/en una máquina o instancia local en la que esté instalado el CodeDeploy agente.</p>
1.0.1.1106	16 de mayo de 2017	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Característica: se introdujo nueva compatibilidad para el tratamiento de contenido en una ubicación de destino que no formaba parte de la revisión de la aplicación de la implementación correcta más reciente. Las opciones de implementación para el contenido existente incluyen ahora conservar el contenido, sobrescribir el contenido o hacer fracasar la implementación.</p> <p>Mejora: se hizo que el CodeDeploy agente fuera compatible con la versión 2.9.2 de la (2.9.2). AWS SDK for Ruby aws-sdk-core</p>

Versión	Fecha de lanzamiento	Detalles
1.0.1.1095	29 de marzo de 2017	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Mejora: se introdujo el soporte para el CodeDeploy agente en la región de China (Pekín).</p> <p>Mejora: se puede ejecutar Puppet en instancias de Windows Server cuando las invoca un enlace de evento de ciclo de vida.</p> <p>Mejora: se ha mejorado el tratamiento de las operaciones <code>untar</code>.</p>
1.0.1.1067	6 de enero de 2017	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Mejora: se han revisado muchos mensajes de error para incluir más causas específicas para los errores de implementación.</p> <p>Mejora: se ha corregido un problema que impedía que el CodeDeploy agente identificara la revisión de la aplicación correcta para implementarla durante algunas implementaciones.</p> <p>Mejora: se ha revertido el uso de <code>pushd</code> y <code>popd</code> antes y después de la operación <code>untar</code>.</p>
1.0.1.1045	21 de noviembre de 2016	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Mejora: se hizo que el CodeDeploy agente fuera compatible con la versión 2.6.11 de la AWS SDK for Ruby (<code>aws-sdk-core2.6.11</code>).</p>

Versión	Fecha de lanzamiento	Detalles
1.0.1.1037	19 de octubre de 2016	<p data-bbox="623 275 1494 359">Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p data-bbox="623 403 1481 579">El CodeDeploy agente para las instancias de Amazon Linux, RHEL y Ubuntu Server se ha actualizado con el siguiente cambio. Para las instancias de Windows Server, la última versión sigue siendo la 1.0.1.998.</p> <p data-bbox="623 625 1494 753">Mejora: el agente puede determinar ahora qué versión de Ruby está instalada en una instancia para que pueda invocar el script <code>codedeploy-agent</code> usando esa versión.</p>
1.0.1.101 1.1	17 de agosto de 2016	<p data-bbox="623 804 1494 888">Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p data-bbox="623 932 1494 1108">Mejora: se han eliminado los cambios introducidos por la versión 1.0.1.1011 debido a problemas de compatibilidad con el shell. Esta versión del agente es funcionalmente equivalente a la versión 1.0.1.998 publicada el 11 de julio de 2016.</p>

Versión	Fecha de lanzamiento	Detalles
1.0.1.1011	15 de agosto de 2016	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>El CodeDeploy agente para las instancias de Amazon Linux, RHEL y Ubuntu Server se ha actualizado con los siguientes cambios. Para las instancias de Windows Server, la última versión sigue siendo la 1.0.1.998.</p> <p>Característica: Se ha añadido compatibilidad para invocar al CodeDeploy agente mediante el shell bash en sistemas operativos en los que se utiliza el sistema de inicio systemd.</p> <p>Mejora: se habilitó la compatibilidad con todas las versiones de Ruby 2.x en el CodeDeploy agente y en el actualizador de agentes. CodeDeploy CodeDeploy Los agentes actualizados ya no dependen únicamente de Ruby 2.0. (Ruby 2.0 sigue siendo necesario para las versiones deb y rpm del instalador del CodeDeploy agente).</p>
1.0.1.998	11 de julio de 2016	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Mejora: se ha corregido el soporte para ejecutar el CodeDeploy y agente con perfiles de usuario distintos del root. La variable denominada USER se ha sustituido por CODEDEPLOY_USER para evitar conflictos con variables del entorno.</p>

Versión	Fecha de lanzamiento	Detalles
1.0.1.966	16 de junio de 2016	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Funcionalidad: Se introdujo la compatibilidad para ejecutar el CodeDeploy agente con perfiles de usuario distintos del root.</p> <p>Mejora: se ha corregido la compatibilidad para especificar el número de revisiones de la aplicación que se desea que el CodeDeploy agente archive para un grupo de despliegue.</p> <p>Mejora: se hizo que el CodeDeploy agente fuera compatible con la versión 2.3 de la AWS SDK for Ruby (aws-sdk-core 2.3).</p> <p>Mejora: se han solucionado los problemas con la codificación UTF-8 durante las implementaciones.</p> <p>Mejora: se ha mejorado la precisión durante la identificación de nombres de proceso.</p>
1.0.1.950	24 de marzo de 2016	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Característica: se ha añadido compatibilidad con el proxy de instalación.</p> <p>Mejora: se ha actualizado el script de instalación para que no se descargue el CodeDeploy agente si la última versión ya está instalada.</p>

Versión	Fecha de lanzamiento	Detalles
1.0.1.934	11 de febrero de 2016	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Funcionalidad: Se ha introducido el soporte para especificar el número de revisiones de la aplicación que desea que el CodeDeploy agente archive para un grupo de implementación.</p>
1.0.1.880	11 de enero de 2016	<p>Nota: esta versión ya no se admite y podría producir un error en las implementaciones.</p> <p>Mejora: se hizo que el CodeDeploy agente fuera compatible con la versión 2.2 de la AWS SDK for Ruby (aws-sdk-core 2.2). La versión 2.1.2 sigue siendo compatible.</p>
1.0.1.854	17 de noviembre de 2015	<p>Nota: esta versión ya no se admite. Si utiliza esta versión, las implementaciones podrían producir un error.</p> <p>Característica: se ha introducido compatibilidad con el algoritmo hash SHA-256.</p> <p>Característica: se ha introducido compatibilidad con el seguimiento de versiones en los archivos <code>.version</code>.</p> <p>Característica: el ID del grupo de implementaciones ahora está disponible mediante el uso de una variable de entorno.</p> <p>Mejora: se agregó soporte para monitorear los registros de los CodeDeploy agentes mediante Amazon CloudWatch Logs.</p>

Para obtener información relacionada, consulte lo siguiente:

- [Determine la versión del CodeDeploy agente](#)
- [Instala el CodeDeploy agente](#)

Para ver un historial de las versiones de los CodeDeploy agentes, consulte el [repositorio de versiones en GitHub](#).

Administrar el CodeDeploy proceso

Todas las distribuciones Linux del CodeDeploy agente (rpm y deb) utilizan [systemd](#) de forma predeterminada para administrar el proceso del agente.

Sin embargo, las distribuciones rpm y deb incluyen scripts de inicio que residen en `/etc/init.d/codedeploy-agent`. En función de la distribución que utilice, cuando utilice un comando como `sudo service codedeploy-agent restart`, es posible que los scripts en `/etc/init.d` se ejecuten para iniciar el proceso del agente, en lugar de permitir que `systemd` administre el proceso. No es aconsejable ejecutar scripts en `/etc/init.d`.

Para evitar este problema, en el caso de los sistemas compatibles con `systemd`, recomendamos utilizar la utilidad `systemctl` para cualquier operación de agente en lugar de utilizar el comando `service`.

Por ejemplo, para reiniciar el CodeDeploy agente, utilice `sudo systemctl restart codedeploy-agent` en lugar del comando equivalente de la utilidad `service`

Limpieza de revisiones de aplicación y archivos de registro

El CodeDeploy agente archiva las revisiones y los archivos de registro de las instancias. El CodeDeploy agente limpia estos artefactos para conservar espacio en disco.

Registros de despliegue de las revisiones de la aplicación: puede utilizar la opción `max_revision` del archivo de configuración del agente para especificar el número de revisiones de la aplicación que se van a archivar introduciendo cualquier número entero positivo. CodeDeploy también archiva los archivos de registro de esas revisiones. El resto se elimina, a excepción del archivo log de la última implementación correcta. Este archivo de registro se conserva siempre, aunque el número de implementaciones que han producido un error sea superior al número de revisiones conservadas. Si no se especifica ningún valor, CodeDeploy conserva las cinco revisiones más recientes además de la revisión actualmente implementada.

CodeDeploy registros: para las instancias de Amazon Linux, Ubuntu Server y RHEL, el CodeDeploy agente rota los archivos de registro de la `/var/log/aws/codedeploy-agent` carpeta. El archivo de registro rota a las 00:00:00 (hora de la instancia) cada día. Los archivos de registro se

eliminan al cabo de siete días. El patrón de nomenclatura para los archivos de registro movidos es `codedeploy-agent.YYYYMMDD.log`.

Archivos instalados por el agente CodeDeploy

El CodeDeploy agente almacena las revisiones, el historial de despliegue y los scripts de despliegue en el directorio raíz de una instancia. El nombre predeterminado y la ubicación de este directorio es:

'`/opt/codedeploy-agent/deployment-root`' para las instancias de Amazon Linux, Ubuntu Server y RHEL.

'`C:\ProgramData\Amazon\CodeDeploy`' para las instancias de Windows Server.

Puede usar el parámetro `root_dir` del archivo de configuración del CodeDeploy agente para configurar el nombre y la ubicación del directorio. Para obtener más información, consulte [CodeDeploy referencia de configuración del agente](#).

A continuación se muestra un ejemplo del archivo y de la estructura de directorios bajo el directorio raíz. La estructura supone que hay un número N de grupos de implementación y cada grupo de implementación contiene un número N de implementaciones.

```
|--deployment-root/
|-- deployment group 1 ID
|   |-- deployment 1 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment 2 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment N ID
|   |   |-- Contents and logs of the deployment's revision
|-- deployment group 2 ID
|   |-- deployment 1 ID
|   |   |-- bundle.tar
|   |   |-- deployment-archive
|   |   |   |-- contents of the deployment's revision
|   |   |-- logs
|   |   |   |-- scripts.log
|   |-- deployment 2 ID
|   |   |-- bundle.tar
|   |   |-- deployment-archive
|   |   |   |-- contents of the deployment's revision
|   |   |-- logs
|   |   |   |-- scripts.log
```

```

|   |-- deployment N ID
|   |   |-- bundle.tar
|   |   |-- deployment-archive
|   |   |   | -- contents of the deployment's revision
|   |   |-- logs
|   |   |   | -- scripts.log
|-- deployment group N ID
|   |-- deployment 1 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment 2 ID
|   |   |-- Contents and logs of the deployment's revision
|   |-- deployment N ID
|   |   |-- Contents and logs of the deployment's revision
|-- deployment-instructions
|   |-- [deployment group 1 ID]_cleanup
|   |-- [deployment group 2 ID]_cleanup
|   |-- [deployment group N ID]_cleanup
|   |-- [deployment group 1 ID]_install.json
|   |-- [deployment group 2 ID]_install.json
|   |-- [deployment group N ID]_install.json
|   |-- [deployment group 1 ID]_last_successful_install
|   |-- [deployment group 2 ID]_last_successful_install
|   |-- [deployment group N ID]_last_successful_install
|   |-- [deployment group 1 ID]_most_recent_install
|   |-- [deployment group 2 ID]_most_recent_install
|   |-- [deployment group N ID]_most_recent_install
|-- deployment-logs
|   |-- codedeploy-agent-deployments.log

```

- Las carpetas Deployment Group ID (ID de grupo de implementaciones) representan cada uno de sus grupos de implementaciones. El nombre de un directorio de grupo de implementaciones es su ID (por ejemplo, acde1916-9099-7caf-fd21-012345abcdef). Cada directorio de grupo de implementaciones contiene un subdirectorio para cada implementación que se ha intentado en dicho grupo de implementaciones.

Puede usar el [batch-get-deployments](#) comando para buscar un ID de grupo de implementación.

- Las carpetas ID de implementación representan cada implementación en un grupo de implementación. El nombre de cada directorio de implementación es su ID. Cada carpeta contiene:
 - `bundle.tar`, un archivo comprimido con el contenido de la revisión de la implementación. Utilice una utilidad de descompresión zip si desea ver la revisión.

- `deployment-archive`, un directorio que contiene el contenido de la revisión de la implementación.
- `logs`, un directorio que contiene un archivo `scripts.log`. Este archivo muestra el resultado de todos los scripts especificados en el AppSpec archivo de la implementación.

Si desea buscar la carpeta de una implementación pero no conoce su ID de implementación o de grupo de implementación, puede usar la [AWS CodeDeploy consola](#) o el AWS CLI para buscarlos. Para obtener más información, consulte [Ver detalles CodeDeploy de la implementación](#).

El número máximo predeterminado de las implementaciones que se pueden archivar en un grupo de implementaciones es de cinco. Cuando se alcanza ese número, se archivan las implementaciones futuras y se elimina el archivo más antiguo. Puede usar el ajuste `max_revision` del archivo de configuración del CodeDeploy agente para cambiar el valor predeterminado. Para obtener más información, consulte [CodeDeploy referencia de configuración del agente](#).

Note

Si desea recuperar el espacio en disco duro utilizado por las implementaciones archivadas, actualice el valor `max_revisions` a un número bajo, como, por ejemplo, 1 o 2. La siguiente implementación elimina las implementaciones archivadas de modo que el número sea igual al especificado.

- `deployment-instructions` contiene cuatro archivos de texto para cada grupo de implementaciones:
 - `[ID grupo de implementaciones]-cleanup`, un archivo de texto con una versión deshacer de cada comando que se ejecuta durante una implementación. Un nombre de archivo de ejemplo es `acde1916-9099-7caf-fd21-012345abcdef-cleanup`.
 - `[ID de grupo de implementaciones]-install.json`, un archivo JSON creado durante la implementación más reciente. Contiene los comandos que se ejecutan durante la implementación. Un nombre de archivo de ejemplo es `acde1916-9099-7caf-fd21-012345abcdef-install.json`.
 - `[ID de grupo de implementaciones]_last_successfull_install`, un archivo de texto que muestra el directorio de archivos de la última implementación correcta. Este archivo se crea cuando el CodeDeploy agente ha copiado todos los archivos de la aplicación de despliegue a la instancia. El CodeDeploy agente lo utilizará durante la siguiente implementación para determinar qué `BeforeInstall` scripts ejecutar `ApplicationStop` y cuáles. Un nombre de archivo de ejemplo es `acde1916-9099-7caf-fd21-012345abcdef_last_successfull_install`.

- `[ID de grupo de implementaciones]_most_recent_install`, un archivo de texto que muestra el nombre del directorio de archivos de la implementación más reciente. Este archivo se crea cuando los archivos de la implementación se han descargado correctamente. El archivo `[ID de grupo de implementaciones]_last_successful_install` se crea después de este archivo, cuando los archivos descargados se copian en su destino final. Un nombre de archivo de ejemplo es `acde1916-9099-7caf-fd21-012345abcdef_most_recent_install`.
- `deployment-logs` contiene los archivos de registro siguientes:
 - Los archivos `codedeploy-agent.aaaammdd.log` se crean para cada día que hay una implementación. Cada archivo de registro contiene información acerca de las implementaciones del día. Estos archivos de registro podrían ser útiles para depurar problemas como un problema de permisos. El archivo de registro se denomina inicialmente `codedeploy-agent.log`. Al día siguiente, se inserta la fecha de sus implementaciones en el nombre del archivo. Por ejemplo, si hoy es 3 de enero de 2018, puede ver información acerca de todas las implementaciones de hoy en `codedeploy-agent.log`. Mañana, 4 de enero de 2018, se cambia el nombre del archivo de registro a `codedeploy-agent.20180103.log`.
 - `codedeploy-agent-deployments.log` compila el contenido de los `scripts.log` archivos de cada implementación. Los archivos `scripts.log` se encuentran en la subcarpeta `logs` debajo de cada carpeta `Deployment ID`. Las entradas de este archivo van precedidas de un ID de implementación. Por ejemplo, `"[d-ABCDEF123]LifecycleEvent - BeforeInstall"` podría haberse escrito durante una implementación con un ID de `d-ABCDEF123`. Cuando `codedeploy-agent-deployments.log` alcanza su tamaño máximo, el CodeDeploy agente continúa escribiendo en él mientras elimina el contenido antiguo.

Gestión de las operaciones CodeDeploy de los agentes

Las instrucciones de esta sección muestran cómo instalar, desinstalar, reinstalar o actualizar el CodeDeploy agente y cómo comprobar que el CodeDeploy agente se está ejecutando.

Temas

- [Compruebe que el CodeDeploy agente se esté ejecutando](#)
- [Determine la versión del CodeDeploy agente](#)
- [Instala el CodeDeploy agente](#)
- [Actualice el CodeDeploy agente](#)
- [Desinstale el CodeDeploy agente](#)

- [Enviar los registros CodeDeploy de los agentes a CloudWatch](#)

Compruebe que el CodeDeploy agente se esté ejecutando

En esta sección, se describen los comandos que se deben ejecutar si sospecha que el CodeDeploy agente ha dejado de ejecutarse en una instancia.

Temas

- [Compruebe que el CodeDeploy agente de Amazon Linux o RHEL se esté ejecutando](#)
- [Compruebe que el CodeDeploy agente del servidor Ubuntu esté en ejecución](#)
- [Compruebe que el CodeDeploy agente de Windows Server se esté ejecutando](#)

Compruebe que el CodeDeploy agente de Amazon Linux o RHEL se esté ejecutando

Para comprobar si el CodeDeploy agente está instalado y en ejecución, inicie sesión en la instancia y ejecute el siguiente comando:

```
systemctl status codedeploy-agent
```

Si el comando devuelve un error, significa que el CodeDeploy agente no está instalado. Instálelo como se describe en [Instalar el CodeDeploy agente para Amazon Linux o RHEL](#).

Si el CodeDeploy agente está instalado y en ejecución, debería aparecer un mensaje como `The AWS CodeDeploy agent is running`.

Si ve un mensaje similar a `error: No AWS CodeDeploy agent running`, inicie el servicio y ejecute uno por uno los dos comandos siguientes:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Compruebe que el CodeDeploy agente del servidor Ubuntu esté en ejecución

Para comprobar si el CodeDeploy agente está instalado y en ejecución, inicia sesión en la instancia y ejecuta el siguiente comando:


```
systemctl status codedeploy-agent
```

Si el comando devuelve un error, significa que el CodeDeploy agente no está instalado. Instálelo como se describe en [Instale el CodeDeploy agente para Ubuntu Server](#).

Si el CodeDeploy agente está instalado y en ejecución, debería aparecer un mensaje como `The AWS CodeDeploy agent is running`.

Si ve un mensaje similar a `error: No AWS CodeDeploy agent running`, inicie el servicio y ejecute uno por uno los dos comandos siguientes:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

Compruebe que el CodeDeploy agente de Windows Server se esté ejecutando

Para comprobar si el CodeDeploy agente está instalado y en ejecución, inicia sesión en la instancia y ejecuta el siguiente comando:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Debería ver una salida similar a esta:

Status	Name	DisplayName
-----	----	-----
Running	codedeployagent	CodeDeploy Host Agent Service

Si el comando devuelve un error, significa que el CodeDeploy agente no está instalado. Instálelo como se describe en [Instale el CodeDeploy agente para Windows Server](#).

Si Status muestra algo distinto de Running, inicie el servicio con el comando siguiente:

```
powershell.exe -Command Start-Service -Name codedeployagent
```

Puede reiniciar el servicio con el comando siguiente:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Puede detener el servicio con el comando siguiente:

```
powershell.exe -Command Stop-Service -Name codedeployagent
```

Determine la versión del CodeDeploy agente

Puedes determinar la versión del CodeDeploy agente que se ejecuta en tu instancia de dos maneras.

En primer lugar, a partir de la versión 1.0.1.854 del CodeDeploy agente, puede ver el número de versión en un `.version` archivo de la instancia. La tabla siguiente muestra la ubicación y un ejemplo de cadena de versión para los distintos sistemas operativos compatibles.

Sistema operativo	Ubicación del archivo	Ejemplo de cadena agent_version
Amazon Linux y Red Hat Enterprise Linux (RHEL)	<code>/opt/codedeploy-agent/.version</code>	<code>OFFICIAL_1.0.1.854_rpm</code>
Servidor Ubuntu	<code>/opt/codedeploy-agent/.version</code>	<code>OFFICIAL_1.0.1.854_deb</code>
Windows Server	<code>C:\ProgramData\Amazon\CodeDeploy\version</code>	<code>OFFICIAL_1.0.1.854_msi</code>

En segundo lugar, puede ejecutar un comando en una instancia para determinar la versión del CodeDeploy agente.

Temas

- [Determinación de la versión en Amazon Linux o RHEL](#)
- [Determinación de la versión en Ubuntu Server](#)
- [Determinación de la versión en Windows Server](#)

Determinación de la versión en Amazon Linux o RHEL

Inicie sesión en la instancia y ejecute el comando siguiente:

```
sudo yum info codedeploy-agent
```

Determinación de la versión en Ubuntu Server

Inicie sesión en la instancia y ejecute el comando siguiente:

```
sudo dpkg -s codedeploy-agent
```

Determinación de la versión en Windows Server

Inicie sesión en la instancia y ejecute el comando siguiente:

```
sc qdescription codedeployagent
```

Instala el CodeDeploy agente

Para usarlo CodeDeploy en instancias EC2 o servidores locales, primero se debe instalar el CodeDeploy agente. Se recomienda instalar y actualizar el CodeDeploy agente con. AWS Systems Manager Para obtener más información acerca de Systems Manager, consulte [¿Qué es AWS Systems Manager?](#). Puede configurar la instalación y las actualizaciones programadas del CodeDeploy agente con Systems Manager en la consola al crear los grupos de despliegue.

También puede instalar el CodeDeploy agente directamente desde un bucket de S3 con la línea de comandos.

Para obtener información sobre la versión que se recomienda instalar, consulte [Historial de versiones del agente CodeDeploy](#).

Temas

- [Instale el CodeDeploy agente mediante AWS Systems Manager](#)
- [Instale el CodeDeploy agente mediante la línea de comandos](#)

Instale el CodeDeploy agente mediante AWS Systems Manager

Puede usar AWS Management Console o AWS CLI para instalar el CodeDeploy agente en sus instancias de Amazon EC2 o locales mediante. AWS Systems Manager Puede elegir entre instalar una versión específica o instalar siempre la última versión del agente. Para obtener más información AWS Systems Manager, consulte [Qué es](#). AWS Systems Manager

Se AWS Systems Manager recomienda utilizar el método para instalar y actualizar el CodeDeploy agente. También puede instalar el CodeDeploy agente desde un bucket de Amazon S3. Para obtener más información acerca del uso del enlace de descarga de Amazon S3, consulte [Instale el CodeDeploy agente mediante la línea de comandos](#).

Temas

- [Requisitos previos](#)
- [Instalación del agente de CodeDeploy](#)

Requisitos previos

Siga los pasos de [Empezar con CodeDeploy](#) para configurar los permisos de IAM y la AWS CLI.

Si va a instalar el CodeDeploy agente en un servidor local con Systems Manager, debe registrar el servidor local con Amazon EC2 Systems Manager. Para obtener más información, consulte [Configuración de Systems Manager en entornos híbridos](#) en la Guía del usuario de AWS Systems Manager .

Instalación del agente de CodeDeploy

Antes de poder usar Systems Manager para instalar el CodeDeploy agente, debe asegurarse de que la instancia está configurada correctamente para Systems Manager.

Instalación o actualización del agente de SSM

En una instancia Amazon EC2, el CodeDeploy agente requiere que la instancia ejecute la versión 2.3.274.0 o posterior. Antes de instalar el CodeDeploy agente, actualice o instale el agente SSM en la instancia si aún no lo ha hecho.

El agente SSM viene preinstalado en algunas AMI de Amazon EC2 proporcionadas por AWS. Para obtener más información, consulte [Imágenes de máquina de Amazon \(AMI\) con el agente de SSM preinstalado](#).

Note

Asegúrese de que el agente también admita el sistema operativo de la instancia. CodeDeploy Para obtener más información, consulte [Sistemas operativos compatibles con el agente CodeDeploy](#) .

Para obtener información sobre la instalación o actualización del agente de SSM en una instancia con Linux, consulte [Instalación y configuración del agente de SSM en instancias de Linux](#) en la Guía del usuario de AWS Systems Manager .

Para obtener información acerca de la instalación o la actualización del agente de SSM en una instancia con Windows Server, consulte [Instalación y configuración del agente de SSM en instancias de Windows](#) en la Guía del usuario de AWS Systems Manager .

(Opcional) Verificación de los requisitos previos de Systems Manager

Antes de usar Systems Manager Run Command para instalar el CodeDeploy agente, compruebe que las instancias cumplen los requisitos mínimos de Systems Manager. Para obtener más información, consulte [Configuración de AWS Systems Manager](#) en la Guía del usuario de AWS Systems Manager .

Instale el CodeDeploy agente

Con SSM, puede instalarlo CodeDeploy una vez o programar la instalación de nuevas versiones.

Para instalar el CodeDeploy agente, elija el `AWSCodeDeployAgent` paquete siguiendo los pasos que se indican en [Instalar o actualizar paquetes con el AWS Systems Manager distribuidor](#).

Instale el CodeDeploy agente mediante la línea de comandos

Note

Se recomienda instalar el CodeDeploy agente con el AWS Systems Manager fin de poder configurar las actualizaciones programadas del agente. Para obtener más información, consulte [Instale el CodeDeploy agente mediante AWS Systems Manager](#).

Utilice los siguientes temas para instalar y ejecutar el CodeDeploy agente mediante la línea de comandos.

Temas

- [Instalar el CodeDeploy agente para Amazon Linux o RHEL](#)
- [Instale el CodeDeploy agente para Ubuntu Server](#)
- [Instale el CodeDeploy agente para Windows Server](#)

Instalar el CodeDeploy agente para Amazon Linux o RHEL

Inicie sesión en la instancia y ejecute uno a uno los comandos siguientes. Ejecutar primero el comando `sudo yum update` se considera una práctica recomendada cuando se usa `yum` para instalar paquetes, pero puede omitirlo si no desea actualizar todos sus paquetes.

```
sudo yum update
```

```
sudo yum install ruby
```

```
sudo yum install wget
```

(Opcional) Para limpiar la AMI de cualquier información de almacenamiento en caché del agente anterior, ejecute el siguiente script:

```
#!/bin/bash
CODEDEPLOY_BIN="/opt/codedeploy-agent/bin/codedeploy-agent"
$CODEDEPLOY_BIN stop
yum erase codedeploy-agent -y
```

Vaya al directorio de inicio:

```
cd /home/ec2-user
```

Note

En el comando anterior, `/home/ec2-user` representa el nombre de usuario predeterminado de una instancia de Amazon EC2 de Amazon Linux o RHEL. Si la instancia se creó con una AMI personalizada, el propietario de la AMI puede haber especificado otro nombre de usuario predeterminado.

Descargue el instalador del CodeDeploy agente:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```

bucket-name es el nombre del depósito de Amazon S3 que contiene los archivos del kit de CodeDeploy recursos de su región y *region-identifier es el identificador* de su región.

Por ejemplo:

```
https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install
```

Para obtener una lista de los nombres de buckets e identificadores de región, consulte [Nombres de bucket del kit de recursos por región](#).

Defina los permisos de ejecución en el archivo `install`:

```
chmod +x ./install
```

Para instalar la versión más reciente del agente: CodeDeploy

- ```
sudo ./install auto
```

Para instalar una versión específica del CodeDeploy agente:

- Obtenga una lista de versiones disponibles en su región:

```
aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier | grep '\.rpm$'
```

- Instale una de las versiones:

```
sudo ./install auto -v releases/codedeploy-agent-version.noarch.rpm
```

#### Note

AWS admite la última versión secundaria del CodeDeploy agente. Actualmente, la última versión secundaria es la 1.7.x.

Para comprobar que el servicio se está ejecutando, ejecute el siguiente comando:

```
systemctl status codedeploy-agent
```

Si el CodeDeploy agente está instalado y en ejecución, debería aparecer un mensaje como. `The AWS CodeDeploy agent is running`

Si ve un mensaje similar a `error: No AWS CodeDeploy agent running`, inicie el servicio y ejecute uno por uno los dos comandos siguientes:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

### Instale el CodeDeploy agente para Ubuntu Server

#### Note

Se recomienda instalar el CodeDeploy agente con AWS Systems Manager para poder configurar las actualizaciones programadas del agente. Para obtener más información, consulte [Instale el CodeDeploy agente mediante AWS Systems Manager](#).

Para instalar el CodeDeploy agente en el servidor Ubuntu

1. Inicie sesión en la instancia.
2. Introduzca los comandos siguientes, uno tras otro:

```
sudo apt update
```

```
sudo apt install ruby-full
```

```
sudo apt install wget
```

3. Escriba el siguiente comando:

```
cd /home/ubuntu
```

*/home/ubuntu* representa el nombre de usuario predeterminado de una instancia de Ubuntu Server. Si la instancia se creó con una AMI personalizada, el propietario de la AMI puede haber especificado otro nombre de usuario predeterminado.

4. Escriba el siguiente comando:

```
wget https://bucket-name.s3.region-identifier.amazonaws.com/latest/install
```



*bucket-name* es el nombre del depósito de Amazon S3 que contiene los archivos del kit de CodeDeploy recursos de su región y *region-identifier es el identificador* de su región.

Por ejemplo:

```
https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/
install
```

Para obtener una lista de los nombres de buckets e identificadores de región, consulte [Nombres de bucket del kit de recursos por región](#).

5. Escriba el siguiente comando:

```
chmod +x ./install
```

6. Realice una de las siguientes acciones siguientes:

- Para instalar la última versión del CodeDeploy agente en cualquier versión compatible de Ubuntu Server, excepto la 20.04:

```
sudo ./install auto
```

- Para instalar la última versión del CodeDeploy agente en Ubuntu Server 20.04:

#### Note

Escribir el resultado en un archivo de registro temporal es una solución alternativa que debería utilizarse mientras solucionamos un error conocido en el script `install` de Ubuntu Server 20.04.


```
sudo ./install auto > /tmp/logfile
```

- Para instalar una versión específica del CodeDeploy agente en cualquier versión compatible de Ubuntu Server, excepto la 20.04:
  - Obtenga una lista de versiones disponibles en su región:

```
aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier | grep '\.deb$'
```

- Instale una de las versiones:

```
sudo ./install auto -v releases/codedeploy-agent-###.deb
```

 Note


AWS admite la última versión secundaria del CodeDeploy agente. Actualmente, la última versión secundaria es la 1.7.x.

- Para instalar una versión específica del CodeDeploy agente en Ubuntu Server 20.04:
  - Obtenga una lista de versiones disponibles en su región:


```
aws s3 ls s3://aws-codedeploy-region-identifier/releases/ --region region-identifier | grep '\.deb$'
```

- Instale una de las versiones:

```
sudo ./install auto -v releases/codedeploy-agent-###.deb > /tmp/logfile
```

 Note

Escribir el resultado en un archivo de registro temporal es una solución alternativa que debería utilizarse mientras solucionamos un error conocido en el script `install` de Ubuntu Server 20.04.

 Note

AWS es compatible con la última versión secundaria del CodeDeploy agente. Actualmente, la última versión secundaria es la 1.7.x.

## Para comprobar que el servicio está en ejecución

1. Escriba el siguiente comando:

```
systemctl status codedeploy-agent
```

Si el CodeDeploy agente está instalado y en ejecución, debería aparecer un mensaje como. The AWS CodeDeploy agent is running

2. Si ve un mensaje similar a `error: No AWS CodeDeploy agent running`, inicie el servicio y ejecute uno por uno los dos comandos siguientes:

```
systemctl start codedeploy-agent
```

```
systemctl status codedeploy-agent
```

## Instale el CodeDeploy agente para Windows Server

En las instancias de Windows Server, puede usar uno de estos métodos para descargar e instalar el CodeDeploy agente:

- Uso AWS Systems Manager (recomendado)
- Ejecute una serie de PowerShell comandos de Windows.
- Elegir un enlace de descarga directa.
- Ejecutar un comando copy de Amazon S3.

### Note

La carpeta en la que está instalado el CodeDeploy agente es `C:\Program Data\Amazon\CodeDeploy`. Asegúrese de que no haya nudos de directorio ni enlaces simbólicos en esta ruta.

## Temas

- [Uso de Systems Manager](#)
- [Utilice Windows PowerShell](#)

- [Uso de un enlace directo](#)
- [Uso de un comando copy de Amazon S3](#)

## Uso de Systems Manager

Siga las instrucciones [Instale el CodeDeploy agente mediante AWS Systems Manager](#) para instalar el CodeDeploy agente.

### Utilice Windows PowerShell

Inicia sesión en la instancia y ejecuta los siguientes comandos en Windows PowerShell:

1. Es necesario que todos los scripts y archivos de configuración que se hayan descargado de Internet estén firmados por un editor de confianza. Si se le pide que cambie la política de ejecución, escriba "Y".

```
Set-ExecutionPolicy RemoteSigned
```

2. Carga el AWS Tools for Windows PowerShell.

```
Import-Module AWSPowerShell
```

3. Cree un directorio en el que se descargue el archivo de instalación del CodeDeploy agente.

```
New-Item -Path "c:\temp" -ItemType "directory" -Force
```

4. Configure AWS las credenciales mediante los `Initialize-AWSDefaultConfiguration` comandos `Set-AWSCredential` y. Para obtener más información, consulte [Uso de AWS credenciales](#) en la Guía del PowerShell usuario de AWS las herramientas.
5. Descargue el archivo CodeDeploy de instalación del agente.

#### Note

AWS es compatible con la última versión secundaria del CodeDeploy agente. Actualmente, la última versión secundaria es la 1.7.x.

Para instalar la última versión del CodeDeploy agente:

- ```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi
```

Para instalar una versión específica del CodeDeploy agente:

- ```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key releases/codedeploy-agent-###.msi -File c:\temp\codedeploy-agent.msi
```

*bucket-name* es el nombre del depósito de Amazon S3 que contiene los archivos del kit CodeDeploy de recursos de su región. Por ejemplo, en el caso de la región Este de EE. UU. (Ohio), sustituya *bucket-name* por `aws-codedeploy-us-east-2`. Para obtener una lista de los nombres de los buckets, consulte [Nombres de bucket del kit de recursos por región](#).

6. Ejecute el archivo de instalación del CodeDeploy agente.

```
c:\temp\codedeploy-agent.msi /quiet /l c:\temp\host-agent-install-log.txt
```

Para comprobar que el servicio se está ejecutando, ejecute el siguiente comando:

```
powershell.exe -Command Get-Service -Name codedeployagent
```

Si el CodeDeploy agente se acaba de instalar y no se ha iniciado, después de ejecutar el `Get-Service` comando, en Estado, debería ver **Start...**:

| Status   | Name            | DisplayName                   |
|----------|-----------------|-------------------------------|
| -----    | ----            | -----                         |
| Start... | codedeployagent | CodeDeploy Host Agent Service |

Si el CodeDeploy agente ya se está ejecutando, después de ejecutar el `Get-Service` comando, en Estado, debería ver **Running**:

| Status  | Name            | DisplayName                   |
|---------|-----------------|-------------------------------|
| -----   | ----            | -----                         |
| Running | codedeployagent | CodeDeploy Host Agent Service |

## Uso de un enlace directo

Si la configuración de seguridad del navegador de la instancia de Windows Server proporciona los permisos (por ejemplo, para `https://s3.*.amazonaws.com`), puede utilizar un enlace directo a su región para descargar el CodeDeploy agente y, a continuación, ejecutar el instalador manualmente.

El enlace es:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent.msi
```

... donde *region* es la región de AWS en la que está implementando la aplicación.

Por ejemplo:

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latest/codedeploy-agent.msi
```

### Important

Obtenga el `.msi` archivo de la misma región que CodeDeploy la aplicación. La elección de una región diferente puede provocar errores de `inconsistent region` en el archivo `codedeploy-agent-log` al ejecutar el archivo `.msi`.

## Uso de un comando copy de Amazon S3

Si AWS CLI está instalado en la instancia, puede usar el comando `cp` de Amazon S3 para descargar el CodeDeploy agente y, a continuación, ejecutar el instalador manualmente. Para obtener más información, consulte [Instalar AWS Command Line Interface en Microsoft Windows](#).

El comando de Amazon S3 es:

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent.msi codedeploy-agent.msi
--region region
```

... donde *region* es la región de AWS en la que está implementando la aplicación.

Por ejemplo:

```
aws s3 cp s3://aws-codedeploy-af-south-1/latest/codedeploy-agent.msi codedeploy-agent.msi --region af-south-1
```

## Actualice el CodeDeploy agente

Puede configurar las actualizaciones automáticas y programadas del CodeDeploy agente en todos los sistemas operativos compatibles mediante AWS Systems Manager. También puede forzar las actualizaciones en todos los sistemas operativos compatibles ejecutando un comando en una instancia.

### Temas

- [Actualizar el CodeDeploy agente en Amazon Linux o RHEL](#)
- [Actualiza el CodeDeploy agente en el servidor Ubuntu](#)
- [Actualice el CodeDeploy agente en Windows Server](#)

## Actualizar el CodeDeploy agente en Amazon Linux o RHEL

Para configurar las actualizaciones automáticas y programadas del CodeDeploy agente mediante AWS Systems Manager, siga los pasos que se indican en [Instalar el CodeDeploy agente con AWS Systems Manager](#).

Si quieres forzar una actualización del CodeDeploy agente, inicia sesión en la instancia y ejecuta el siguiente comando:

```
sudo /opt/codedeploy-agent/bin/install auto
```

## Actualiza el CodeDeploy agente en el servidor Ubuntu

Para configurar las actualizaciones automáticas y programadas del CodeDeploy agente mediante AWS Systems Manager, siga los pasos que se indican en [Instalar el CodeDeploy agente con AWS Systems Manager](#).

Si quieres forzar una actualización del CodeDeploy agente, inicia sesión en la instancia y ejecuta el siguiente comando:

```
sudo /opt/codedeploy-agent/bin/install auto
```

## Actualice el CodeDeploy agente en Windows Server

Puede activar las actualizaciones automáticas del CodeDeploy agente con AWS Systems Manager. Utilice State Manager para configurar una programación de actualización de las instancias de Amazon EC2 o en las instalaciones mediante la creación de una asociación con el Administrador de estados de Systems Manager. También puede actualizar el CodeDeploy agente manualmente desinstalando la versión actual e instalando una más reciente.

### Temas

- [Configure la actualización automática CodeDeploy del agente con AWS Systems Manager](#)
- [Actualice el CodeDeploy agente manualmente](#)
- [\(Obsoleto\) Actualice el CodeDeploy agente con Windows Server Updater](#)

### Configure la actualización automática CodeDeploy del agente con AWS Systems Manager

Para configurar Systems Manager y activar las actualizaciones automáticas del CodeDeploy agente, siga las instrucciones de [Instalar el CodeDeploy agente mediante AWS Systems Manager](#).

### Actualice el CodeDeploy agente manualmente

Para actualizar el CodeDeploy agente manualmente, puede instalar la última versión desde la CLI o mediante Systems Manager. Siga las instrucciones que se indican en [Instalar el CodeDeploy agente](#). Se recomienda desinstalar las versiones anteriores del CodeDeploy agente siguiendo las instrucciones de la sección [Desinstalar el CodeDeploy agente](#).

### (Obsoleto) Actualice el CodeDeploy agente con Windows Server Updater

#### Note

El actualizador de CodeDeploy agentes para Windows Server está obsoleto y no se actualizará a ninguna versión posterior a la 1.0.1.1597.

Para habilitar las actualizaciones automáticas del CodeDeploy agente, instale el actualizador de CodeDeploy agentes para Windows Server en instancias nuevas o existentes. El actualizador comprueba periódicamente si existen versiones nuevas. Cuando detecta una versión nueva del agente, el actualizador desinstala la versión actual, si la hay, antes de instalar la última versión.



Si una implementación ya está en curso cuando el actualizador detecta una nueva versión, continuará hasta su finalización. Si se intenta iniciar una implementación durante el proceso de actualización, la implementación falla.

Si quiere forzar una actualización del CodeDeploy agente, siga las instrucciones que se indican en [Instale el CodeDeploy agente para Windows Server](#).

En las instancias de Windows Server, puede descargar e instalar el actualizador de CodeDeploy agentes ejecutando PowerShell comandos de Windows, utilizando un enlace de descarga directa o ejecutando un comando de copia de Amazon S3.

## Temas

- [Utilice Windows PowerShell](#)
- [Uso de un enlace directo](#)
- [Uso de un comando copy de Amazon S3](#)

## Utilice Windows PowerShell

Inicia sesión en la instancia y ejecuta los siguientes comandos en Windows PowerShell, de uno en uno:

```
Set-ExecutionPolicy RemoteSigned
```

Si se te pide que cambies la política de ejecución, elige **Y** que Windows PowerShell requiera que todos los scripts y archivos de configuración descargados de Internet estén firmados por un editor de confianza.

```
Import-Module AWSPowerShell
```

```
New-Item -Path "c:\temp" -ItemType "directory" -Force
```

```
powershell.exe -Command Read-S3Object -BucketName bucket-name -Key latest/codedeploy-agent-updater.msi -File c:\temp\codedeploy-agent-updater.msi
```

```
c:\temp\codedeploy-agent-updater.msi /quiet /l c:\temp\host-agent-updater-log.txt
```

```
powershell.exe -Command Get-Service -Name codedeployagent
```

*bucket-name* es el nombre del depósito de Amazon S3 que contiene los archivos del kit CodeDeploy de recursos de su región. Por ejemplo, en el caso de la región Este de EE. UU. (Ohio), sustituya *bucket-name* por `aws-codedeploy-us-east-2`. Para obtener una lista de los nombres de los buckets, consulte [Nombres de bucket del kit de recursos por región](#).

Si necesita solucionar un error en el proceso de actualización, escriba el siguiente comando para abrir el archivo de registro del CodeDeploy actualizador del agente:

```
notepad C:\ProgramData\Amazon\CodeDeployUpdater\log\codedeploy-agent.updater.log
```

### Uso de un enlace directo

Si la configuración de seguridad del navegador de la instancia de Windows Server proporciona los permisos necesarios (por ejemplo, para `http://s3.*.amazonaws.com`), puede utilizar un enlace directo para descargar el CodeDeploy actualizador del agente.

El enlace es:

```
https://s3.region.amazonaws.com/aws-codedeploy-region/latest/codedeploy-agent-updater.msi
```

... donde *region* es la región de AWS en la que está actualizando la aplicación.

Por ejemplo:

```
https://s3.af-south-1.amazonaws.com/aws-codedeploy-af-south-1/latest/codedeploy-agent-updater.msi
```

### Uso de un comando copy de Amazon S3

Si AWS CLI está instalado en la instancia, puede usar el comando `cp` de Amazon S3 para descargar el actualizador del CodeDeploy agente y, a continuación, ejecutar el instalador manualmente. Para obtener más información, consulte [Instalar AWS Command Line Interface en Microsoft Windows](#).

El comando de Amazon S3 es:

```
aws s3 cp s3://aws-codedeploy-region/latest/codedeploy-agent-updater.msi codedeploy-agent-updater.msi --region region
```

... donde *region* es la región de AWS en la que está actualizando la aplicación.

Por ejemplo:

```
aws s3 cp s3://aws-codedeploy-af-south-1/latest/codedeploy-agent-updater.msi
codedeploy-agent-updater.msi --region af-south-1
```

## Desinstale el CodeDeploy agente

Puede eliminar el CodeDeploy agente de las instancias cuando ya no lo necesite o cuando desee realizar una instalación nueva.

### Desinstalar el CodeDeploy agente de Amazon Linux o RHEL

Para desinstalar el CodeDeploy agente, inicie sesión en la instancia y ejecute el siguiente comando:

```
sudo yum erase codedeploy-agent
```

### Desinstala el CodeDeploy agente del servidor Ubuntu

Para desinstalar el CodeDeploy agente, inicia sesión en la instancia y ejecuta el siguiente comando:

```
sudo dpkg --purge codedeploy-agent
```

### Desinstale el CodeDeploy agente de Windows Server

Para desinstalar el CodeDeploy agente, inicia sesión en la instancia y ejecuta los tres comandos siguientes, de uno en uno:

```
wmic
```

```
product where name="CodeDeploy Host Agent" call uninstall /nointeractive
```

```
exit
```

También puedes iniciar sesión en la instancia y, en el Panel de control, abrir Programas y características, elegir CodeDeploy Host Agent y, a continuación, elegir Desinstalar.

## Enviar los registros CodeDeploy de los agentes a CloudWatch

Puede enviar las métricas y los datos de registro del CodeDeploy agente CloudWatch mediante el [CloudWatch agente unificado](#) o, más simplemente, el CloudWatch agente.

Siga las instrucciones siguientes para instalar el CloudWatch agente y configurarlo para su uso con CodeDeploy los agentes.

### Requisitos previos

Antes de empezar, complete las siguientes tareas:

- Instale el CodeDeploy agente y asegúrese de que se esté ejecutando. Para obtener más información, consulte [Instala el CodeDeploy agente](#) y [Compruebe que el CodeDeploy agente se esté ejecutando](#).
- Instale el CloudWatch agente. Para obtener más información, consulte [Instalación del CloudWatch agente](#).
- Añada los siguientes permisos al perfil de instancia de CodeDeploy IAM:
  - CloudWatchLogsFullAccess
  - CloudWatchAgentServerPolicy

Para obtener más información sobre el perfil de la CodeDeploy instancia, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2. Empezar con CodeDeploy](#)

### Configure el CloudWatch agente para recopilar CodeDeploy registros

Puede configurar el CloudWatch agente siguiendo un asistente o creando o editando manualmente un archivo de configuración.

Para configurar el CloudWatch agente mediante el asistente (Linux)

1. Ejecute el asistente, tal y como se describe en [Ejecutar el asistente de configuración del CloudWatch agente](#).
2. En el asistente, cuando se le pida Do you want to monitor any log files?, introduzca **1**.
3. Especifique el archivo de registro del CodeDeploy agente de la siguiente manera:

1. Para Log file path introducir la ruta del archivo de CodeDeploy registro, por ejemplo: **/var/log/aws/codedeploy-agent/codedeploy-agent.log**.
2. Para Log group name, introduzca un nombre de grupo de registro, por ejemplo: **codedeploy-agent-log**.
3. Para Log stream name, introduzca un nombre de flujo de registro, por ejemplo: **{instance\_id}-codedeploy-agent-log**.
4. Cuando se le pida Do you want to specify any additional log files?, introduzca **1**.
5. Especifique los registros de despliegue del CodeDeploy agente, de la siguiente manera:
  1. Para Log file path introducir la ruta del archivo de registro de CodeDeploy despliegue, por ejemplo: **/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log**.
  2. Para Log group name, introduzca un nombre de grupo de registro, por ejemplo: **codedeploy-agent-deployment-log**.
  3. Para Log stream name, introduzca un nombre de flujo de registro, por ejemplo: **{instance\_id}-codedeploy-agent-deployment-log**.
6. Cuando se le pida Do you want to specify any additional log files?, introduzca **1**.
7. Especifique los registros del actualizador del CodeDeploy agente, de la siguiente manera:
  1. Para Log file path introducir la ruta del archivo de registro del CodeDeploy actualizador, por ejemplo: **./tmp/codedeploy-agent.update.log**
  2. Para Log group name, introduzca un nombre de grupo de registro, por ejemplo: **codedeploy-agent-updater-log**.
  3. Para Log stream name, introduzca un nombre de flujo de registro, por ejemplo: **{instance\_id}-codedeploy-agent-updater-log**.

Para configurar el CloudWatch agente mediante el asistente (Windows)

1. Ejecute el asistente, tal y como se describe en [Ejecutar el asistente de configuración del CloudWatch agente](#).
2. En el asistente, cuando se le pida Do you want to monitor any customized log files?, introduzca **1**.

3. Especifique el archivo de CodeDeploy registro de la siguiente manera:
  1. Para Log file path introducir la ruta o el archivo de registro del CodeDeploy agente, por ejemplo:**C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt**.
  2. Para Log group name, introduzca un nombre de grupo de registro, por ejemplo:**codedeploy-agent-log**.
  3. Para Log stream name, introduzca un nombre de flujo de registro, por ejemplo:**{instance\_id}-codedeploy-agent-log**.
4. Cuando se le pida Do you want to specify any additional log files?, introduzca **1**.
5. Especifique los registros de despliegue del CodeDeploy agente, de la siguiente manera:
  1. Para Log file path introducir la ruta del archivo CodeDeploy de registro de despliegue, por ejemplo:**C:\ProgramData\Amazon\CodeDeploy\deployment-logs\codedeploy-agent-deployments.log**.
  2. Para Log group name, introduzca un nombre de grupo de registro, por ejemplo:**codedeploy-agent-deployment-log**.
  3. Para Log stream name, introduzca un nombre de flujo de registro, por ejemplo:**{instance\_id}-codedeploy-agent-deployment-log**.

Para configurar el CloudWatch agente mediante la creación o edición manual de un archivo de configuración (Linux)

1. Cree o edite el archivo de configuración del CloudWatch agente como se describe en [Crear o editar manualmente el archivo de configuración del CloudWatch agente](#).
2. Asegúrese de que el archivo se llame /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json y de que contenga el siguiente código:

```
...
"logs": {
 "logs_collected": {
 "files": {
 "collect_list": [
 {
 "file_path": "/var/log/aws/codedeploy-agent/codedeploy-
agent.log",
 "log_group_name": "codedeploy-agent-log",
```

```

 "log_stream_name": "{instance_id}-agent-log"
 },
 {
 "file_path": "/opt/codedeploy-agent/deployment-root/deployment-
logs/codedeploy-agent-deployments.log",
 "log_group_name": "codedeploy-agent-deployment-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-deployment-
log"
 },
 {
 "file_path": "/tmp/codedeploy-agent.update.log",
 "log_group_name": "codedeploy-agent-updater-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-updater-log"
 }
]
}
}
...

```

Para configurar el CloudWatch agente mediante la creación o edición manual de un archivo de configuración (Windows)

1. Cree o edite el archivo de configuración del CloudWatch agente como se describe en [Crear o editar manualmente el archivo de configuración del CloudWatch agente](#).
2. Asegúrese de que el archivo se llame C:\ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json y de que contenga el siguiente código:

```

...
"logs": {
 "logs_collected": {
 "files": {
 "collect_list": [
 {
 "file_path": "C:\\ProgramData\\Amazon\\CodeDeploy\\log\\
\\codedeploy-agent-log.txt",
 "log_group_name": "codedeploy-agent-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-log"
 },
 {

```

```
 "file_path": "C:\\ProgramData\\Amazon\\CodeDeploy\\
\\deployment-logs\\codedeploy-agent-deployments.log",
 "log_group_name": "codedeploy-agent-deployment-log",
 "log_stream_name": "{instance_id}-codedeploy-agent-
deployment-log"
 }
]
},
...
}
},
...
```

## Reinicie el CloudWatch agente

Tras realizar los cambios, reinicie el CloudWatch agente tal y como se describe en [Iniciar el CloudWatch agente](#).



## Trabajar con instancias para CodeDeploy

CodeDeploy admite implementaciones en instancias que ejecutan Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) y Windows Server.

Se puede utilizar CodeDeploy para realizar la implementación tanto en instancias de Amazon EC2 como en instancias locales. Una instancia local es cualquier dispositivo físico que no sea una instancia de Amazon EC2 que pueda ejecutar CodeDeploy el agente y conectarse a puntos finales de servicio AWS público. Puede utilizarla CodeDeploy para implementar simultáneamente una aplicación en las instancias de Amazon EC2 en la nube y en los PC de escritorio de su oficina o en los servidores de su propio centro de datos.

## Comparación de las instancias de Amazon EC2 y las instancias en las instalaciones

En la siguiente tabla se comparan las instancias de Amazon EC2 y las instancias en las instalaciones:

| Asunto                                                                                                                                                               | Instancias de Amazon EC2 | Instancias en las instalaciones |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|---------------------------------|
| Requiere que instale y ejecute una versión del CodeDeploy y agente que sea compatible con el sistema operativo que se ejecuta en la instancia.                       | Sí                       | Sí                              |
| Requiere que la instancia se pueda conectar a CodeDeploy.                                                                                                            | Sí                       | Sí                              |
| Es necesario adjuntar un perfil de instancia de IAM a la instancia. El perfil de la instancia de IAM debe tener permisos para participar en CodeDeploy las implement | Sí                       | No                              |

| Asunto                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Instancias de Amazon EC2 | Instancias en las instalaciones |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|---------------------------------|
| <p>aciones. Para obtener más información, consulte <a href="#">Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2</a>.</p>                                                                                                                                                                                                                                                                                                                                                                             |                          |                                 |
| <p>Es necesario realizar uno de los siguientes procedimientos para autenticar y registrar las instancias:</p> <ul style="list-style-type: none"> <li>• Crear un rol de IAM que pueda asumir un usuario de IAM en cada instancia para recuperar las credenciales temporales que se actualizan periódicamente generadas a través de AWS Security Token Service.</li> <li>• Crear un usuario de IAM para cada instancia y almacenar las credenciales de la cuenta de usuario de IAM en texto sin formato en la instancia.</li> </ul> | No                       | Sí                              |
| <p>Requiere que registre cada instancia CodeDeploy antes de poder realizar la implementación en ella.</p>                                                                                                                                                                                                                                                                                                                                                                                                                         | No                       | Sí                              |
| <p>Requiere que etiquete cada instancia antes de CodeDeploy y poder implementarla en ella.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                    | Sí                       | Sí                              |

| Asunto                                                                                                                                                                   | Instancias de Amazon EC2 | Instancias en las instalaciones |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|---------------------------------|
| Puede participar en los escenarios de Auto Scaling y Elastic Load Balancing de Amazon EC2 como parte de las CodeDeploy implementaciones.                                 | Sí                       | No                              |
| Se puede implementar desde depósitos y GitHub repositorios de Amazon S3.                                                                                                 | Sí                       | Sí                              |
| Admiten disparadores que solicitan el envío de notificaciones de correo electrónico o SMS cuando se producen eventos especificados en las implementaciones o instancias. | Sí                       | Sí                              |
| Se pueden facturar para implementaciones asociadas.                                                                                                                      | No                       | Sí                              |

## Tareas de instancia para CodeDeploy

Para lanzar o configurar instancias para su uso en implementaciones, elija entre las siguientes instrucciones:

Quiero lanzar una nueva instancia de Amazon EC2 en Amazon Linux o Windows Server.

Para lanzar la instancia de Amazon EC2 con el mínimo esfuerzo, consulte [Crear una instancia de Amazon EC2 para CodeDeploy \(plantilla\)AWS CloudFormation](#).

Para lanzar la instancia de Amazon EC2 por su cuenta, consulte [Cree una instancia de](#)

|                                                                                                                                                          |                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                          | <a href="#">Amazon EC2 para CodeDeploy (AWS CLI o una consola Amazon EC2)</a> .                                |
| Quiero lanzar una nueva instancia de Amazon EC2 con Ubuntu Server o RHEL.                                                                                | Consulte <a href="#">Cree una instancia de Amazon EC2 para CodeDeploy (AWS CLI o una consola Amazon EC2)</a> . |
| Quiero configurar una instancia de Amazon EC2 en Amazon Linux, Windows Server, Ubuntu Server o RHEL.                                                     | Consulte <a href="#">Configurar una instancia de Amazon EC2 con la que trabajar CodeDeploy</a> .               |
| Quiero configurar una instancia en las instalaciones (dispositivos físicos que no son instancias de Amazon EC2) en Windows Server, Ubuntu Server o RHEL. | Consulte <a href="#">Working with On-Premises Instances</a> .                                                  |
| Quiero CodeDeploy aprovisionar una flota de instancias de reemplazo durante una implementación azul/verde.                                               | Consulte <a href="#">Trabajar con despliegues en CodeDeploy</a> .                                              |

Para preparar instancias de Amazon EC2 en grupos de Amazon EC2 Auto Scaling, debe seguir algunos pasos adicionales. Para obtener más información, consulte [Integración CodeDeploy con Amazon EC2 Auto Scaling](#).

## Temas

- [Tagging Instances for Deployments](#)
- [Working with Amazon EC2 Instances](#)
- [Working with On-Premises Instances](#)
- [View Instance Details](#)
- [Instance Health](#)

# Etiquetado de instancias para grupos de despliegue en CodeDeploy

Para ayudar a administrar las instancias de Amazon EC2 y las instancias en las instalaciones, puede usar etiquetas para asignar sus propios metadatos a cada recurso. Las etiquetas le permiten clasificar las instancias de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. Esto es útil cuando dispone de muchas instancias. Puede identificar rápidamente una instancia o grupo de instancias en función de las etiquetas que haya asignado a ellos. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario. Para obtener más información, consulte [Etiquetado de los recursos de Amazon EC2](#).

Para especificar qué instancias se incluyen en un grupo de CodeDeploy implementación, especifique las etiquetas en uno o más grupos de etiquetas. Las instancias que cumplan sus criterios de etiqueta son las instancias en las que está instalada la última versión de la aplicación en el momento de creación de la implementación de ese grupo de implementaciones.

## Note

También puede incluir grupos de Amazon EC2 Auto Scaling en grupos de implementación, pero estos se identifican por sus nombres y no por las etiquetas que se aplican a las instancias. Para obtener más información, consulte [Integración CodeDeploy con Amazon EC2 Auto Scaling](#).

Los criterios de las instancias en un grupo de implementaciones pueden ser tan sencillos como una única etiqueta en un solo grupo de etiquetas. Puede ser tan complejo como 10 etiquetas cada una en un máximo de tres grupos de etiquetas.

Si utiliza un único grupo de etiquetas, cualquier instancia que se identifique mediante al menos una etiqueta del grupo se incluye en el grupo de implementaciones. Si utiliza varios grupos de etiquetas, solo se incluyen las instancias que se identifican por al menos una etiqueta en cada uno de los grupos de etiquetas.

En los siguientes ejemplos se muestra cómo las etiquetas y los grupos de etiquetas se pueden utilizar para seleccionar las instancias de un grupo de implementaciones.

## Temas

- [Ejemplo 1: un único grupo de etiquetas, una única etiqueta](#)

- [Ejemplo 2: un único grupo de etiquetas, varias etiquetas](#)
- [Ejemplo 3: varios grupos de etiquetas, etiquetas únicas](#)
- [Ejemplo 4: varios grupos de etiquetas, varias etiquetas](#)

## Ejemplo 1: un único grupo de etiquetas, una única etiqueta

Puede especificar una única etiqueta en un único grupo de etiquetas:

Grupo de etiquetas 1

| Clave  | Valor          |
|--------|----------------|
| Nombre | AppVersion-ABC |

Cada instancia que se etiqueta con Name=AppVersion-ABC forma parte del grupo de implementaciones, incluso si tiene otras etiquetas aplicadas.

CodeDeploy vista de configuración de la consola:

Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.  
**One tag group:** Any instance identified by the tag group will be deployed to.  
**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

Tag group 1

| Key - <i>optional</i>                                                                    | Value - <i>optional</i>                                                                            |
|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <input type="text" value="Name"/> <span style="float: right; font-size: 0.8em;">✕</span> | <input type="text" value="AppVersion-ABC"/> <span style="float: right; font-size: 0.8em;">✕</span> |
| <input type="text"/>                                                                     | <input type="text"/>                                                                               |

Estructura JSON:

```
"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
```

```
 "Type": "KEY_AND_VALUE",
 "Key": "Name",
 "Value": "AppVersion-ABC"
 }
]
},
```

## Ejemplo 2: un único grupo de etiquetas, varias etiquetas

También puede especificar varias etiquetas en un único grupo de etiquetas:

### Grupo de etiquetas 1

| Clave  | Valor |
|--------|-------|
| Región | Norte |
| Región | Sur   |
| Región | Este  |

Una instancia que se etiqueta con cualquiera de estas tres etiquetas forma parte del grupo de implementaciones, incluso si tiene otras etiquetas aplicadas. Si, por ejemplo, tenía otras instancias etiquetadas `Region=West`, estas no se incluirían en el grupo de implementaciones.

CodeDeploy vista de configuración de la consola:

Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

## Tag group 1

Key - optional

Value - optional

|                                     |                                    |                                           |
|-------------------------------------|------------------------------------|-------------------------------------------|
| <input type="text" value="Region"/> | <input type="text" value="North"/> |                                           |
| <input type="text" value="Region"/> | <input type="text" value="South"/> | <input type="button" value="Remove tag"/> |
| <input type="text" value="Region"/> | <input type="text" value="East"/>  | <input type="button" value="Remove tag"/> |

## Estructura JSON:

```

"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "North"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "South"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "East"
 }
]
]
},

```



## Ejemplo 3: varios grupos de etiquetas, etiquetas únicas

También puede utilizar varios conjuntos de grupos de etiquetas con un único par de clave-valor en cada uno para especificar los criterios de las instancias de un grupo de implementaciones. Cuando utiliza varios grupos de etiquetas en un grupo de implementaciones, solo las instancias identificadas por todos los grupos de etiquetas se incluyen en el grupo de implementaciones.

### Grupo de etiquetas 1

| Clave  | Valor          |
|--------|----------------|
| Nombre | AppVersion-ABC |

### Grupo de etiquetas 2

| Clave  | Valor |
|--------|-------|
| Región | Norte |

### Grupo de etiquetas 3

| Clave | Valor     |
|-------|-----------|
| Tipo  | t2.medium |

Puede tener instancias en varias regiones y distintos tipos de instancia etiquetados con `Name=AppVersion-ABC`. En este ejemplo, solo las instancias también etiquetadas con `Region=North` y `Type=t2.medium` forman parte del grupo de implementaciones.

CodeDeploy vista de configuración de la consola:

Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

## Tag group 1

Key - *optional*Value - *optional*
   


## Tag group 2

Key - *optional*Value - *optional*
   


## Tag group 3

Key - *optional*Value - *optional*
   


## Estructura JSON:

```
"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Name",
 "Value": "AppVersion-ABC"
 }
]
],

```

```
[
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "North"
 },
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Type",
 "Value": "t2.medium"
 }
],
]
```

## Ejemplo 4: varios grupos de etiquetas, varias etiquetas

Al utilizar varios grupos de etiquetas con varias etiquetas en uno o más grupos, una instancia debe coincidir con al menos una de las etiquetas de cada uno de los grupos.

### Grupo de etiquetas 1

| Clave   | Valor   |
|---------|---------|
| Entorno | Beta    |
| Entorno | Staging |

### Grupo de etiquetas 2

| Clave  | Valor |
|--------|-------|
| Región | Norte |
| Región | Sur   |
| Región | Este  |

## Grupo de etiquetas 3

| Clave | Valor     |
|-------|-----------|
| Tipo  | t2.medium |
| Tipo  | t2.large  |

En este ejemplo, para que se incluya en el grupo de implementaciones, una instancia debe ser etiquetada con (1) `Environment=Beta` o `Environment=Staging`, (2) `Region=North`, `Region=South` o `Region=East`, y (3) `Type=t2.medium` o `Type=t2.large`.

En modo de ilustración, las instancias con los siguientes grupos de etiquetas estarían entre las que se incluyen en el grupo de implementaciones:

- `Environment=Beta, Region=North, Type=t2.medium`
- `Environment=Staging, Region=East, Type=t2.large`
- `Environment=Staging, Region=South, Type=t2.large`

Las instancias con los siguientes grupos de etiquetas no estarían entre las que se incluyen en el grupo de implementaciones. Los valores de clave resaltados hacen que las instancias que se excluyan:

- `Environment=Beta, Region=West, Type=t2.medium`
- `Environment=Staging, Region=East, Type=t2.micro`
- `Environment=Production, Region=South, Type=t2.large`

CodeDeploy vista de configuración de la consola:

Amazon EC2 instances

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

## Tag group 1

Key - *optional*Value - *optional*       

## Tag group 2

Key - *optional*Value - *optional*           

## Tag group 3

Key - *optional*Value - *optional*

## Estructura JSON:

```
"ec2TagSet": {
 "ec2TagSetList": [
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Environment",
 "Value": "Beta"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Environment",
 "Value": "Staging"
 }
],
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "North"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "South"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Region",
 "Value": "East"
 }
],
 [
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Type",
 "Value": "t2.medium"
 },
 {
 "Type": "KEY_AND_VALUE",
 "Key": "Type",
 "Value": "t2.large"
 }
]
]
}
```

```
],
],
},
```

## Trabajar con instancias de Amazon EC2 para CodeDeploy

Una instancia de Amazon EC2 es un entorno informático virtual que se crea y configura mediante Amazon Elastic Compute Cloud. Amazon EC2 proporciona una capacidad informática escalable en la AWS nube. Puede usar Amazon EC2 para lanzar tantos o tan pocos servidores virtuales como necesite para sus CodeDeploy implementaciones.

Para obtener más información acerca de Amazon EC2, consulte la [Guía de introducción de Amazon EC2](#).

Las instrucciones de esta sección muestran cómo crear y configurar instancias de Amazon EC2 para usarlas en sus CodeDeploy implementaciones.

### Temas

- [Cree una instancia de Amazon EC2 para CodeDeploy \(AWS CLI o una consola Amazon EC2\)](#)
- [Crear una instancia de Amazon EC2 para CodeDeploy \(plantilla\)AWS CloudFormation](#)
- [Configurar una instancia de Amazon EC2 con la que trabajar CodeDeploy](#)

## Cree una instancia de Amazon EC2 para CodeDeploy (AWS CLI o una consola Amazon EC2)

Estas instrucciones le muestran cómo lanzar una nueva instancia de Amazon EC2 que esté configurada para su uso en CodeDeploy las implementaciones.

Puede utilizar nuestra AWS CloudFormation plantilla para lanzar una instancia de Amazon EC2 que ejecute Amazon Linux o Windows Server y que ya esté configurada para su uso en CodeDeploy las implementaciones. No proporcionamos una AWS CloudFormation plantilla para las instancias de Amazon EC2 que ejecutan Ubuntu Server o Red Hat Enterprise Linux (RHEL). Si desea conocer otras alternativas al uso de la plantilla, consulte [Trabajar con instancias para CodeDeploy](#).

Puede utilizar la consola Amazon EC2 o las API de Amazon EC2 para lanzar una instancia de Amazon EC2. AWS CLI

## Lanzamiento de una instancia de Amazon EC2 (consola)

### Requisitos previos

Si aún no lo ha hecho, siga las instrucciones [Empezar con CodeDeploy](#) para configurar AWS CLI y crear un perfil de instancia de IAM.

### Lanzamiento de una instancia de Amazon EC2

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. En el panel de navegación, elija Instancias (Instancias) y seleccione Launch Instance (Lanzar instancia).
3. En la página Step 1: Choose an Amazon Machine Image (AMI) (Paso 1: Elegir una imagen de máquina de Amazon [AMI]) de la pestaña Quick Start (Inicio rápido), busque el sistema operativo y la versión que quiera usar y luego elija Select (Seleccionar). Debe elegir un sistema operativo AMI Amazon EC2 compatible con CodeDeploy. Para obtener más información, consulte [Sistemas operativos compatibles con el agente CodeDeploy](#).
4. En la página Paso 2: Elegir un tipo de instancia, elija cualquier tipo de instancia de Amazon EC2 disponible y seleccione Siguiente: Configurar los detalles de la instancia.
5. En la página Paso 3: Configurar los detalles de la instancia, en la lista IAM role (Rol de IAM), elija el rol de instancia que ha creado en [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#). Si ha utilizado el nombre de rol sugerido, elija **CodeDeployDemo-EC2-Instance-Profile**. Si ha creado su propio nombre de rol, selecciónelo.

#### Note

Si una nube privada virtual (VPC) predeterminada no aparece en la lista Red, debe elegir o crear una VPC y una subred de Amazon. Elija Crear nueva VPC o Crear nueva subred o ambas opciones. Para obtener más información, consulte [VPC y subredes](#).

6. Elija Siguiente: Agregar almacenamiento.
7. Deje la página Paso 4: Agregar almacenamiento sin cambiar y elija Siguiente: Añadir etiquetas.
8. En la página Paso 5: Añadir etiquetas, elija Añadir etiqueta.
9. En el cuadro Clave, escriba **Name**. En el cuadro Valor, escriba **CodeDeployDemo**.



**⚠ Important**

El contenido de los cuadros Clave y Valor distingue entre mayúsculas y minúsculas.

10. Elija Siguiente: Configurar un grupo de seguridad.
11. En la página Paso 6: Configurar un grupo de seguridad, deje la opción Crear un grupo de seguridad nuevo seleccionada.

Se configura un rol de SSH predeterminado para instancias de Amazon EC2 que ejecutan Amazon Linux, Ubuntu Server o RHEL. Se configura un rol de RDP predeterminado para instancias de Amazon EC2 que ejecutan Windows Server.

12. Si desea abrir el puerto HTTP, elija el botón Add Rule (Añadir regla) y en la lista desplegable Type (Tipo), elija **HTTP**. Acepte el valor Source (Origen) predeterminado de Custom 0.0.0.0/0 (Personalizado 0.0.0.0/0) y luego elija Review and Launch (Revisar y lanzar).

**ℹ Note**

En un entorno de producción, recomendamos restringir el acceso a los puertos SSH, RDP y HTTP, en lugar de especificar Anywhere 0.0.0.0/0. CodeDeploy no requiere acceso a los puertos sin restricciones ni acceso HTTP. Para obtener más información, consulte [Sugerencias para proteger la instancia de Amazon EC2](#).

Si aparece el cuadro de diálogo Boot from General Purpose (SSD) (Arrancar desde uso general [SSD]), siga las instrucciones y luego seleccione Next (Siguiente).

13. Deje la página Step 7: Review Instance Launch (Paso 7: Revisar el lanzamiento de la instancia) sin cambiar y elija Launch (Lanzar).
14. En el cuadro de diálogo Select an existing key pair or create a new key pair (Seleccionar par de claves existentes o crear nuevo par de claves), elija Choose an existing key pair (Elegir un par de claves existente) o Create a new key pair (Crear un nuevo par de claves). Si ya ha configurado un par de claves de la instancia de Amazon EC2, puede elegirlo aquí.

Si aún no tiene un par de claves de instancia de Amazon EC2, elija Create a new key pair (Crear un nuevo par de claves) y asígnele un nombre fácil de reconocer. Elija Descargar par de claves para descargar el par de claves de la instancia de Amazon EC2 en su equipo.

**⚠ Important**

Debe disponer de un par de claves si desea obtener acceso a su instancia de Amazon EC2 con SSH o RDP.

15. Elija Launch Instances.
16. Elija el ID de su instancia de Amazon EC2. No continúe hasta que la instancia se haya lanzado y haya superado todas las comprobaciones.

### Instale el agente CodeDeploy

El CodeDeploy agente debe estar instalado en la instancia de Amazon EC2 antes de usarlo en CodeDeploy las implementaciones. Para obtener más información, consulte [Instala el CodeDeploy agente](#).

**ℹ Note**

Puede configurar la instalación y las actualizaciones automáticas del CodeDeploy agente al crear el grupo de implementación en la consola.

## Lanzamiento de una instancia de Amazon EC2 (CLI)

### Requisitos previos

Si aún no lo ha hecho, siga las instrucciones [Empezar con CodeDeploy](#) para configurar AWS CLI y crear un perfil de instancia de IAM.

### Lanzamiento de una instancia de Amazon EC2

1. Solo para Windows Server Si va a crear una instancia de Amazon EC2 que ejecuta Windows Server, llame a los comandos `create-security-group` y `authorize-security-group-ingress` para crear un grupo de seguridad que permita el acceso RDP (que no se permite de forma predeterminada) y, en su caso, el acceso HTTP. Por ejemplo, para crear un grupo de seguridad denominado `CodeDeployDemo-Windows-Security-Group`, ejecute los siguientes comandos, de uno en uno:

```
aws ec2 create-security-group --group-name CodeDeployDemo-Windows-Security-Group --description "For launching Windows Server images for use with CodeDeploy"
```

```
aws ec2 authorize-security-group-ingress --group-name CodeDeployDemo-Windows-Security-Group --to-port 3389 --ip-protocol tcp --cidr-ip 0.0.0.0/0 --from-port 3389
```

```
aws ec2 authorize-security-group-ingress --group-name CodeDeployDemo-Windows-Security-Group --to-port 80 --ip-protocol tcp --cidr-ip 0.0.0.0/0 --from-port 80
```

### Note

Para fines de demostración, estos comandos crean un grupo de seguridad que permite acceso ilimitado a RDP a través del puerto 3389 y, en su caso, a HTTP a través del puerto 80. Es recomendable que restrinja el acceso a los puertos HTTP y RDP. CodeDeploy no requiere acceso a los puertos sin restricciones ni acceso HTTP. Para obtener más información, consulte [Sugerencias para proteger la instancia de Amazon EC2](#).


## 2. Llame al comando `run-instances` para crear y lanzar la instancia de Amazon EC2.

Antes de llamar a este comando, debe recopilar lo siguiente:

- El ID de una imagen de máquina de Amazon (AMI) (*id-ami*) que utiliza para la instancia. Para obtener el ID, consulte [Búsqueda de una AMI de Linux](#).
- El nombre del tipo de instancia de Amazon EC2 (*instance-type*) que crea, como `t1.micro`. Para ver una lista, consulte [Tipos de instancias de Amazon EC2](#).
- El nombre de un perfil de instancia de IAM con permiso para acceder al bucket de Amazon S3 donde se almacenan los archivos de instalación del CodeDeploy agente de su región.

Para obtener información sobre la creación de un perfil de instancia de IAM, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#).

- El nombre de un par de claves de instancia de Amazon EC2 (*key-name*) para permitir el acceso SSH a una instancia de Amazon EC2 que ejecute Amazon Linux, Ubuntu Server o el acceso RHEL o RDP a una instancia de Amazon EC2 que ejecute Windows Server.

 Important

Escriba solo el nombre del par de claves, no la extensión de archivo del par de claves. Por ejemplo, my-keypair, no my-keypair.pem.


Para buscar un nombre de par de claves, abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>. En el panel de navegación, en Network & Security (Red y seguridad), elija Key Pairs (Pares de claves) y anote el nombre del par de claves de la lista.

Para generar un par de claves, consulte [Crear un par de claves con Amazon EC2](#). Compruebe que crea el par de claves en una de las regiones que se encuentran en la lista de [Regiones y puntos de enlace](#) de la Referencia general de AWS. De lo contrario, no podrá utilizar el par de claves de instancia de Amazon EC2 con CodeDeploy

Para Amazon Linux, RHEL y Ubuntu Server

Para llamar al comando run-instances para lanzar una instancia de Amazon EC2 que ejecuta Amazon Linux, Ubuntu Server o RHEL y adjuntar el perfil de instancia de IAM que creó en [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#). Por ejemplo:

```
aws ec2 run-instances \
 --image-id ami-id \
 --key-name key-name \
 --count 1 \
 --instance-type instance-type \
 --iam-instance-profile Name=iam-instance-profile
```

 Note

Este comando crea un grupo de seguridad predeterminado para la instancia de Amazon EC2 que permite el acceso a varios puertos, incluido acceso ilimitado a SSH a través del puerto 22 y, en su caso, a HTTP a través del puerto 80. Como práctica recomendada, recomendamos restringir el acceso únicamente a los puertos SSH y HTTP. CodeDeploy no requiere acceso a los puertos sin restricciones ni acceso a los puertos HTTP. Para

obtener más información, consulte [Sugerencias para proteger la instancia de Amazon EC2](#).

## Para Windows Server

Para llamar al comando `run-instances` y lanzar una instancia de Amazon EC2 ejecutando Windows Server, así como adjuntar el perfil de instancia de IAM que ha creado en [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#) y especificar el nombre del grupo de seguridad que ha creado en el paso 1. Por ejemplo:

```
aws ec2 run-instances --image-id ami-id --key-name key-name --count 1 --instance-type instance-type --iam-instance-profile Name=iam-instance-profile --security-groups CodeDeploy-Windows-Security-Group
```

Estos comandos lanzan una sola instancia de Amazon EC2 con la AMI, el par de claves y el tipo de instancia especificados, con el perfil de instancia de IAM indicado, y ejecutan el script especificado durante el lanzamiento.

3. Anote el valor de `InstanceID` en la salida. Si olvida este valor, puede obtenerlo más tarde si llama al comando `describe-instances` con el par de claves de la instancia de Amazon EC2.

```
aws ec2 describe-instances --filters "Name=key-name,Values=keyName" --query "Reservations[*].Instances[*].[InstanceId]" --output text
```

Use el ID de instancia para llamar al `create-tags` comando, que etiqueta la instancia de Amazon EC2 para que CodeDeploy pueda encontrarla más adelante durante una implementación. En el siguiente ejemplo, la etiqueta se denomina **CodeDeployDemo**, pero puede especificar cualquier etiqueta de instancia de Amazon EC2 que desee.

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=CodeDeployDemo
```

Puede aplicar varias etiquetas a una instancia al mismo tiempo. Por ejemplo:

```
aws ec2 create-tags --resources instance-id --tags Key=Name,Value=testInstance Key=Region,Value=West Key=Environment,Value=Beta
```

Para verificar que la instancia de Amazon EC2 se ha lanzado y ha superado todas las comprobaciones, utilice el ID de instancia para llamar al comando `describe-instance-status`.

```
aws ec2 describe-instance-status --instance-ids instance-id --query
"InstanceStatuses[*].InstanceStatus.[Status]" --output text
```

Si la instancia se ha lanzado y ha superado todas las comprobaciones, aparece ok en la salida:

Instale el agente CodeDeploy

El CodeDeploy agente debe estar instalado en la instancia de Amazon EC2 antes de usarlo en CodeDeploy las implementaciones. Para obtener más información, consulte [Instala el CodeDeploy agente](#).

#### Note

Puede configurar la instalación y las actualizaciones automáticas del CodeDeploy agente al crear el grupo de implementación en la consola.

## Crear una instancia de Amazon EC2 para CodeDeploy (plantilla)AWS CloudFormation

Puede utilizar nuestra AWS CloudFormation plantilla para lanzar rápidamente una instancia de Amazon EC2 que ejecute Amazon Linux o Windows Server. Puede usar la AWS CLI CodeDeploy consola o las AWS API para lanzar la instancia con la plantilla. Además de lanzar la instancia, la plantilla hace lo siguiente:

- Indica AWS CloudFormation que dé permiso a la instancia para participar en CodeDeploy las implementaciones.
- Etiqueta la instancia para CodeDeploy poder encontrarla durante una implementación.
- Instala y ejecuta el CodeDeploy agente en la instancia.

No es necesario que utilice nuestro AWS CloudFormation para configurar una instancia de Amazon EC2. Si desea conocer otras alternativas, consulte [Trabajar con instancias para CodeDeploy](#).

No proporcionamos una AWS CloudFormation plantilla para las instancias de Amazon EC2 que ejecutan Ubuntu Server o Red Hat Enterprise Linux (RHEL).

## Temas

- [Antes de empezar](#)
- [Lance una instancia de Amazon EC2 con la AWS CloudFormation plantilla \(consola\)](#)
- [Lance una instancia de Amazon EC2 con la AWS CloudFormation plantilla \(AWS CLI\)](#)

## Antes de empezar

Antes de poder usar la AWS CloudFormation plantilla para lanzar instancias de Amazon EC2, asegúrese de completar los siguientes pasos.

1. Asegúrese de haber creado un usuario administrativo, tal y como se describe en [Paso 1: Configurar](#). Compruebe que el usuario tiene los siguientes permisos mínimos y añada los que no estén presentes:
  - cloudformation:\*
  - codedeploy:\*
  - ec2:\*
  - Soy: AddRoleToInstanceProfile
  - objetivo: CreateInstanceProfile
  - objetivo: CreateRole
  - objetivo: DeleteInstanceProfile
  - objetivo: DeleteRole
  - objetivo: DeleteRolePolicy
  - objetivo: GetRole
  - objetivo: DeleteRolePolicy
  - objetivo: PutRolePolicy
  - objetivo: RemoveRoleFromInstanceProfile
2. Asegúrese de tener un par de claves de instancia para permitir el acceso SSH a la instancia de Amazon EC2 que ejecuta Amazon Linux o el acceso RDP a la instancia que ejecuta Windows Server.

Para buscar un nombre de par de claves, abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>. En el panel de navegación, en Network & Security (Red y seguridad), elija Key Pairs (Pares de claves) y anote el nombre del par de claves de la lista.

Para generar un nuevo par de claves, consulte [Crear un par de claves con Amazon EC2](#). Asegúrese de que crea el par de claves en una de las regiones que se encuentran en la lista de [Regiones y puntos de conexión](#) de la Referencia general de AWS. De lo contrario, no podrá usar el key pair de instancias con CodeDeploy.

## Lance una instancia de Amazon EC2 con la AWS CloudFormation plantilla (consola)

1. Inicie sesión en la AWS CloudFormation consola AWS Management Console y ábrala en <https://console.aws.amazon.com/cloudformation>.

### Important

Inicia sesión en la AWS Management Console con la misma cuenta que utilizaste [Empezar con CodeDeploy](#). En la barra de navegación, en el selector de regiones, elige una de las regiones que aparecen en [Región y los puntos finales](#) en Referencia general de AWS. CodeDeploy solo admite estas regiones.

2. Elija Crear pila.
3. En Elegir una plantilla, seleccione Especificar una URL de plantilla de Amazon S3. En el cuadro, escriba la ubicación de la AWS CloudFormation plantilla para su región y, a continuación, seleccione Siguiente.

| Región                                         | Ubicación de la AWS CloudFormation plantilla                                                                                                                                                                                              |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Región del Este de EE. UU. (Ohio)              | <a href="http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json">http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json</a> |
| Región del Este de EE. UU. (Norte de Virginia) | <a href="http://s3.amazonaws.com/aws-codedeploy-us-east-1/templ">http://s3.amazonaws.com/aws-codedeploy-us-east-1/templ</a>                                                                                                               |




| Región                                           | Ubicación de la AWS CloudFormation plantilla                                                                                                                                             |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Región del Oeste de EE. UU (Norte de California) | <code>ates/latest/CodeDeploy_SampleCF_Template.json</code><br><code>http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Región del Oeste de EE. UU (Oregón)              | <code>http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>                                                               |
| Región de Canadá (centro)                        | <code>http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>                                                         |
| Región de Europa (Irlanda)                       | <code>http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>                                                               |
| Región de Europa (Londres)                       | <code>http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>                                                               |
| Región Europa (París)                            | <code>http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json</code>                                                               |

| Región                              | Ubicación de la AWS CloudFormation plantilla                                                                                         |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Región de Europa (Fráncfort)        | <code>http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>     |
| Región Israel (Tel Aviv)            | <code>http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>     |
| Región de Asia-Pacífico (Hong Kong) | <code>http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>           |
| Asia Pacífico (Tokio)               | <code>http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Región de Asia-Pacífico (Seúl)      | <code>http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Región de Asia-Pacífico (Singapur)  | <code>http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |

| Región                                | Ubicación de la AWS CloudFormation plantilla                                                                                          |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Región de Asia-Pacífico (Sídney)      | <code>http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>  |
| Región de Asia-Pacífico (Melbourne)   | <code>https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Región de Asia-Pacífico (Bombay)      | <code>http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>          |
| Región de América del Sur (São Paulo) | <code>aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code>              |


4. En el cuadro Stack name (Nombre de la pila), escriba un nombre para la pila (por ejemplo, **CodeDeployDemoStack**).
5. En Parameters (Parámetros), escriba lo siguiente y seleccione Next (Siguiente).
  - Para InstanceCount, escriba el número de instancias que desea lanzar. (Le recomendamos que deje el valor predeterminado, 1).
  - Para InstanceType, escriba el tipo de instancia que desee lanzar (o deje t1.micro como valor predeterminado).
  - Para KeyPairName, escriba el nombre del key pair de la instancia. Escriba solo el nombre del par de claves, no la extensión de archivo del par de claves.

- En OperatingSystemBox, escriba **Windows** para lanzar instancias que ejecuten Windows Server (o deje Linux como predeterminado).
- En SSHLocation, escriba el rango de direcciones IP que desea utilizar para conectarse a la instancia con SSH o RDP (o deje el valor predeterminado, 0.0.0.0/0).

 Important

El valor predeterminado de **0.0.0.0/0** se proporciona únicamente con fines de demostración. CodeDeploy no requiere que las instancias de Amazon EC2 tengan acceso ilimitado a los puertos. Es recomendable que restrinja el acceso a los puertos SSH (y HTTP). Para obtener más información, consulte [Sugerencias para proteger la instancia de Amazon EC2](#).

- Para TagKey, escriba la etiqueta de instancia que la clave CodeDeploy utilizará para identificar las instancias durante la implementación (o deje el nombre predeterminado).
  - Para TagValue, escriba el valor de la etiqueta de instancia que CodeDeploy se utilizará para identificar las instancias durante la implementación (o deje el valor predeterminado CodeDeployDemo).
6. En la página Options (Opciones), deje las casillas de opción vacías y elija Next (Siguiente).

 Important

AWS CloudFormation las etiquetas son diferentes de las CodeDeploy etiquetas. AWS CloudFormation usa etiquetas para simplificar la administración de su infraestructura. CodeDeploy utiliza etiquetas para identificar las instancias de Amazon EC2. Especifiqué las CodeDeploy etiquetas en la página Especificar parámetros.

7. En la página de revisión, en Capacidades, seleccione la casilla Reconozco que AWS CloudFormation podría crear recursos de IAM y, a continuación, elija Crear.

Una vez AWS CloudFormation creada la pila y lanzado las instancias de Amazon EC2, en la AWS CloudFormation consola, se mostrará CREATE\_COMPLETE en la columna Estado. Este proceso puede tardar varios minutos.

Para comprobar que el CodeDeploy agente se está ejecutando en las instancias de Amazon EC2, consulte y, a continuación [Gestión de las operaciones CodeDeploy de los agentes](#), continúe con [Cree una aplicación con CodeDeploy](#)

## Lance una instancia de Amazon EC2 con la AWS CloudFormation plantilla ( )AWS CLI

1. Utilice nuestra AWS CloudFormation plantilla en una llamada al create-stack comando. Esta pila lanzará una nueva instancia de Amazon EC2 con el CodeDeploy agente instalado.

Para lanzar una instancia de Amazon EC2 que ejecute Amazon Linux:

```
aws cloudformation create-stack \
 --stack-name CodeDeployDemoStack \
 --template-url templateURL \
 --parameters ParameterKey=InstanceCount,ParameterValue=1
 ParameterKey=InstanceType,ParameterValue=t1.micro \
 ParameterKey=KeyName,ParameterValue=keyName \
 ParameterKey=OperatingSystem,ParameterValue=Linux \
 ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0
 ParameterKey=TagKey,ParameterValue=Name \
 ParameterKey=TagValue,ParameterValue=CodeDeployDemo \
 --capabilities CAPABILITY_IAM
```

Para lanzar una instancia de Amazon EC2 que ejecute Windows Server:

```
aws cloudformation create-stack --stack-name CodeDeployDemoStack --template-
url template-url --parameters ParameterKey=InstanceCount,ParameterValue=1
 ParameterKey=InstanceType,ParameterValue=t1.micro
 ParameterKey=KeyName,ParameterValue=keyName
 ParameterKey=OperatingSystem,ParameterValue=Windows
 ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0
 ParameterKey=TagKey,ParameterValue=Name
 ParameterKey=TagValue,ParameterValue=CodeDeployDemo --capabilities CAPABILITY_IAM
```

*keyName* es el nombre del par de claves de la instancia. Escriba solo el nombre del par de claves, no la extensión de archivo del par de claves.

*template-url* es la ubicación de la AWS CloudFormation plantilla en su región:

| Región                                           | Ubicación de la plantilla AWS CloudFormation                                                                                     |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Región del Este de EE. UU. (Ohio)                | <code>http://s3-us-east-2.amazonaws.com/aws-codedeploy-us-east-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |
| Región del Este de EE. UU (Norte de Virginia)    | <code>http://s3.amazonaws.com/aws-codedeploy-us-east-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>                 |
| Región del Oeste de EE. UU (Norte de California) | <code>http://s3-us-west-1.amazonaws.com/aws-codedeploy-us-west-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |
| Región del Oeste de EE. UU (Oregón)              | <code>http://s3-us-west-2.amazonaws.com/aws-codedeploy-us-west-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |
| Región de Canadá (centro)                        | <code>http://s3-ca-central-1.amazonaws.com/aws-codedeploy-ca-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Región de Europa (Irlanda)                       | <code>http://s3-eu-west-1.amazonaws.com/aws-codedeploy-eu-west-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>       |

| Región                              | Ubicación de la plantilla AWS CloudFormation                                                                                         |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Región de Europa (Londres)          | <code>http://s3-eu-west-2.amazonaws.com/aws-codedeploy-eu-west-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>           |
| Región Europa (París)               | <code>http://s3-eu-west-3.amazonaws.com/aws-codedeploy-eu-west-3/templates/latest/CodeDeploy_SampleCF_Template.json</code>           |
| Región de Europa (Fráncfort)        | <code>http://s3-eu-central-1.amazonaws.com/aws-codedeploy-eu-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>     |
| Región Israel (Tel Aviv)            | <code>http://s3-il-central-1.amazonaws.com/aws-codedeploy-il-central-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>     |
| Región de Asia-Pacífico (Hong Kong) | <code>http://s3-ap-east-1.amazonaws.com/aws-codedeploy-ap-east-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>           |
| Asia Pacífico (Tokio)               | <code>http://s3-ap-northeast-1.amazonaws.com/aws-codedeploy-ap-northeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code> |

| Región                                | Ubicación de la plantilla AWS CloudFormation                                                                                          |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Región de Asia-Pacífico (Seúl)        | <code>http://s3-ap-northeast-2.amazonaws.com/aws-codedeploy-ap-northeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>  |
| Región de Asia-Pacífico (Singapur)    | <code>http://s3-ap-southeast-1.amazonaws.com/aws-codedeploy-ap-southeast-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>  |
| Región de Asia-Pacífico (Sídney)      | <code>http://s3-ap-southeast-2.amazonaws.com/aws-codedeploy-ap-southeast-2/templates/latest/CodeDeploy_SampleCF_Template.json</code>  |
| Región de Asia-Pacífico (Melbourne)   | <code>https://aws-codedeploy-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code> |
| Región de Asia-Pacífico (Bombay)      | <code>http://s3-ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/templates/latest/CodeDeploy_SampleCF_Template.json</code>          |
| Región de América del Sur (São Paulo) | <code>aws-codedeploy-ap-northeast-1.s3.sa-east-1.amazonaws.com/templates/latest/CodeDeploy_SampleCF_Template.json</code>              |



Este comando crea una AWS CloudFormation pila denominada **CodeDeployDemoStack**, utilizando la AWS CloudFormation plantilla del bucket de Amazon S3 especificado. La instancia de Amazon EC2 se basa en el tipo de instancia t1.micro, pero puede utilizar cualquier tipo. Está etiquetada con el valor **CodeDeployDemo**, pero se puede etiquetar con cualquier valor. Se le ha aplicado el par de claves de instancia especificado.

2. Llama al describe-stacks comando para comprobar que la AWS CloudFormation pila nombrada **CodeDeployDemoStack** se creó correctamente:

```
aws cloudformation describe-stacks --stack-name CodeDeployDemoStack --query "Stacks[0].StackStatus" --output text
```

No continúe hasta que se devuelva el valor CREATE\_COMPLETE.

Para comprobar que el CodeDeploy agente se está ejecutando en la instancia Amazon EC2, consulte y, a continuación [Gestión de las operaciones CodeDeploy de los agentes](#), continúe con. [Cree una aplicación con CodeDeploy](#)

## Configurar una instancia de Amazon EC2 con la que trabajar CodeDeploy

Estas instrucciones muestran cómo configurar una instancia de Amazon EC2 que ejecute Amazon Linux, Ubuntu Server, Red Hat Enterprise Linux (RHEL) o Windows Server para utilizarla en las implementaciones. CodeDeploy

### Note

Si no tiene una instancia de Amazon EC2, puede utilizar la AWS CloudFormation plantilla para lanzar una que ejecute Amazon Linux o Windows Server. No proporcionamos una plantilla para Ubuntu Server o RHEL.

## Paso 1: Verificar que se haya asociado un perfil de instancia de IAM a la instancia de Amazon EC2

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. En el panel de navegación, bajo Instances, elija Instances.

3. Busque su instancia de Amazon EC2 en la lista y selecciónela.
4. En el panel de detalles de la pestaña Description, tome nota del valor del campo Rol de IAM y, a continuación, vaya a la siguiente sección.

Si el campo está vacío, puede adjuntar a la instancia un perfil de instancia de IAM. Para obtener más información, consulte [Asociar un rol de IAM a una instancia](#).

## Paso 2: Verificar que el perfil de instancia de IAM asociado tiene los permisos de acceso correctos

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Seleccione Roles en el panel de navegación.
3. Busque y elija el nombre de rol de IAM que anotó en el paso 4 de la sección anterior.

### Note

Si desea utilizar el rol de servicio generado por la AWS CloudFormation plantilla en lugar del que creó siguiendo las instrucciones que se indican en [Paso 2: Crear un rol de servicio para CodeDeploy](#), tenga en cuenta lo siguiente:

En algunas versiones de nuestra AWS CloudFormation plantilla, el nombre mostrado del perfil de instancia de IAM generado y adjunto a las instancias de Amazon EC2 no es el mismo que el nombre mostrado en la consola de IAM. Por ejemplo, el perfil de instancia de IAM podría tener un nombre de visualización de CodeDeploySampleStack-expny16-InstanceRoleInstanceProfile-IK8J8A9123EX, mientras que el perfil de la instancia de IAM en la consola de IAM podría tener un nombre de visualización de CodeDeploySampleStack-expny16-InstanceRole-C5P33V1L64EX.

Para ayudarle a identificar el perfil de instancia en la consola de IAM, verá que el prefijo de CodeDeploySampleStack-expny16-InstanceRole es el mismo para ambos. Para obtener información acerca de por qué estos nombres de visualización podrían ser diferentes, consulte [Perfiles de instancia](#).

4. Seleccione la pestaña Relaciones de confianza. Si no hay ninguna entrada en Entidades de confianza que diga The identity provider(s) ec2.amazonaws.com, no puede utilizar esta instancia de Amazon EC2. Deténgase y cree una instancia de Amazon EC2 con la información de [Trabajar con instancias para CodeDeploy](#).

Si hay una entrada que dice Los proveedores de identidad ec2.amazonaws.com y está almacenando sus aplicaciones únicamente en GitHub repositorios, pase a [Paso 3: Etiquetar la instancia de Amazon EC2](#)

Si hay una entrada que dice The identity provider(s) ec2.amazonaws.com y almacenará sus aplicaciones en buckets de Amazon S3, elija la pestaña Permisos.

5. Si hay una política en el área Políticas de permisos, amplíe la política y, a continuación, elija Editar política.
6. Seleccione la pestaña JSON. Si está almacenando sus aplicaciones en buckets de Amazon S3, asegúrese de que "s3:Get\*" y "s3:List\*" estén en la lista de acciones especificadas.

Puede tener un aspecto similar al siguiente:

```
{"Statement":[{"Resource":"*","Action":["... Some actions may already be listed here ...","s3:Get*","s3:List*","... Some more actions may already be listed here ..."],"Effect":"Allow"}]}
```

O puede tener un aspecto similar al siguiente:

```
{ "Version": "2012-10-17", "Statement": [{ "Action": ["... Some actions may already be listed here ...", "s3:Get*", "s3:List*", "... Some more actions may already be listed here ..."], ... }]}
```

Si ni "s3:Get\*" ni "s3:List\*" figuran en la lista de acciones especificadas, elija Edit (Editar) para añadirlas y, a continuación, elija Save (Guardar). (Si ni "s3:Get\*" ni "s3:List\*" es la

última acción de la lista, asegúrese de añadir una coma después de la acción, por lo tanto se valida el documento de la política).

### Note

Le recomendamos que restrinja esta política únicamente a los buckets de Amazon S3 a los que las instancias de Amazon EC2 deban tener acceso. Asegúrese de dar acceso a los buckets de Amazon S3 que contienen el CodeDeploy agente. De lo contrario, podría producirse un error cuando el CodeDeploy agente esté instalado o actualizado en las instancias. Para conceder al perfil de instancia de IAM acceso únicamente a algunos buckets del kit de CodeDeploy recursos de Amazon S3, utilice la siguiente política, pero elimine las líneas de los buckets a los que desee impedir el acceso:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Resource": [
 "arn:aws:s3:::replace-with-your-s3-bucket-name/*",
 "arn:aws:s3:::aws-codedeploy-us-east-2/*",
 "arn:aws:s3:::aws-codedeploy-us-east-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-2/*",
 "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
 "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-central-2/*",
 "arn:aws:s3:::aws-codedeploy-eu-north-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-south-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-south-2/*",
 "arn:aws:s3:::aws-codedeploy-il-central-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-east-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
 "arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
 "arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
```

```
"arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
"arn:aws:s3:::aws-codedeploy-ap-south-1/*",
"arn:aws:s3:::aws-codedeploy-ap-south-2/*",
"arn:aws:s3:::aws-codedeploy-me-central-1/*",
"arn:aws:s3:::aws-codedeploy-me-south-1/*",
"arn:aws:s3:::aws-codedeploy-sa-east-1/*"
]
}
]
}
```

### Paso 3: Etiquetar la instancia de Amazon EC2

Para obtener instrucciones sobre cómo etiquetar la instancia de Amazon EC2 para que CodeDeploy pueda encontrarla durante una implementación, consulte [Trabajar con etiquetas en la consola](#) y, a continuación, vuelva a esta página.

#### Note

Puede etiquetar la instancia de Amazon EC2 con la clave y el valor que desee. Pero asegúrese de especificar esta clave y valor cuando realice una implementación en ella.

### Paso 4: Instalar el AWS CodeDeploy agente en la instancia de Amazon EC2

Para obtener instrucciones sobre cómo instalar el CodeDeploy agente en la instancia Amazon EC2 y comprobar que se está ejecutando, consulte y, a continuación, [Gestión de las operaciones CodeDeploy de los agentes](#), continúe con. [Cree una aplicación con CodeDeploy](#)

## Trabajar con instancias locales para CodeDeploy

Una instancia local es cualquier dispositivo físico que no sea una instancia de Amazon EC2 que pueda ejecutar CodeDeploy el agente y conectarse a puntos finales de servicio AWS público.

La implementación de una revisión de CodeDeploy la aplicación en una instancia local implica dos pasos principales:

- Paso 1: Configure cada instancia local, regístrela y, a continuación CodeDeploy, etiquétela.
- Paso 2: Implementar revisiones de aplicaciones en la instancia en las instalaciones

#### Note

Para experimentar con la creación e implementación de una revisión de la aplicación de muestra en una instancia on-premises correctamente configurada y registrada, consulte [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#). Para obtener información sobre las instancias locales y cómo funcionan con ellas CodeDeploy, consulte. [Working with On-Premises Instances](#)

Si ya no desea usar una instancia en las instalaciones en las implementaciones, simplemente puede eliminar las etiquetas de la instancia en las instalaciones de los grupos de implementación. Para utilizar un método más eficaz, elimine las etiquetas de la instancia en las instalaciones de la instancia. También puede anular el registro de una instancia on-premises de forma explícita para que ya no se pueda utilizar en ninguna implementación. Para obtener más información, consulte [Administrar las operaciones de instancias locales en CodeDeploy](#).

Las instrucciones de esta sección muestran cómo configurar una instancia local y, a continuación, registrarla y etiquetarla CodeDeploy para que pueda usarse en las implementaciones. En esta sección también se describe cómo CodeDeploy obtener información sobre las instancias locales y anular el registro de una instancia local cuando ya no se planea realizar la implementación en ella.

#### Temas

- [Requisitos previos para configurar una instancia en las instalaciones](#)
- [Registre una instancia local con CodeDeploy](#)
- [Administrar las operaciones de instancias locales en CodeDeploy](#)

## Requisitos previos para configurar una instancia en las instalaciones

Se deben cumplir los siguientes requisitos previos para poder registrar una instancia en las instalaciones.

**⚠ Important**

Si utilizas el [register-on-premises-instance](#) comando y actualizas periódicamente las credenciales temporales generadas con AWS Security Token Service (AWS STS), existen otros requisitos previos. Para obtener más información, consulte [Requisitos previos del registro de ARN de sesión de IAM](#).

## Requisitos de los dispositivos

El dispositivo que deseas preparar, registrar y etiquetar como instancia local CodeDeploy debe ejecutar un sistema operativo compatible. Para ver una lista, consulte [Sistemas operativos compatibles con el agente CodeDeploy](#).

Si su sistema operativo no es compatible, el CodeDeploy agente está disponible como código abierto para que lo adapte a sus necesidades. Para obtener más información, consulte el repositorio de [CodeDeploy agentes](#) en GitHub.

## Comunicación saliente

La instancia local debe poder conectarse a los puntos finales del AWS servicio público para poder comunicarse con ellos. CodeDeploy

El CodeDeploy agente se comunica de forma saliente mediante HTTPS a través del puerto 443.

## Control administrativo

La cuenta local o de red utilizada en la instancia en las instalaciones para configurar dicha instancia debe poder ejecutarse como sudo o root (para Ubuntu Server) o como administrador (para Windows Server).

## Permisos de IAM

La identidad de IAM que utiliza para registrar la instancia en las instalaciones debe tener permisos para completar el registro (y para anular el registro de la instancia en las instalaciones, según sea necesario).

Además de la política descrita en [Paso 3: Limitar los permisos del CodeDeploy usuario](#), asegúrese de que la identidad de IAM que efectúa la llamada tiene asociada la siguiente política adicional.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:CreateAccessKey",
 "iam:CreateUser",
 "iam>DeleteAccessKey",
 "iam>DeleteUser",
 "iam>DeleteUserPolicy",
 "iam:ListAccessKeys",
 "iam:ListUserPolicies",
 "iam:PutUserPolicy",
 "iam:GetUser"
],
 "Resource": "*"
 }
]
```

Para obtener información acerca de cómo adjuntar directivas de IAM, consulte [Administración de directivas de IAM](#).

## Registre una instancia local con CodeDeploy

Para registrar una instancia on-premise, debe utilizar una identidad de IAM para autenticar las solicitudes. Puede elegir entre las siguientes opciones para la identidad de IAM y el método de registro que utilice:

- Utilice un ARN de rol de IAM para autenticar las solicitudes.
  - Use el [register-on-premises-instance](#) comando y las credenciales temporales que se actualizan periódicamente generadas con AWS Security Token Service (AWS STS) para configurar manualmente la mayoría de las opciones de registro. Esta opción ofrece el nivel más alto de seguridad, ya que la autenticación se realiza mediante un token temporal que se agota y se debe actualizar periódicamente. Esta opción se recomienda para implementaciones de producción de cualquier tamaño. Para obtener más información, consulte [Usa el register-on-premises-instance comando \(ARN de sesión de IAM\) para registrar una instancia local](#).
- (No recomendado) Utilice un ARN de usuario de IAM para autenticar las solicitudes.
  - Utilice el comando [register](#) para el proceso de registro más automatizado. Esta opción solo debe usarse para implementaciones que no sean de producción en las que la seguridad



sea un problema menor. Esta opción es menos segura porque utiliza credenciales estáticas (permanentes) para la autenticación. Esta opción funciona bien para registrar una instancia en las instalaciones individual. Para obtener más información, consulte [Uso del comando “register” \(ARN de usuario de IAM\) para registrar una instancia en las instalaciones](#).

- Utilice el [register-on-premises-instance](#) comando para configurar manualmente la mayoría de las opciones de registro. Es adecuado para registrar un pequeño número de instancias en las instalaciones. Para obtener más información, consulte [Usa el register-on-premises-instance comando \(ARN del usuario de IAM\) para registrar una instancia local](#).

## Temas

- [Usa el register-on-premises-instance comando \(ARN de sesión de IAM\) para registrar una instancia local](#)
- [Uso del comando “register” \(ARN de usuario de IAM\) para registrar una instancia en las instalaciones](#)
- [Usa el register-on-premises-instance comando \(ARN del usuario de IAM\) para registrar una instancia local](#)

## Usa el register-on-premises-instance comando (ARN de sesión de IAM) para registrar una instancia local

Para tener el máximo control sobre la autenticación y el registro de las instancias locales, puede usar el [register-on-premises-instance](#) comando y actualizar periódicamente las credenciales temporales generadas con (). AWS Security Token Service AWS STS Una función de IAM estática para la instancia asume la función de estas AWS STS credenciales actualizadas para realizar las operaciones de implementación. CodeDeploy

Este método es muy útil cuando necesita registrar un gran número de instancias. Permite automatizar el proceso de registro con. CodeDeploy Puede usar su propio sistema de identidad y autenticación para autenticar las instancias locales y distribuir las credenciales de sesión de IAM desde el servicio a las instancias para su uso. CodeDeploy

### Note

Como alternativa, puedes usar un usuario de IAM compartido distribuido entre todas las instancias locales para llamar a la AWS STS [AssumeRole](#) API y recuperar las credenciales

de sesión de las instancias locales. Este método es menos seguro y no se recomienda para entornos de producción o críticos.

Usa la información de los siguientes temas para configurar una instancia local con las credenciales de seguridad temporales generadas con. AWS STS

## Temas

- [Requisitos previos del registro de ARN de sesión de IAM](#)
- [Paso 1: Crear el rol de IAM que asumirán las instancias en las instalaciones](#)
- [Paso 2: Genera credenciales temporales para una instancia individual mediante AWS STS](#)
- [Paso 3: Añadir un archivo de configuración a la instancia en las instalaciones](#)
- [Paso 4: Preparar una instancia local para las implementaciones CodeDeploy](#)
- [Paso 5: Registre la instancia local con CodeDeploy](#)
- [Paso 6: Etiquetar la instancia en las instalaciones](#)
- [Paso 7: Implementar revisiones de aplicaciones en la instancia on-premises](#)
- [Paso 8: Hacer un seguimiento de las implementaciones en la instancia en las instalaciones](#)

## Requisitos previos del registro de ARN de sesión de IAM

Además de los requisitos previos indicados en [Requisitos previos para configurar una instancia en las instalaciones](#), deben cumplirse los siguientes requisitos adicionales:

## Permisos de IAM

La identidad de IAM que utilices para registrar una instancia local debe tener permisos para realizar operaciones. CodeDeploy asegúrese de que la política `AWSCodeDeployFullAccess` administrada esté asociada a la identidad de IAM. Para obtener más información, consulte [Políticas administradas de AWS](#) en la Guía del usuario de IAM.

## Sistema para actualizar las credenciales temporales

Si utiliza un ARN de sesión de IAM para registrar instancias on-premises, debe disponer de un sistema para actualizar periódicamente las credenciales temporales. Las credenciales temporales caducan después de una hora o antes si se especifica un periodo más corto cuando se generan las credenciales. Existen dos métodos para actualizar las credenciales:

- Método 1: utilice el sistema de identidad y autenticación de su red corporativa con un script CRON que sondee periódicamente el sistema de identidad y autenticación y copie las credenciales de la última sesión en la instancia. Esto le permite integrar su estructura de autenticación e identidad AWS sin necesidad de realizar cambios en el CodeDeploy agente o el servicio para admitir los tipos de autenticación que utiliza en su organización.
- Método 2: ejecuta periódicamente un trabajo CRON en la instancia para activar la AWS STS [AssumeRole](#) acción y escribir las credenciales de la sesión en un archivo al que pueda acceder el CodeDeploy agente. Este método requiere igualmente utilizar un usuario de IAM y copiar las credenciales en la instancia on-premises, pero puede volver a utilizar el mismo usuario de IAM y las mismas credenciales en toda la flota de instancias on-premises.

#### Note

Independientemente de si utilizas el método 1 o el 2, debes configurar un proceso para reiniciar el CodeDeploy agente una vez que se hayan actualizado las credenciales de sesión temporales, de modo que las nuevas credenciales surtan efecto.

Para obtener información sobre cómo crear AWS STS credenciales y trabajar con ellas, consulte [Referencia de AWS Security Token Service API](#) y [Uso de credenciales de seguridad temporales para solicitar acceso a AWS los recursos](#).

Paso 1: Crear el rol de IAM que asumirán las instancias en las instalaciones

Puede usar la consola de IAM AWS CLI o la consola de IAM para crear un rol de IAM que las instancias locales utilizarán para autenticarse e interactuar con él. CodeDeploy

Solo tiene que crear un único rol de IAM. Cada una de las instancias on-premises puede asumir este rol para recuperar las credenciales de seguridad temporales que proporcionan los permisos concedidos a este rol.

El rol que cree necesitará los siguientes permisos para acceder a los archivos necesarios para instalar el agente: CodeDeploy

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
```

```
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "*"
 }
]
```

Le recomendamos que restrinja esta política únicamente a los buckets de Amazon S3 a los que su instancia en las instalaciones necesite tener acceso. Si restringe esta política, asegúrese de dar acceso a los buckets de Amazon S3 que contienen el CodeDeploy agente. De lo contrario, podría producirse un error cada vez que el CodeDeploy agente se instale o actualice en la instancia local. Para obtener más información acerca del control del acceso a los recursos de Amazon S3, consulte [Administración de permisos de acceso para los recursos de Amazon S3](#).

## Cómo crear el rol de IAM

1. Llame al comando [create-role](#) usando la opción `--role-name` para especificar un nombre para el rol de IAM (por ejemplo, `CodeDeployInstanceRole`) y la opción `--assume-role-policy-document` para proporcionar los permisos.

Cuando cree el rol de IAM para esta instancia, puede asignarle el nombre `CodeDeployInstanceRole` e incluir los permisos necesarios en un archivo denominado `CodeDeployRolePolicy.json`:

```
aws iam create-role --role-name CodeDeployInstanceRole --assume-role-policy-document file://CodeDeployRolePolicy.json
```

2. En la salida de la llamada al comando `create-role`, anote el valor del campo ARN. Por ejemplo:

```
arn:aws:iam::123456789012:role/CodeDeployInstanceRole
```

Necesitará el rol ARN cuando utilice la AWS STS [AssumeRole](#) API para generar credenciales a corto plazo para cada instancia.

Para obtener más información sobre la creación de funciones de IAM, consulte [Crear una función para delegar permisos a un AWS servicio](#) en la Guía del usuario de IAM.

[Para obtener información sobre la asignación de permisos a un rol existente, consulte `put-role-policy` la Referencia de comandos.AWS CLI](#)

Paso 2: Genera credenciales temporales para una instancia individual mediante AWS STS

Antes de generar las credenciales temporales que se utilizarán para registrar una instancia on-premises, debe crear o elegir la identidad de IAM (usuario o rol) para la que se generarán las credenciales temporales. El permiso `sts:AssumeRole` debe estar incluido en la configuración de política de esta identidad de IAM.

Para obtener información sobre la concesión de `sts:AssumeRole` permisos a una identidad de IAM, consulte [Crear un rol para delegar permisos a un AWS servicio](#) y [AssumeRole](#).

Hay dos formas de generar credenciales temporales:

- Use el comando [assume-role](#) con la AWS CLI. Por ejemplo:

```
aws sts assume-role --role-arn arn:aws:iam::12345ACCOUNT:role/role-arn --role-session-name session-name
```

Donde:

- *12345ACCOUNT* es el número de cuenta de 12 dígitos de su organización.
- *role-arn* es el ARN del rol que se va a asumir, que generó en [Paso 1: Crear el rol de IAM que asumirán las instancias en las instalaciones](#).
- *session-name* es el nombre que desea asignar a la sesión del rol que va a crear ahora.

#### Note


Si utilizas un script CRON que sondea periódicamente el sistema de identidad y autenticación y copia las credenciales de sesión más recientes en la instancia (el método 1 para actualizar las credenciales temporales se describe en [Requisitos previos del registro de ARN de sesión de IAM](#)), puedes utilizar cualquier AWS SDK compatible para realizar la llamada. [AssumeRole](#)

- Usa una herramienta proporcionada por AWS.

La `aws-codedeploy-session-helper` herramienta genera AWS STS credenciales y las escribe en un archivo que coloques en la instancia. Esta herramienta es adecuada para el método 2 de

actualización de credenciales temporales que se describe en [Requisitos previos del registro de ARN de sesión de IAM](#). En este método, la `aws-codedeploy-session-helper` herramienta se coloca en cada instancia y ejecuta el comando con los permisos de un usuario de IAM. Cada instancia utiliza las mismas credenciales de usuario de IAM junto con esta herramienta.

Para obtener más información, consulta el [aws-codedeploy-session-helper](#) GitHub repositorio.

 Note

Una vez que haya creado las credenciales de inicio de sesión de IAM, colóquelas en cualquier ubicación de la instancia on-premises. En el siguiente paso, configurará el CodeDeploy agente para que acceda a las credenciales de esta ubicación.

Antes de continuar, asegúrese de que el sistema que se utilizará para actualizar periódicamente las credenciales temporales está instalado. Si las credenciales temporales no se actualizan, las implementaciones en la instancia on-premises producirán un error. Para obtener más información, consulte "Sistema para actualizar las credenciales temporales" en [Requisitos previos del registro de ARN de sesión de IAM](#).

### Paso 3: Añadir un archivo de configuración a la instancia en las instalaciones

Añada un archivo de configuración a la instancia on-premises, utilizando los permisos raíz o de administrador. Este archivo de configuración se utiliza para declarar las credenciales de IAM y la AWS región de destino para CodeDeploy la que se van a utilizar. El archivo debe añadirse a una ubicación específica de la instancia on-premises. El archivo debe incluir el ARN de la sesión temporal de IAM, su ID de clave secreta y su clave de acceso secreta, y la AWS región de destino.

Para añadir un archivo de configuración

1. Cree un archivo denominado `codedeploy.onpremises.yml` (si se trata de una instancia en las instalaciones de Ubuntu Server o RHEL) o `conf.onpremises.yml` (en el caso de una instancia en las instalaciones de Windows) en la siguiente ubicación de la instancia en las instalaciones:
  - Para Ubuntu Server: `/etc/codedeploy-agent/conf`
  - Para Windows Server: `C:\ProgramData\Amazon\CodeDeploy`
2. Utilice un editor de texto para añadir la siguiente información al archivo `codedeploy.onpremises.yml` (Linux) o `conf.onpremises.yml` (Windows) recién creado:

```

iam_session_arn: iam-session-arn
aws_credentials_file: credentials-file
region: supported-region
```

Donde:


- *iam-session-arn* es el ARN de sesión de IAM en el que anotó. [Paso 2: Genera credenciales temporales para una instancia individual mediante AWS STS](#)
- *credentials-file* es la ubicación del archivo de credenciales del ARN de sesión temporal anotado en [Paso 2: Genera credenciales temporales para una instancia individual mediante AWS STS](#).
- *la región CodeDeploy compatible es una de las regiones compatibles, tal y como se indica en Región y puntos finales en. Referencia general de AWS*

Paso 4: Preparar una instancia local para las implementaciones CodeDeploy

Instale y configure el AWS CLI


Instale y configure AWS CLI la instancia local. (Se AWS CLI usará para descargar e instalar el CodeDeploy agente en la instancia local).

1. Para instalarlo AWS CLI en la instancia local, siga las instrucciones que se indican en la [Guía del AWS Command Line Interface usuario para realizar la AWS CLI configuración con](#) el.

 Note

CodeDeploy los comandos para trabajar con instancias locales estuvieron disponibles en la versión 1.7.19 de. AWS CLI Si AWS CLI ya tienes una versión de la instalada, puedes comprobar su versión llamando a. `aws --version`

2. Para configurar AWS CLI la instancia local, siga las instrucciones de la Guía del AWS Command Line Interface usuario sobre [cómo configurar AWS CLI la](#) instancia local.

 Important

Al configurar la AWS CLI (por ejemplo, mediante una llamada al `aws configure` comando), asegúrese de especificar el ID de clave secreta y la clave de acceso secreta de un usuario

de IAM que tenga, como mínimo, los permisos descritos en. [Requisitos previos del registro de ARN de sesión de IAM](#)

Establecer la variable de entorno AWS\_REGION (solo Ubuntu Server y RHEL)

Si no está ejecutando Ubuntu Server o RHEL en su instancia local, omita este paso y vaya directamente a «Instalar el CodeDeploy agente».

Instala el CodeDeploy agente en una instancia local de Ubuntu Server o RHEL y habilita la instancia para que actualice el CodeDeploy agente cada vez que haya una nueva versión disponible. Para ello, defina la variable de entorno de la instancia en el identificador de una de las regiones compatibles. CodeDeploy Le recomendamos que establezca el valor en la región en la que se encuentran CodeDeploy las aplicaciones, los grupos de implementación y las revisiones de las aplicaciones (por ejemplo, us-west-2). Para obtener una lista de regiones, consulte [Regiones y puntos de conexión](#) en la Referencia general de AWS.

Para establecer la variable de entorno, llame al siguiente comando desde el terminal:

```
export AWS_REGION=supported-region
```

Donde *supported-region* es el identificador de región (por ejemplo, us-west-2).

Instale el CodeDeploy agente

- Si se trata de una instancia en las instalaciones de Ubuntu Server, siga las instrucciones de [Instale el CodeDeploy agente para Ubuntu Server](#) y después vuelva a esta página.
- Si se trata de una instancia en las instalaciones de RHEL, siga las instrucciones de [Instalar el CodeDeploy agente para Amazon Linux o RHEL](#) y después vuelva a esta página.
- Si se trata de una instancia en las instalaciones de Windows Server, siga las instrucciones de [Instale el CodeDeploy agente para Windows Server](#) y después vuelva a esta página.

Paso 5: Registre la instancia local con CodeDeploy

En las instrucciones de este paso se da por supuesto que va a registrar la instancia on-premises desde la propia instancia on-premises. Puede registrar una instancia local desde un dispositivo o instancia independiente que la tenga AWS CLI instalada y configurada.



AWS CLI Utilícela para registrar la instancia local con el CodeDeploy fin de utilizarla en las implementaciones.

Antes de poder usar el AWS CLI, necesitará el ARN de las credenciales de sesión temporales en las que creó. [Paso 3: Añadir un archivo de configuración a la instancia en las instalaciones](#) Por ejemplo, en el caso de una instancia que identifique como AssetTag12010298EX:

```
arn:sts:iam::123456789012:assumed-role/CodeDeployInstanceRole/AssetTag12010298EX
```

Ejecute el comando [register-on-premises-instance](#) especificando lo siguiente:

- Un nombre que identifique de manera única la instancia en las instalaciones (con la opción `--instance-name`).

#### Important

Para ayudar a identificar la instancia on-premises, especialmente a efectos de depuración, es absolutamente recomendable que especifique un nombre que se corresponda con algunas características exclusivas de la instancia on-premises (por ejemplo, el nombre de sesión de las credenciales de STS y el número de serie de un identificador de recurso interno, si procede). Si especifica una dirección MAC como nombre, tenga en cuenta que las direcciones MAC contienen caracteres CodeDeploy no permitidos, como dos puntos (:). Para obtener una lista de los caracteres permitidos, consulte [CodeDeploy cuotas](#).

- El ARN de sesión de IAM que configura para autenticar varias instancias on-premises en [Paso 1: Crear el rol de IAM que asumirán las instancias en las instalaciones](#).

Por ejemplo:

```
aws deploy register-on-premises-instance --instance-name name-of-instance --iam-session-arn arn:aws:sts::account-id:assumed-role/role-to-assume/session-name
```

Donde:

- *name-of-instance* es el nombre que se utiliza para identificar la instancia local, por ejemplo. AssetTag12010298EX
- *account-id* es el ID de cuenta de 12 dígitos de su organización, como 111222333444.

- *role-to-assume* es el nombre del rol de IAM que creaste para la instancia, por ejemplo. `CodeDeployInstanceRole`
- *session-name* es el nombre del rol de la sesión que especificó en [Paso 2: Genera credenciales temporales para una instancia individual mediante AWS STS](#).

## Paso 6: Etiquetar la instancia en las instalaciones

Puedes usar la consola AWS CLI o la CodeDeploy consola para etiquetar la instancia local. (CodeDeploy usa etiquetas de instancias locales para identificar los objetivos de implementación durante una implementación).

### Para etiquetar la instancia en las instalaciones (CLI)

- Llama al comando [add-tags-to-on-premises-instances](#) y especifica:
  - El nombre que identifica de manera única la instancia on-premises (con la opción `--instance-names`).
  - El nombre de la clave de etiqueta y el valor de etiqueta de la instancia on-premises que desea utilizar (con la opción `--tags`). Debe especificar un nombre y un valor. CodeDeploy no permite etiquetas de instancias locales que solo tengan valores.

Por ejemplo:

```
aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
--tags Key=Name,Value=CodeDeployDemo-OnPrem
```

### Para etiquetar la instancia on-premises (consola)

1. Inicia sesión en la CodeDeploy consola AWS Management Console y ábrela en <https://console.aws.amazon.com/codedeploy>.

#### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Instancias en las instalaciones.
3. En la lista de instancias on-premises, elija el nombre de la instancia on-premises que desea etiquetar.

4. En la lista de etiquetas, seleccione o introduzca la clave y el valor que desee para la etiqueta. Cuando introduzca la clave y el valor de la etiqueta, aparecerá otra fila. Puede repetir esta acción para un máximo de 10 etiquetas. Para eliminar una etiqueta, elija Eliminar.
5. Después de añadir las etiquetas, seleccione Update Tags.

## Paso 7: Implementar revisiones de aplicaciones en la instancia on-premises

Ahora está listo para implementar revisiones de aplicaciones en la instancia on-premises registrada y etiquetada.

La implementación de revisiones de aplicaciones en instancias en las instalaciones es similar a la implementación de revisiones de aplicaciones en instancias de Amazon EC2. Para ver instrucciones, consulte [Cree una implementación con CodeDeploy](#). Estas instrucciones incluyen un enlace a los requisitos previos, incluida la creación de una aplicación, la creación de un grupo de implementaciones y la preparación de la revisión de una aplicación. Si necesita una revisión de una aplicación sencilla para implementar, puede crear la que se describe en [Paso 2: Crear una revisión de aplicación de ejemplo](#) en el [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#).

### Important

Si reutilizas un rol de CodeDeploy servicio como parte de la creación de un grupo de implementación dirigido a instancias locales, debes `Tag: get*` incluirlo en la `Action` parte de la declaración de política del rol de servicio. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).

## Paso 8: Hacer un seguimiento de las implementaciones en la instancia en las instalaciones

Después de implementar una revisión de una aplicación en instancias en las instalaciones registradas y etiquetadas, puede hacer un seguimiento del progreso de la implementación.

El seguimiento de las implementaciones en instancias en las instalaciones es similar al seguimiento de las implementaciones en instancias de Amazon EC2. Para obtener instrucciones, consulte [Ver detalles CodeDeploy de la implementación](#).

## Uso del comando “register” (ARN de usuario de IAM) para registrar una instancia en las instalaciones

### Important

No se recomienda registrar una instancia con un usuario de IAM porque utiliza credenciales estáticas (permanentes) para la autenticación. Para mejorar la seguridad, recomendamos registrar una instancia con credenciales temporales para la autenticación. Para obtener más información, consulte [Usa el register-on-premises-instance comando \(ARN de sesión de IAM\) para registrar una instancia local](#).

### Important

Asegúrese de tener un plan para rotar las claves de acceso del usuario de IAM (credenciales permanentes). Para obtener más información, consulte [Rotación de las claves de acceso](#).

En esta sección, se describe cómo configurar una instancia local, registrarla y etiquetarla CodeDeploy con el mínimo esfuerzo. El comando register resulta especialmente útil cuando se trabaja con una sola flota o con flotas pequeñas de instancias en las instalaciones. Solo puede utilizar el comando register si emplea un ARN de usuario de IAM para autenticar una instancia. No puede utilizar el comando register con un ARN de sesión de IAM para la autenticación.

Al usar el register comando, puede CodeDeploy hacer lo siguiente:

- Crea un usuario de IAM AWS Identity and Access Management para la instancia local si no especificas uno con el comando.
- Guardar las credenciales del usuario de IAM en un archivo de configuración de la instancia en las instalaciones.
- Registre la instancia local con. CodeDeploy
- Añadir etiquetas a la instancia en las instalaciones, si las especifica como parte del comando.

### Note

El [register-on-premises-instance](#) comando es una alternativa al comando [register](#). Usa el register-on-premises-instance comando si quieres configurar una instancia local y

registrarla y etiquetarla CodeDeploy casi siempre por tu cuenta. El comando `register-on-premises-instance` también le da la opción de utilizar un ARN de sesión de IAM para registrar instancias en lugar de un ARN de usuario de IAM. Este enfoque proporciona una gran ventaja si tiene grandes flotas de instancias en las instalaciones. Específicamente, puede utilizar un solo ARN de sesión de IAM para autenticar múltiples instancias en lugar de tener que crear un usuario de IAM para cada instancia en las instalaciones a la vez. Para obtener más información, consulte [Usa el register-on-premises-instance comando \(ARN del usuario de IAM\) para registrar una instancia local](#) y [Usa el register-on-premises-instance comando \(ARN de sesión de IAM\) para registrar una instancia local](#).

## Temas

- [Paso 1: Instalar y configurar AWS CLI la instancia local](#)
- [Paso 2: Llamar al comando “register”](#)
- [Paso 3: Llamar al comando “install”](#)
- [Paso 4: Implementar revisiones de la aplicación en la instancia en las instalaciones](#)
- [Paso 5: Hacer un seguimiento de las implementaciones en la instancia en las instalaciones](#)

## Paso 1: Instalar y configurar AWS CLI la instancia local

1. Instálelo AWS CLI en la instancia local. Siga las instrucciones en [Configuración inicial de la AWS CLI](#) en la Guía del usuario de la AWS Command Line Interface .

### Note

CodeDeploy Los comandos para trabajar con instancias locales están disponibles en la AWS CLI versión 1.7.19 y versiones posteriores. Si AWS CLI ya la tienes instalada, llama `aws --version` para comprobar su versión.

2. Configure AWS CLI la instancia local. Siga las instrucciones de [Configuración de la AWS CLI](#) en la Guía del usuario de la AWS Command Line Interface .


### Important

Al configurar AWS CLI (por ejemplo, mediante una llamada al `aws configure` comando), asegúrese de especificar el ID de clave secreta y la clave de acceso secreta de un

usuario de IAM que tenga, como mínimo, los siguientes permisos de AWS acceso además de los permisos especificados en [Requisitos previos para configurar una instancia en las instalaciones](#). Esto permite descargar e instalar el CodeDeploy agente en la instancia local. Los permisos de acceso podrían tener un aspecto similar a este:

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:*",
 "iam:CreateAccessKey",
 "iam:CreateUser",
 "iam>DeleteAccessKey",
 "iam>DeleteUser",
 "iam>DeleteUserPolicy",
 "iam:ListAccessKeys",
 "iam:ListUserPolicies",
 "iam:PutUserPolicy",
 "iam:GetUser",
 "tag:getTagKeys",
 "tag:getTagValues",
 "tag:GetResources"
],
 "Resource" : "*"
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "s3:Get*",
 "s3:List*"
],
 "Resource" : [
 "arn:aws:s3:::aws-codedeploy-us-east-2/*",
 "arn:aws:s3:::aws-codedeploy-us-east-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-2/*",
 "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-1/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-2/*",
 "arn:aws:s3:::aws-codedeploy-eu-west-3/*",
 "arn:aws:s3:::aws-codedeploy-eu-central-1/*",
```

```
"arn:aws:s3:::aws-codedeploy-il-central-1/*",
"arn:aws:s3:::aws-codedeploy-ap-east-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
"arn:aws:s3:::aws-codedeploy-ap-south-1/*",
"arn:aws:s3:::aws-codedeploy-sa-east-1/*"
]
}
]
}
```

 Note


Si ve errores de acceso denegado al intentar acceder a uno de los buckets de Amazon S3 mostrados anteriormente, intente omitir la parte de /\* del ARN de recursos del bucket, por ejemplo `arn:aws:s3:::aws-codedeploy-sa-east-1`.

## Paso 2: Llamar al comando “register”

Para este paso se da por sentado que va a registrar la instancia on-premises desde la propia instancia on-premises. También puede registrar una instancia local desde un dispositivo o instancia independiente que la tenga AWS CLI instalada y configurada tal y como se describe en el paso anterior.

Use el comando `register` AWS CLI para llamar al comando [register](#) y especifique:

- Un nombre que identifica de forma exclusiva a la instancia local CodeDeploy (con la `--instance-name` opción).

 Important

Para ayudarle a identificar la instancia on-premises en otro momento, especialmente a efectos de depuración, recomendamos encarecidamente que utilice un nombre que se asigne a algunas características exclusivas de la instancia on-premises (por ejemplo, el

número de serie o algún identificador único de recurso interno, si procede). Si especificas una dirección MAC para un nombre, ten en cuenta que las direcciones MAC contienen caracteres CodeDeploy no permitidos, como dos puntos (:). Para obtener una lista de los caracteres permitidos, consulte [CodeDeploy cuotas](#).

- Si lo desea, el ARN de un usuario de IAM existente que desea asociar a esta instancia en las instalaciones (con la opción `--iam-user-arn`). Para obtener el ARN de un usuario de IAM, llame al comando [get-user](#) o elija el nombre de usuario de IAM en la sección Users de la consola de IAM y, a continuación, busque el valor ARN de usuario en la sección Resumen. Si no se especifica esta opción, CodeDeploy creará un usuario de IAM en su nombre en su AWS cuenta y lo asociará a la instancia local.

#### Important

Si especifica la opción `--iam-user-arn`, además debe crear manualmente el archivo de configuración de la instancia en las instalaciones, como se describe en [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#).

Puede asociar un solo usuario de IAM a una sola instancia en las instalaciones. Intentar asociar un solo usuario de IAM a múltiples instancias en las instalaciones puede producir errores, implementaciones fallidas en dichas instancias en las instalaciones o implementaciones en dichas instancias en las instalaciones que permanecen en estado pendiente perpetuo.

- Opcionalmente, un conjunto de etiquetas de instancias locales (con la `--tags` opción) que CodeDeploy se utilizarán para identificar el conjunto de instancias de Amazon EC2 en las que se va a implementar. Especifique cada etiqueta con `Key=tag-key, Value=tag-value` (por ejemplo, `Key=Name, Value=Beta` `Key=Name, Value=WestRegion`). Si no se especifica esta opción, no se registrarán etiquetas. Para registrar las etiquetas más adelante, ejecute el comando [add-tags-to-on-premises-instances](#).
- De forma opcional, la AWS región en la que se registrará la instancia local CodeDeploy (con la opción) `--region` Debe ser una de las regiones admitidas que se encuentran en la lista de [Regiones y puntos de enlace](#) de la Referencia general de AWS (por ejemplo, `us-west-2`). Si no se especifica esta opción, se utilizará la AWS región predeterminada asociada al usuario de IAM que realiza la llamada.

Por ejemplo:



```
aws deploy register --instance-name AssetTag12010298EX --iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem --tags Key=Name,Value=CodeDeployDemo-OnPrem --region us-west-2
```

El comando register hace lo siguiente:

1. Si no se especifica un usuario de IAM existente, lo crea, le asocia los permisos necesarios y genera la clave secreta y la clave de acceso secreta correspondientes. La instancia local utilizará este usuario de IAM y sus permisos y credenciales para autenticarse e interactuar con él. CodeDeploy
2. Registra la instancia local con CodeDeploy
3. Si se especifica, asocia CodeDeploy las etiquetas especificadas con la `--tags` opción al nombre de la instancia local registrada.
4. Si ya se ha creado el usuario de IAM, también crea el archivo de configuración necesario en el mismo directorio desde el que se ejecutó el comando register.

Si este comando encuentra algún error, aparece un mensaje en el que se describe cómo realizar manualmente los pasos faltantes. En caso contrario, aparece un mensaje de confirmación en el que se describe cómo ejecutar el comando install según se indica en el siguiente paso.

### Paso 3: Llamar al comando "install"

Desde la instancia local, usa el comando AWS CLI para llamar al comando [install](#) y especifica:

- La ruta al archivo de configuración (con la opción `--config-file`).
- Si lo desea, si se reemplazará el archivo de configuración que ya existe en la instancia on-premises (con la opción `--override-config`). Si no se especifica, no se sustituirá el archivo de configuración existente.
- De forma opcional, la AWS región en la que se registrará la instancia local CodeDeploy (con la `--region` opción). Debe ser una de las regiones admitidas que se encuentran en la lista de [Regiones y puntos de enlace](#) de la Referencia general de AWS (por ejemplo, `us-west-2`). Si no se especifica esta opción, se utilizará la AWS región predeterminada asociada al usuario de IAM que realiza la llamada.
- Opcionalmente, una ubicación personalizada desde la que instalar el CodeDeploy agente (con la `--agent-installer` opción). Esta opción resulta útil para instalar una versión personalizada del CodeDeploy agente que CodeDeploy no sea compatible oficialmente (por ejemplo, una versión

personalizada basada en el repositorio de [CodeDeployagentes](#) de GitHub). El valor debe ser la ruta a un bucket de Amazon S3 que contenga cualquiera de los siguientes:

- Un script de instalación del CodeDeploy agente (para sistemas operativos basados en Linux o Unix, similar al archivo de instalación del repositorio del [CodeDeployagente](#)). GitHub
- Un archivo de paquete de instalación del CodeDeploy agente (.msi) (para sistemas operativos basados en Windows).

Si no se especifica esta opción, CodeDeploy intentará instalar desde su propia ubicación una versión oficial del CodeDeploy agente que sea compatible con el sistema operativo de la instancia local.

Por ejemplo:

```
aws deploy install --override-config --config-file /tmp/codedeploy.onpremises.yml --
region us-west-2 --agent-installer s3://aws-codedeploy-us-west-2/latest/codedeploy-
agent.msi
```

El comando `install` hace lo siguiente:

1. Comprueba si la instancia en las instalaciones es una instancia de Amazon EC2. Si lo es, aparece un mensaje de error.
2. Copia el archivo de configuración de las instancias locales de la ubicación especificada de la instancia a la ubicación en la que el CodeDeploy agente espera encontrarlo, siempre que el archivo no esté ya en esa ubicación.

Para Ubuntu Server y Red Hat Enterprise Linux (RHEL), es `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`.

Para Windows Server, es `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.


Si se especificó la opción `--override-config`, crea o sobrescribe el archivo.

3. Instala el CodeDeploy agente en la instancia local y, a continuación, la inicia.

**Paso 4: Implementar revisiones de la aplicación en la instancia en las instalaciones**

Ahora está listo para implementar revisiones de aplicaciones en la instancia on-premises registrada y etiquetada.

La implementación de revisiones de aplicaciones en instancias en las instalaciones es similar a la implementación de revisiones de aplicaciones en instancias de Amazon EC2. Para ver instrucciones, consulte [Cree una implementación con CodeDeploy](#). Estas instrucciones incluyen un enlace a los requisitos previos, incluida la creación de una aplicación, la creación de un grupo de implementaciones y la preparación de la revisión de una aplicación. Si necesita una revisión de una aplicación sencilla para implementar, puede crear la que se describe en [Paso 2: Crear una revisión de aplicación de ejemplo](#) en el [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#).

 Important

Si reutilizas un rol de CodeDeploy servicio existente como parte de la creación de un grupo de implementación orientado a instancias locales, debes `Tag:get*` incluirlo en esa `Action` parte de la declaración de política del rol de servicio. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).


Paso 5: Hacer un seguimiento de las implementaciones en la instancia en las instalaciones

Después de implementar una revisión de una aplicación en instancias en las instalaciones registradas y etiquetadas, puede hacer un seguimiento del progreso de la implementación.

El seguimiento de las implementaciones en instancias en las instalaciones es similar al seguimiento de las implementaciones en instancias de Amazon EC2. Para ver instrucciones, consulte [Ver detalles CodeDeploy de la implementación](#).

Para obtener más opciones, consulte [Administrar las operaciones de instancias locales en CodeDeploy](#).

Usa el `register-on-premises-instance` comando (ARN del usuario de IAM) para registrar una instancia local

 Important

No se recomienda registrar una instancia con un usuario de IAM porque utiliza credenciales estáticas (permanentes) para la autenticación. Para mejorar la seguridad, recomendamos registrar una instancia con credenciales temporales para la autenticación. Para obtener más

información, consulte [Usa el register-on-premises-instance comando \(ARN de sesión de IAM\) para registrar una instancia local.](#)

 Important

Asegúrese de tener un plan para rotar las claves de acceso del usuario de IAM (credenciales permanentes). Para obtener más información, consulte [Rotación de las claves de acceso.](#)

Sigue estas instrucciones para configurar una instancia local y registrarla y etiquetarla CodeDeploy prácticamente por tu cuenta, utilizando credenciales de usuario de IAM estáticas para la autenticación.

## Temas

- [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#)
- [Paso 2: Asignar permisos al usuario de IAM](#)
- [Paso 3: Obtener las credenciales del usuario de IAM](#)
- [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#)
- [Paso 5: Instalar y configurar el AWS CLI](#)
- [Paso 6: Establecer la variable de entorno AWS\\_REGION \(solo Ubuntu Server y RHEL\)](#)
- [Paso 7: Instalar el CodeDeploy agente](#)
- [Paso 8: registre la instancia local con CodeDeploy](#)
- [Paso 9: Etiquetar la instancia en las instalaciones](#)
- [Paso 10: Implementar revisiones de aplicaciones en la instancia en las instalaciones](#)
- [Paso 11: Hacer un seguimiento de las implementaciones en la instancia en las instalaciones](#)

## Paso 1: Crear un usuario de IAM para la instancia en las instalaciones

Creas un usuario de IAM que la instancia local utilice para autenticarse e interactuar con él.  
CodeDeploy

**⚠ Important**

Debe crear un usuario de IAM diferente para cada instancia en las instalaciones que emplee. Si intentas reutilizar un usuario de IAM individual para varias instancias locales, es posible que no puedas registrar o etiquetar correctamente esas instancias locales. CodeDeploy Las implementaciones en esas instancias on-premises pueden quedarse bloqueadas en un estado pendiente perpetuo o producir errores.

Le recomendamos que asigne al usuario de IAM un nombre que identifique su propósito, como `-CodeDeployUser OnPrem`

Puede utilizar la consola de IAM AWS CLI o la consola de IAM para crear un usuario de IAM. Para obtener información, consulte [Creación de un usuario de IAM en su cuenta de AWS](#).

**⚠ Important**

Ya sea que utilice la consola de IAM AWS CLI o la consola de IAM para crear un nuevo usuario de IAM, anote el ARN de usuario proporcionado al usuario. Necesitará esta información más tarde en [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#) y [Paso 8: registre la instancia local con CodeDeploy](#).

## Paso 2: Asignar permisos al usuario de IAM

Si su instancia en las instalaciones va a implementar revisiones de aplicaciones desde buckets de Amazon S3, debe asignar al usuario de IAM los permisos necesarios para interactuar con esos buckets. Puede utilizar la consola de IAM AWS CLI o la consola de IAM para asignar permisos.

**ℹ Note**

Si va a implementar las revisiones de las aplicaciones únicamente desde GitHub los repositorios, omite este paso y vaya directamente a [Paso 3: Obtener las credenciales del usuario de IAM](#) (Todavía necesitará información acerca del usuario de IAM que creó en [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#). Se utilizará en pasos posteriores.)

## Para asignar permisos (CLI)

1. Cree un archivo con el siguiente contenido de política en la instancia de Amazon EC2 o en el dispositivo que va a utilizar para llamar a la AWS CLI. Asigne al archivo un nombre como **CodeDeploy-OnPrem-Permissions.json** y, a continuación, guárdelo.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "*"
 }
]
}
```

### Note

Le recomendamos que restrinja esta política únicamente a los buckets de Amazon S3 a los que su instancia en las instalaciones necesite tener acceso. Si restringe esta política, asegúrese de dar acceso también a los buckets de Amazon S3 que contienen el AWS CodeDeploy agente. De lo contrario, podría producirse un error cada vez que el CodeDeploy agente se instale o actualice en la instancia local asociada.

Por ejemplo:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Resource": [
 "arn:aws:s3:::replace-with-your-s3-bucket-name/*",
 "arn:aws:s3:::aws-codedeploy-us-east-2/*",

```

```

"arn:aws:s3:::aws-codedeploy-us-east-1/*",
"arn:aws:s3:::aws-codedeploy-us-west-1/*",
"arn:aws:s3:::aws-codedeploy-us-west-2/*",
"arn:aws:s3:::aws-codedeploy-ca-central-1/*",
"arn:aws:s3:::aws-codedeploy-eu-west-1/*",
"arn:aws:s3:::aws-codedeploy-eu-west-2/*",
"arn:aws:s3:::aws-codedeploy-eu-west-3/*",
"arn:aws:s3:::aws-codedeploy-eu-central-1/*",
"arn:aws:s3:::aws-codedeploy-eu-central-2/*",
"arn:aws:s3:::aws-codedeploy-eu-north-1/*",
"arn:aws:s3:::aws-codedeploy-eu-south-1/*",
"arn:aws:s3:::aws-codedeploy-eu-south-2/*",
"arn:aws:s3:::aws-codedeploy-il-central-1/*",
"arn:aws:s3:::aws-codedeploy-ap-east-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-3/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-3/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
"arn:aws:s3:::aws-codedeploy-ap-south-1/*",
"arn:aws:s3:::aws-codedeploy-ap-south-2/*",
"arn:aws:s3:::aws-codedeploy-me-central-1/*",
"arn:aws:s3:::aws-codedeploy-me-south-1/*",
"arn:aws:s3:::aws-codedeploy-sa-east-1/*"
]
}
]
}

```

2. Ejecute el `put-user-policy` comando y especifique el nombre del usuario de IAM (con la `--user-name` opción), un nombre para la política (con la `--policy-name` opción) y la ruta al documento de política recién creado (con la `--policy-document` opción). Por ejemplo, suponiendo que el archivo **CodeDeploy-OnPrem-Permissions.json** está en el mismo directorio (carpeta) desde el que llama a este comando:

#### Important

Asegúrese de incluir `file://` antes del nombre de archivo. Es obligatorio en este comando.

```
aws iam put-user-policy --user-name CodeDeployUser-OnPrem --policy-name CodeDeploy-OnPrem-Permissions --policy-document file://CodeDeploy-OnPrem-Permissions.json
```

Para asignar permisos (consola)

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, seleccione Políticas y, a continuación, seleccione Create Policy. Si aparece el botón Empezar, selecciónelo y, a continuación, seleccione Crear política.
3. Junto a Create Your Own Policy, seleccione Select.
4. En el cuadro Policy Name (Nombre de la política), escriba un nombre para esta política (por ejemplo, **CodeDeploy-OnPrem-Permissions**).
5. En el cuadro Documento de política, escriba o pegue la siguiente expresión de permisos, que AWS CodeDeploy permite implementar revisiones de aplicaciones desde cualquier bucket de Amazon S3 especificado en la política en la instancia local en nombre del usuario de IAM:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "*"
 }
]
}
```

6. Elija Crear política.
7. En el panel de navegación, seleccione Usuarios.
8. En la lista de usuarios, busque y elija el nombre del usuario de IAM que creó en [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#).
9. En la pestaña Permissions, en Managed Policies, seleccione Attach Policy.



10. Seleccione la política denominada **CodeDeploy-OnPrem-Permissions** y, a continuación, elija **Attach Policy** (Asociar política).

### Paso 3: Obtener las credenciales del usuario de IAM

Obtenga el ID de la clave secreta y la clave de acceso secreta para el usuario de IAM. Los necesitará para [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#). Puede utilizar la consola de IAM AWS CLI o la consola de IAM para obtener el ID de la clave secreta y la clave de acceso secreta.

#### Note

Si ya tiene el ID de la clave secreta y la clave de acceso secreta, omita este paso y vaya directamente a [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#). Los usuarios necesitan acceso programático si quieren interactuar con personas AWS ajenas a. AWS Management Console La forma de conceder el acceso programático depende del tipo de usuario que acceda. AWS Para conceder acceso programático a los usuarios, elija una de las siguientes opciones.

| ¿Qué usuario necesita acceso programático?                                   | Para                                                                                                       | Mediante                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identidad del personal<br>(Usuarios administrados en el IAM Identity Center) | Usa credenciales temporales para firmar las solicitudes programáticas a los AWS CLI AWS SDK o las API. AWS | Siga las instrucciones de la interfaz que desea utilizar: <ul style="list-style-type: none"> <li>• Para ello AWS CLI, consulte <a href="#">Configuración del uso AWS IAM Identity Center en AWS CLI la Guía del AWS Command Line Interface</a> usuario.</li> <li>• Para ver AWS los SDK, las herramientas y las AWS API, consulte la <a href="#">autenticación del IAM Identity Center</a> en la Guía</li> </ul> |

| ¿Qué usuario necesita acceso programático? | Para                                                                                                                                      | Mediante                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                            |                                                                                                                                           | de referencia de AWS los SDK y las herramientas.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| IAM                                        | Utilice credenciales temporales para firmar las solicitudes programáticas a los AWS SDK o las AWS CLI API. AWS                            | Siga las instrucciones de <a href="#">Uso de credenciales temporales con AWS recursos</a> de la Guía del usuario de IAM.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| IAM                                        | (No recomendado)<br>Utilice credenciales de larga duración para firmar las solicitudes programáticas a los AWS CLI AWS SDK o las API. AWS | Siga las instrucciones de la interfaz que desea utilizar: <ul style="list-style-type: none"> <li>• Para ello AWS CLI, consulte <a href="#">Autenticación con credenciales de usuario de IAM en la Guía del usuario.AWS Command Line Interface</a></li> <li>• Para obtener información AWS sobre los SDK y las herramientas, consulte <a href="#">Autenticarse con credenciales de larga duración</a> en la Guía de referencia de los AWS SDK y las herramientas.</li> <li>• Para obtener información AWS sobre las API, consulte <a href="#">Administrar las claves de acceso para los usuarios de IAM</a> en la Guía del usuario de IAM.</li> </ul> |

## Para obtener las credenciales (CLI)

1. Ejecuta el [list-access-keys](#) comando, especifica el nombre del usuario de IAM (con la `--user-name` opción) y consulta solo los ID de las claves de acceso (con las `--query` opciones y). `--output` Por ejemplo:

```
aws iam list-access-keys --user-name CodeDeployUser-OnPrem --query
"AccessKeyMetadata[*].AccessKeyId" --output text
```

2. Si no aparece ninguna tecla en el resultado o solo aparece información sobre una tecla en el resultado, [create-access-key](#) ejecute el comando y especifique el nombre del usuario de IAM (con la `--user-name` opción):

```
aws iam create-access-key --user-name CodeDeployUser-OnPrem
```

En la salida de la llamada al comando `create-access-key`, anote el valor de los campos `AccessKeyId` y `SecretAccessKey`. Necesitará esta información en [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#).

### Important

Esta será la única vez que tendrá acceso a la clave de acceso secreta. Si olvida o pierde el acceso a esta clave de acceso secreta, tendrá que generar una nueva siguiendo los pasos de [Paso 3: Obtener las credenciales del usuario de IAM](#).

3. Si ya hay dos claves de acceso en la lista, debe eliminar una de ellas. Para ello, [delete-access-key](#) ejecute el comando y especifique el nombre del usuario de IAM (con la `--user-name` opción) y el ID de la clave de acceso que desee eliminar (con la `--access-key-id` opción). A continuación, llame al comando `create-access-key`, como se ha descrito anteriormente en este paso. A continuación se muestra un ejemplo de cómo llamar al comando `delete-access-key`:

```
aws iam delete-access-key --user-name CodeDeployUser-OnPrem --access-key-id access-
key-ID
```

### Important


Si llama al comando `delete-access-key` para eliminar una de estas claves de acceso, y una instancia en las instalaciones ya está utilizando esta clave de acceso como

se describe en [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#), tendrá que volver a seguir las instrucciones de [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#) para especificar otro ID de clave de acceso y otra clave de acceso secreta asociada a este usuario de IAM. De lo contrario, todas las implementaciones de esa instancia on-premises pueden quedarse bloqueadas en un estado pendiente perpetuo o producir errores.

Para obtener las credenciales (consola)

1.
  - a. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
  - b. Si la lista de usuarios no se muestra, en el panel de navegación, elija Users.
  - c. En la lista de usuarios, busque y elija el nombre del usuario de IAM que creó en [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#).
2. En la pestaña Security credentials, si en la lista no figura ninguna clave o si figura una sola una clave, seleccione Create access key.

Si se muestran dos claves de acceso, debe eliminar una de ellas. Elija Delete junto a una de las claves de acceso y luego seleccione Create access key.

 Important

Si elige Eliminar junto a una de estas claves de acceso, y una instancia en las instalaciones ya está utilizando esta clave de acceso como se describe en [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#), tendrá que volver a seguir las instrucciones de [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#) para especificar otro ID de clave de acceso y otra clave de acceso secreta asociada a este usuario de IAM. De lo contrario, las implementaciones de esa instancia on-premises pueden quedarse bloqueadas en un estado pendiente perpetuo o producir errores.

3. Elija Show y anote el ID de la clave de acceso y la clave de acceso secreta. Necesitará esta información para el siguiente paso. Como alternativa, puede elegir Download .csv file para guardar una copia del ID de la clave de acceso y la clave de acceso secreta.

**⚠ Important**

A menos que anote o descargue las credenciales, esta será la única vez que tendrá acceso a la clave de acceso secreta. Si olvida o pierde el acceso a esta clave de acceso secreta, tendrá que generar una nueva siguiendo los pasos de [Paso 3: Obtener las credenciales del usuario de IAM](#).

4. Seleccione Close para volver a la página Users > **IAM User Name**.

**Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones**

Añada un archivo de configuración a la instancia on-premises, utilizando los permisos raíz o de administrador. Este archivo de configuración se utilizará para declarar las credenciales del usuario de IAM y la AWS región de destino en la que se utilizará. CodeDeploy El archivo debe añadirse a una ubicación específica de la instancia on-premises. El archivo debe incluir el ARN del usuario de IAM, el ID de clave secreta, la clave de acceso secreta y la AWS región de destino. Este archivo debe seguir un formato específico.

1. Cree un archivo denominado `codedeploy.onpremises.yml` (si se trata de una instancia en las instalaciones de Ubuntu Server o RHEL) o `conf.onpremises.yml` (en el caso de una instancia en las instalaciones de Windows) en la siguiente ubicación de la instancia en las instalaciones:
  - Para Ubuntu Server: `/etc/codedeploy-agent/conf`
  - Para Windows Server: `C:\ProgramData\Amazon\CodeDeploy`
2. Utilice un editor de texto para añadir la siguiente información al archivo `codedeploy.onpremises.yml` o `conf.onpremises.yml` recién creado:

```

aws_access_key_id: secret-key-id
aws_secret_access_key: secret-access-key
iam_user_arn: iam-user-arn
region: supported-region
```

Donde:

- *secret-key-id* es la ID de clave secreta del usuario de IAM correspondiente que anotó en o. [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#) [Paso 3: Obtener las credenciales del usuario de IAM](#)
- *secret-access-key* es la clave de acceso secreta del usuario de IAM correspondiente que anotó en [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#) o. [Paso 3: Obtener las credenciales del usuario de IAM](#)
- *iam-user-arn* es el ARN del usuario de IAM que indicó anteriormente. [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#)
- *supported-region* es el identificador de una región compatible con la ubicación de CodeDeploy las aplicaciones, los grupos de implementación y las revisiones de las aplicaciones (por CodeDeploy ejemplo,). us-west-2 Para obtener una lista de regiones, consulte [Regiones y puntos de conexión](#) en la Referencia general de AWS.

**⚠ Important**

Si eligió Delete junto a una de las claves de acceso en [Paso 3: Obtener las credenciales del usuario de IAM](#), y su instancia en las instalaciones ya está utilizando el ID de la clave de acceso y la clave de acceso secreta asociados, tendrá que seguir las instrucciones de [Paso 4: Añadir un archivo de configuración a la instancia en las instalaciones](#) para especificar otro ID de clave de acceso y otra clave de acceso secreta asociados a este usuario de IAM. De lo contrario, todas las implementaciones de su instancia on-premises pueden quedarse bloqueadas en un estado pendiente perpetuo o producir errores.

## Paso 5: Instalar y configurar el AWS CLI

Instale y configure AWS CLI la instancia local. (Se AWS CLI usará [Paso 7: Instalar el CodeDeploy agente](#) para descargar e instalar el CodeDeploy agente en la instancia local).

1. Para instalarlo AWS CLI en la instancia local, siga las instrucciones de la Guía del AWS Command Line Interface usuario sobre [cómo configurarlo con AWS CLI](#) el.

**Note**

CodeDeploy los comandos para trabajar con instancias locales estuvieron disponibles en la versión 1.7.19 de AWS CLI. Si AWS CLI ya tienes una versión de la instalada, puedes comprobar su versión llamando a `aws --version`

2. Para configurar AWS CLI la instancia local, siga las instrucciones de la Guía del AWS Command Line Interface usuario sobre [cómo configurar AWS CLI la instancia local](#).

**Important**

Al configurar AWS CLI (por ejemplo, mediante una llamada al `aws configure` comando), asegúrese de especificar el ID de clave secreta y la clave de acceso secreta de un usuario de IAM que tenga, como mínimo, los siguientes permisos de AWS acceso además de los permisos de acceso especificados en la [Requisitos previos para configurar una instancia en las instalaciones](#). Esto le permite descargar e instalar el CodeDeploy agente en la instancia local:

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:*"
],
 "Resource" : "*"
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "s3:Get*",
 "s3:List*"
],
 "Resource" : [
 "arn:aws:s3:::aws-codedeploy-us-east-2/*",
 "arn:aws:s3:::aws-codedeploy-us-east-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-1/*",
 "arn:aws:s3:::aws-codedeploy-us-west-2/*",
 "arn:aws:s3:::aws-codedeploy-ca-central-1/*",
```

```
"arn:aws:s3:::aws-codedeploy-eu-west-1/*",
"arn:aws:s3:::aws-codedeploy-eu-west-2/*",
"arn:aws:s3:::aws-codedeploy-eu-west-3/*",
"arn:aws:s3:::aws-codedeploy-eu-central-1/*",
"arn:aws:s3:::aws-codedeploy-il-central-1/*",
"arn:aws:s3:::aws-codedeploy-ap-east-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-northeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-1/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-2/*",
"arn:aws:s3:::aws-codedeploy-ap-southeast-4/*",
"arn:aws:s3:::aws-codedeploy-ap-south-1/*",
"arn:aws:s3:::aws-codedeploy-sa-east-1/*"
]
}
]
```

Estos permisos de acceso se pueden asignar al usuario de IAM que creó en [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#) o a otro usuario de IAM. Para asignar estos permisos a un usuario de IAM, siga las instrucciones de [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#), utilizando estos permisos de acceso en lugar de los indicados en ese paso.

Paso 6: Establecer la variable de entorno AWS\_REGION (solo Ubuntu Server y RHEL)

Si no ejecuta Ubuntu Server o RHEL en su instancia en las instalaciones, omita este paso y vaya directamente a [Paso 7: Instalar el CodeDeploy agente](#).

Instale el CodeDeploy agente en una instancia local de Ubuntu Server o RHEL y habilite la instancia para que actualice el CodeDeploy agente cada vez que haya una nueva versión disponible. Para ello, defina la variable de AWS\_REGION entorno de la instancia en el identificador de una de las regiones compatibles. CodeDeploy Le recomendamos que establezca el valor en la región en la que se encuentran CodeDeploy las aplicaciones, los grupos de implementación y las revisiones de las aplicaciones (por ejemplo, us-west-2). Para obtener una lista de regiones, consulte [Regiones y puntos de conexión](#) en la Referencia general de AWS.

Para establecer la variable de entorno, llame al siguiente comando desde el terminal:



```
export AWS_REGION=supported-region
```

Donde *supported-region* es el identificador de región (por ejemplo, us-west-2).

## Paso 7: Instalar el CodeDeploy agente

Instale el CodeDeploy agente en la instancia local:

- Si se trata de una instancia en las instalaciones de Ubuntu Server, siga las instrucciones de [Instale el CodeDeploy agente para Ubuntu Server](#) y después vuelva a esta página.
- Si se trata de una instancia en las instalaciones de RHEL, siga las instrucciones de [Instalar el CodeDeploy agente para Amazon Linux o RHEL](#) y después vuelva a esta página.
- Si se trata de una instancia en las instalaciones de Windows Server, siga las instrucciones de [Instale el CodeDeploy agente para Windows Server](#) y después vuelva a esta página.

## Paso 8: registre la instancia local con CodeDeploy

En las instrucciones de este paso se da por supuesto que va a registrar la instancia on-premises desde la propia instancia on-premises. Puede registrar una instancia local desde un dispositivo o instancia independiente que la tenga AWS CLI instalada y configurada, tal y como se describe en.

### [Paso 5: Instalar y configurar el AWS CLI](#)

AWS CLI Utilícela para registrar la instancia local con el CodeDeploy fin de utilizarla en las implementaciones.

1. Para poder usar el AWS CLI, necesitará el ARN de usuario del usuario de IAM en el que lo creó. [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#) Si aún no tiene el ARN de usuario, llame al comando `get-user`, especificando el nombre del usuario de IAM (con la opción `--user-name`) y consultando únicamente el ARN de usuario (con las opciones `--query` y `--output`):

```
aws iam get-user --user-name CodeDeployUser-OnPrem --query "User.Arn" --output text
```

2. Ejecute el comando [register-on-premises-instance](#) especificando lo siguiente:
  - Un nombre que identifique de manera única la instancia en las instalaciones (con la opción `--instance-name`).

**⚠ Important**

Para ayudarle a identificar la instancia on-premises, especialmente a efectos de depuración, recomendamos encarecidamente que especifique un nombre que se asigne a algunas características exclusivas de la instancia on-premises (por ejemplo, el número de serie o un identificador de recurso interno, si procede). Si especifica una dirección MAC como nombre, tenga en cuenta que las direcciones MAC contienen caracteres CodeDeploy no permitidos, como dos puntos (:). Para obtener una lista de los caracteres permitidos, consulte [CodeDeploy cuotas](#).

- El ARN del usuario de IAM que creó en [Paso 1: Crear un usuario de IAM para la instancia en las instalaciones](#) (con la opción `--iam-user-arn`).

Por ejemplo:

```
aws deploy register-on-premises-instance --instance-name AssetTag12010298EX --iam-user-arn arn:aws:iam::444455556666:user/CodeDeployUser-OnPrem
```

## Paso 9: Etiquetar la instancia en las instalaciones

Puede usar la consola AWS CLI o la CodeDeploy consola para etiquetar la instancia local. (CodeDeploy usa etiquetas de instancias locales para identificar los objetivos de implementación durante una implementación).

Para etiquetar la instancia en las instalaciones (CLI)


- Llama al comando [add-tags-to-on-premises-instances](#) y especifica:
  - El nombre que identifica de manera única la instancia on-premises (con la opción `--instance-names`).
  - El nombre de la clave de etiqueta y el valor de etiqueta de la instancia on-premises que desea utilizar (con la opción `--tags`). Debe especificar un nombre y un valor. CodeDeploy no permite etiquetas de instancias locales que solo tengan valores.

Por ejemplo:


```
aws deploy add-tags-to-on-premises-instances --instance-names AssetTag12010298EX
--tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Para etiquetar la instancia on-premises (consola)

1. Inicia sesión en la CodeDeploy consola AWS Management Console y ábrela en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el CodeDeploy menú, selecciona Instancias locales.
3. En la lista de instancias on-premises, elija la flecha situada junto a la instancia on-premises que desea etiquetar.
4. En la lista de etiquetas, seleccione o introduzca la clave y el valor que desee para la etiqueta. Cuando introduzca la clave y el valor de la etiqueta, aparecerá otra fila. Puede repetir esta acción para un máximo de 10 etiquetas. Para quitar una etiqueta, elija el icono de eliminación ().
5. Después de añadir las etiquetas, seleccione Update Tags.

## Paso 10: Implementar revisiones de aplicaciones en la instancia en las instalaciones

Ahora está listo para implementar revisiones de aplicaciones en la instancia on-premises registrada y etiquetada.

La implementación de revisiones de aplicaciones en instancias en las instalaciones es similar a la implementación de revisiones de aplicaciones en instancias de Amazon EC2. Para ver instrucciones, consulte [Cree una implementación con CodeDeploy](#). Estas instrucciones incluyen un enlace a los requisitos previos, incluida la creación de una aplicación, la creación de un grupo de implementaciones y la preparación de la revisión de una aplicación. Si necesita una revisión de una aplicación sencilla para implementar, puede crear la que se describe en [Paso 2: Crear una revisión de aplicación de ejemplo](#) en el [Tutorial: Implemente una aplicación en una instancia local con CodeDeploy \(Windows Server, Ubuntu Server o Red Hat Enterprise Linux\)](#).

**⚠ Important**

Si reutilizas un rol de CodeDeploy servicio como parte de la creación de un grupo de implementación dirigido a instancias locales, debes `Tag: get*` incluirlo en la `Action` parte de la declaración de política del rol de servicio. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).

## Paso 11: Hacer un seguimiento de las implementaciones en la instancia en las instalaciones

Después de implementar una revisión de una aplicación en instancias en las instalaciones registradas y etiquetadas, puede hacer un seguimiento del progreso de la implementación.

El seguimiento de las implementaciones en instancias en las instalaciones es similar al seguimiento de las implementaciones en instancias de Amazon EC2. Para obtener instrucciones, consulte [Ver detalles CodeDeploy de la implementación](#).

## Administrar las operaciones de instancias locales en CodeDeploy

Sigue las instrucciones de esta sección para gestionar las operaciones en las instancias locales después de haberlas registrado, por ejemplo CodeDeploy, para obtener más información sobre las instancias locales, quitarles las etiquetas, desinstalarlas y anular el registro.

### Temas

- [Obtiene información acerca de una instancia en las instalaciones individual.](#)
- [Obtener información acerca de varias instancias en las instalaciones](#)
- [Eliminar manualmente las etiquetas de instancias en las instalaciones de una instancia](#)
- [Desinstale automáticamente el CodeDeploy agente y elimine el archivo de configuración de una instancia local](#)
- [Anulación automática del registro de una instancia en las instalaciones](#)
- [Anulación manual del registro de una instancia en las instalaciones](#)

### Obtiene información acerca de una instancia en las instalaciones individual.

En las instrucciones de [Ver detalles CodeDeploy de la implementación](#) se describe cómo obtener información acerca de una instancia en las instalaciones individual. Puedes usar la consola AWS CLI o la CodeDeploy consola para obtener más información sobre una única instancia local.

## Para obtener información acerca de una instancia en las instalaciones individual (CLI)

- Llama al [get-on-premises-instance](#) comando y especifica el nombre que identifica de forma exclusiva a la instancia local (con la `--instance-name` opción):

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

## Para obtener información acerca de una instancia en las instalaciones individual (consola)

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Instancias en las instalaciones.
3. En la lista de instancias en las instalaciones, elija el nombre de una instancia en las instalaciones para ver sus detalles.

## Obtener información acerca de varias instancias en las instalaciones

En las instrucciones de [Ver detalles CodeDeploy de la implementación](#) se describe cómo obtener información acerca de instancias en las instalaciones. Puedes usar la consola AWS CLI o la CodeDeploy consola para obtener más información sobre las instancias locales.

## Para obtener información acerca de varias instancias en las instalaciones (CLI)

1. Para obtener una lista de los nombres de las instancias locales, ejecuta el [list-on-premises-instances](#) comando y especifica:
  - Si obtener información acerca de todas instancias on-premises registradas o cuyos registros se hayan anulado, con las opciones `--registration-status` y `Registered` o `Deregistered`, respectivamente. Si omite este paso, se devolverán tanto los nombres de instancias on-premises registrados como aquellos cuyos registros se hayan anulado.
  - Si obtener solo información acerca de las instancias on-premises con etiquetas específicas de instancia on-premises, con la opción `--tag-filters`. Especifique `Key`, `Value` y `Type`,

que deberá ser `KEY_AND_VALUE`, para cada etiqueta de instancia on-premises. Separe las etiquetas de las instancias on-premises con espacios entre cada conjunto de `Key`, `Value` y `Type`.

Por ejemplo:


```
aws deploy list-on-premises-instances --registration-status Registered
--tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
Key=Name,Value=CodeDeployDemo-OnPrem-Beta,Type=KEY_AND_VALUE
```

2. Para obtener información más detallada, ejecuta el comando [batch-get-on-premises-instances](#) con los nombres de las instancias locales (con la `--instance-names` opción):

```
aws deploy batch-get-on-premises-instances --instance-names AssetTag12010298EX
AssetTag09920444EX
```

Para obtener información acerca de varias instancias en las instalaciones (consola)

1. [Inicia sesión en la CodeDeploy consola AWS Management Console y ábrela en https://console.aws.amazon.com/codedeploy.](https://console.aws.amazon.com/codedeploy)

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Instancias en las instalaciones.

Se mostrará Información acerca de las instancias on-premises.

## Eliminar manualmente las etiquetas de instancias en las instalaciones de una instancia

Por lo general, una etiqueta de instancia en las instalaciones se elimina de dicha instancia cuando deja de usarse o en caso de querer eliminar la instancia en las instalaciones del grupo de implementación que la utiliza. Puedes usar la consola AWS CLI o la AWS CodeDeploy consola para eliminar etiquetas de instancias locales de las instancias locales.

No necesita eliminar las etiquetas de instancias on-premises desde una instancia on-premises antes anular su registro.

Eliminar las etiquetas de instancias on-premises de una instancia on-premises no anula su registro. No desinstala el CodeDeploy agente de la instancia. No elimina el archivo de configuración de la instancia, ni desinstala el usuario de IAM asociado a la instancia.

Para anular el registro de una instancia on-premises automáticamente, consulte [Anulación automática del registro de una instancia en las instalaciones](#).

Para anular el registro de una instancia on-premises manualmente, consulte [Anulación manual del registro de una instancia en las instalaciones](#).

Para desinstalar automáticamente el CodeDeploy agente y eliminar el archivo de configuración de la instancia local, consulte [Desinstale automáticamente el CodeDeploy agente y elimine el archivo de configuración de una instancia local](#).

Para desinstalar manualmente solo el CodeDeploy agente de la instancia local, consulte. [Gestión de las operaciones CodeDeploy de los agentes](#)

Para eliminar manualmente el usuario de IAM asociado, consulte [Eliminación de un usuario de IAM de su cuenta de AWS](#).

Para eliminar etiquetas de instancias on-premises de una instancia on-premises (CLI)

- Llame a [remove-tags-from-on-premises-instances](#) y especifique:
  - Los nombres que identifican de manera única las instancias on-premises, con la opción `--instance-names`.
  - Los nombres y valores de las etiquetas que desea quitar, con la opción `--tags`.

Por ejemplo:

```
aws deploy remove-tags-from-on-premises-instances --instance-names
AssetTag12010298EX --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Para eliminar etiquetas de instancias en las instalaciones de una instancia en las instalaciones (consola)

1. [Inicie sesión AWS Management Console y abra la CodeDeploy consola en https://console.aws.amazon.com/codedeploy](https://console.aws.amazon.com/codedeploy).

**Note**

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Instancias en las instalaciones.
3. En la lista de instancias en las instalaciones, elija el nombre de la instancia en las instalaciones cuyas etiquetas desea eliminar.
4. En la sección Tags (Etiquetas), elija Remove (Quitar) junto a cada etiqueta que desee quitar.
5. Después de eliminar las etiquetas, seleccione Update tags.

## Desinstale automáticamente el CodeDeploy agente y elimine el archivo de configuración de una instancia local

Por lo general, se desinstala el CodeDeploy agente y se elimina el archivo de configuración de una instancia local cuando ya no se planea realizar la implementación en ella.

**Note**

La desinstalación automática del CodeDeploy agente y la eliminación del archivo de configuración de una instancia local no anulan el registro de una instancia local. Tampoco desvincula ninguna etiqueta de instancia on-premises asociada a la instancia on-premises, ni elimina el usuario de IAM asociado a la instancia en las instalaciones.

Para anular el registro de una instancia on-premises automáticamente, consulte [Anulación automática del registro de una instancia en las instalaciones](#).

Para anular el registro de una instancia on-premises manualmente, consulte [Anulación manual del registro de una instancia en las instalaciones](#).

Para desvincular etiquetas asociadas a una instancia on-premises manualmente, consulte [Eliminar manualmente las etiquetas de instancias en las instalaciones de una instancia](#).

Para desinstalar manualmente el CodeDeploy agente de la instancia local, consulte [Gestión de las operaciones CodeDeploy de los agentes](#)

Para eliminar manualmente el usuario de IAM asociado, consulte [Eliminación de un usuario de IAM de su cuenta de AWS](#).

Desde la instancia local, utilice el AWS CLI para llamar al comando de [desinstalación](#).



Por ejemplo:

```
aws deploy uninstall
```

El comando `uninstall` hace lo siguiente:

1. Detiene el CodeDeploy agente en ejecución en la instancia local.
2. Desinstala el CodeDeploy agente de la instancia local.
3. Elimina el archivo de configuración de la instancia en las instalaciones. (Para Ubuntu Server y RHEL, es `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml`. Para Windows Server, es `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml`.)

## Anulación automática del registro de una instancia en las instalaciones

Por lo general, si ya no tiene pensado implementar la instancia on-premises, debe anular su registro. Al anular el registro de una instancia on-premises, dicha instancia deja de incluirse en cualquier implementación, incluso si forma parte de las etiquetas de instancias on-premises de un grupo de implementaciones. Puede utilizarla para anular el registro de AWS CLI las instancias locales.

### Note

No puedes usar la CodeDeploy consola para anular el registro de una instancia local. Además, la anulación del registro de una instancia en las instalaciones elimina aquellas etiquetas que estén asociadas a la instancia en las instalaciones. No desinstala el CodeDeploy agente de la instancia local. ni elimina el archivo de configuración de instancia on-premises de dicha instancia.

Para usar la CodeDeploy consola para realizar algunas (pero no todas) de las actividades de esta sección, consulte la sección de CodeDeploy consolas de [Anulación manual del registro de una instancia en las instalaciones](#).

Para desvincular etiquetas asociadas a una instancia on-premises manualmente, consulte [Eliminar manualmente las etiquetas de instancias en las instalaciones de una instancia](#).

Para desinstalar automáticamente el CodeDeploy agente y eliminar el archivo de configuración de la instancia local, consulte [Desinstale automáticamente el CodeDeploy agente y elimine el archivo de configuración de una instancia local](#).

Para desinstalar manualmente solo el CodeDeploy agente de la instancia local, consulte [Gestión de las operaciones CodeDeploy de los agentes](#)

Use el comando `deregister` AWS CLI para llamar al comando [deregister](#) y especifique:

- El nombre que identifica de forma exclusiva a la instancia local CodeDeploy (con la `--instance-name` opción).
- De manera opcional, si desea eliminar el usuario de IAM asociado a la instancia en las instalaciones. El comportamiento predeterminado es eliminar al usuario de IAM. Si no desea eliminar el usuario de IAM asociado a la instancia on-premises, especifique la opción `--no-delete-iam-user` en el comando.
- De forma opcional, la AWS región en la que se registró la instancia local CodeDeploy (con la `--region` opción). Debe ser una de las regiones admitidas que se encuentran en la lista de [Regiones y puntos de enlace](#) de la Referencia general de AWS (como, por ejemplo, `us-west-2`). Si no se especifica esta opción, se utilizará la AWS región predeterminada asociada al usuario de IAM que realiza la llamada.

Ejemplo que anula el registro de una instancia y elimina al usuario:

```
aws deploy deregister --instance-name AssetTag12010298EX --region us-west-2
```

Ejemplo que anula el registro de una instancia y no elimina al usuario:

```
aws deploy deregister --instance-name AssetTag12010298EX --no-delete-iam-user --region us-west-2
```

El comando `deregister` hace lo siguiente:

1. Anula el registro de la instancia local con. CodeDeploy
2. Si se especifica, elimina el usuario de IAM asociado a la instancia en las instalaciones.

Después de cancelar el registro de una instancia en las instalaciones:

- Deja de aparecer en la consola inmediatamente.
- Puede crear otra instancia con el mismo nombre inmediatamente.

Si este comando encuentra algún error, aparece un mensaje en el que se describe cómo realizar manualmente los pasos faltantes. En caso contrario, aparece un mensaje de confirmación en el que se describe cómo realizar una llamada al comando `uninstall`.

## Anulación manual del registro de una instancia en las instalaciones

Por lo general, si ya no tiene pensado implementar la instancia on-premises, debe anular su registro. Se usa para anular el registro manual AWS CLI de las instancias locales.

Al anular manualmente el registro de una instancia local, no se desinstala el agente. CodeDeploy No elimina el archivo de configuración de la instancia, ni desinstala el usuario de IAM asociado a la instancia. ni elimina etiquetas asociadas a la instancia.

Para desinstalar automáticamente el CodeDeploy agente y eliminar el archivo de configuración de la instancia local, consulte. [Desinstale automáticamente el CodeDeploy agente y elimine el archivo de configuración de una instancia local](#)

Para desinstalar manualmente solo el CodeDeploy agente, consulte [Gestión de las operaciones CodeDeploy de los agentes](#).

Para eliminar manualmente el usuario de IAM asociado, consulte [Eliminación de un usuario de IAM de su cuenta de AWS](#).

Para eliminar manualmente solo las etiquetas de instancias en las instalaciones asociadas, consulte [Eliminar manualmente las etiquetas de instancias en las instalaciones de una instancia](#).

- Ejecute el [deregister-on-premises-instance](#) comando y especifique el nombre que identifica de forma exclusiva a la instancia local (con la `--instance-name` opción):

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

Después de cancelar el registro de una instancia en las instalaciones:

- Deja de aparecer en la consola inmediatamente.
- Puede crear otra instancia con el mismo nombre inmediatamente.

## Vea los detalles de la instancia con CodeDeploy

Puede usar la CodeDeploy consola AWS CLI, las API o las CodeDeploy API para ver los detalles sobre las instancias utilizadas en una implementación.

Para obtener información sobre el uso de las acciones de la CodeDeploy API para ver las instancias [GetDeploymentInstance](#), consulte [ListDeploymentInstances](#), y [ListOnPremisesInstances](#).

## Temas

- [Visualización de los detalles de la instancia \(consola\)](#)
- [Visualización de los detalles de la instancia \(CLI\)](#)

## Visualización de los detalles de la instancia (consola)

Para ver los detalles de la instancia:

1. Inicia sesión AWS Management Console y abre la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y después elija Aplicaciones.

### Note

Si no se muestra ninguna entrada, asegúrese de seleccionar la región correcta. En la barra de navegación, en el selector de regiones, elija una de las regiones que aparecen en [Región y puntos finales](#) en. Referencia general de AWS CodeDeploy solo se admite en estas regiones.

3. Para ver los detalles de la implementación, elija el ID de implementación para la instancia.
4. Puede ver todas las instancias en la sección Instance activity (Actividad de instancia) de la página de la implementación.
5. Para ver información acerca de los eventos del ciclo de vida de la implementación individual de una instancia, en la página de detalles de la implementación, en la columna Events, elija View events.

### Note

Si se muestra Failed en cualquiera de los eventos del ciclo de vida, en la página de detalles de la instancia, elija View logs, View in EC2 o ambas opciones. Puede encontrar consejos de solución de problemas en [Solución de problemas de instancias](#).

- Si desea ver más información sobre una instancia de Amazon EC2, elija el ID de la instancia en la columna ID de instancia.

## Visualización de los detalles de la instancia (CLI)

Para usar el AWS CLI para ver los detalles de la instancia, llame al `get-deployment-instance` comando o al `list-deployment-instances` comando.

Para ver los detalles de una sola instancia, llama al [get-deployment-instance](#) comando y especifica:

- El ID de implementación único. Para obtener el ID de implementación, llame al comando [list-deployments](#).
- El ID de instancia único. Para obtener el ID de la instancia, llama al [list-deployment-instances](#) comando.

Para ver una lista de los ID de las instancias utilizadas en una implementación, llama al [list-deployment-instances](#) comando y especifica:

- El ID de implementación único. Para obtener el ID de implementación, llame al comando [list-deployments](#).
- Si lo prefiere, puede incluir únicamente los ID de instancia específicos por su estado de implementación. (Si no se especifican, se enumerarán todos los ID de instancia coincidentes, independientemente de su estado de implementación).

## CodeDeploy estado de la instancia

CodeDeploy supervisa el estado de las instancias de un grupo de implementación. Produce un error en las implementaciones si el número de instancias en buen estado es inferior al número de instancias en buen estado que se han especificado para el grupo de implementaciones durante una implementación. Por ejemplo, si el 85% de las instancias deben estar en buen estado durante una implementación y el grupo de implementación contiene 10 instancias, toda la implementación dará un error aunque solo una única instancia produzca un error. Esto se debe a que cuando se desconecta una instancia para que se pueda instalar la última revisión de la aplicación, el número de instancias en buen estado ya está al 90%. Una instancia fallida y otra fuera de línea significaría que solo el 80% de las instancias están en buen estado y disponibles. CodeDeploy fallará en la implementación general.

Es importante recordar que debe cumplirse lo siguiente para que la implementación global tenga éxito:

- CodeDeploy puede realizar la implementación en cada instancia de la implementación.
- La implementación debe tener éxito en una instancia como mínimo. Esto significa que, aunque el valor mínimo de hosts en buen estado sea 0, la implementación debe realizarse correctamente en una instancia como mínimo (es decir, al menos una instancia debe estar en buen estado) para que la implementación global se realice correctamente.

## Temas

- [Estado](#)
- [Acerca del número mínimo de instancias en buen estado](#)
- [Acerca del número mínimo de instancias en buen estado por zona de disponibilidad](#)

## Estado

CodeDeploy asigna dos valores de estado a cada instancia: el estado de revisión y el estado de la instancia.

### Estado de revisión

El estado de revisión se basa en la revisión de aplicación instalada actualmente en la instancia. Tiene los siguientes valores de estado:

- **Current:** la revisión instalada en la instancia coincide con la revisión de la última implementación correcta del grupo de implementación.
- **Old:** la revisión instalada en la instancia coincide con una versión anterior de la aplicación.
- **Unknown:** la revisión de la aplicación no se ha instalado correctamente en la instancia.

### Estado de la instancia

El estado de la instancia se basa en si las implementaciones en una instancia se han realizado correctamente. Tiene los siguientes valores:

- **Healthy:** la última implementación en la instancia se realizó correctamente.
- **Unhealthy:** el intento de implementar una revisión en la instancia dio un error o aún no se ha implementado una revisión en la instancia.

CodeDeploy utiliza el estado de las revisiones y el estado de las instancias para programar el despliegue en las instancias del grupo de implementación en el siguiente orden:

1. Estado de la instancia Unhealthy.
2. Estado de revisión Unknown.
3. Estado de revisión Old.
4. Estado de revisión Current.

Si toda la implementación se realiza correctamente, la revisión se actualiza y los valores de estado del grupo de implementaciones se actualizan para reflejar la última implementación.

- Todas las instancias actuales que tenían una implementación satisfactoria siguen vigentes. De lo contrario, pasan a tener el estado Unknown.
- Todas las instancias Old o Unknown que tenían una implementación satisfactoria siguen vigentes. De lo contrario, pasan a tener el estado Old o Unknown.
- Todas las instancias Healthy que tenían una implementación satisfactoria siguen en buen estado. De lo contrario, pasan a tener el estado Unhealthy.
- Todas las instancias Unhealthy que tenían una implementación satisfactoria pasan a tener el estado Healthy. De lo contrario, permanecen con el estado Unhealthy.

Si se produce un error en toda la implementación o si se detiene la implementación:

- Cada instancia en la que se CodeDeploy intentó implementar la revisión de la aplicación tiene el estado de la instancia establecido en buen estado o en mal estado, dependiendo de si el intento de implementación de esa instancia se realizó correctamente o no.
- Cada instancia en la que CodeDeploy no se intentó implementar la revisión de la aplicación conserva su valor de estado actual.
- La revisión del grupo de implementación sigue siendo la misma.

## Acerca del número mínimo de instancias en buen estado

El número mínimo de instancias en buen estado necesarias se define como parte de la configuración de una implementación.

**⚠ Important**

Durante una implementación blue/green, la configuración de la implementación y el valor mínimo de hosts en buen estado se aplican a las instancias del entorno de sustitución y no a las del entorno original. Sin embargo, cuando se cancela el registro de las instancias del entorno original en el balanceador de carga, toda la implementación se marca como errónea aunque solo haya una instancia original cuyo registro no se haya podido cancelar correctamente.

CodeDeploy proporciona tres configuraciones de implementación predeterminadas que suelen utilizar valores mínimos de host en buen estado:

| Nombre de configuración de la implementación predeterminada | Valor mínimo de hosts en buen estado predefinido |
|-------------------------------------------------------------|--------------------------------------------------|
| CodeDeployDefault.OneAtATime                                | 1                                                |
| CodeDeployDefault.HalfAtATime                               | 50%                                              |
| CodeDeployDefault.AllAtOnce                                 | 0                                                |

Encontrará más información acerca de las configuraciones de implementación predeterminadas en [Trabajar con configuraciones de implementación en CodeDeploy](#).

Puede crear configuraciones de implementación personalizadas CodeDeploy para definir sus propios valores de host mínimos en buen estado. Puede definir estos valores como números enteros o como porcentajes mediante las siguientes operaciones:

- Como `minimum-healthy-hosts` cuando usa el [create-deployment-config](#) comando en AWS CLI.
- Como `Value` en el tipo de [MinimumHealthyHosts](#) datos de la CodeDeploy API.
- Como `MinimumHealthyHosts` cuando se usa [AWS::CodeDeploy::DeploymentConfig](#) en una AWS CloudFormation plantilla.

CodeDeploy permite especificar un número mínimo de instancias en buen estado para la implementación con dos finalidades principales:



- Para determinar si la implementación global se realizó correctamente o dio un error. La implementación se realiza correctamente si la revisión de la aplicación se implementó satisfactoriamente en al menos el número mínimo de instancias en buen estado.
- Para determinar el número de instancias que deben estar en buen estado durante una implementación para que la implementación se pueda realizar correctamente.

Puede especificar el número mínimo de instancias en buen estado para su grupo de implementaciones como un número de instancias o como un porcentaje del total de instancias. Si especificas un porcentaje, al inicio de la implementación, CodeDeploy convierte el porcentaje en el número equivalente de instancias y redondea las instancias fraccionarias.

CodeDeploy realiza un seguimiento del estado de las instancias del grupo de implementación durante el proceso de implementación y utiliza la cantidad mínima especificada de instancias en buen estado para la implementación para determinar si se debe continuar con la implementación. El principio básico es que una implementación no debe provocar nunca que el número de instancias en buen estado sea inferior al número mínimo especificado. La única excepción a esta regla es cuando un grupo de implementaciones tiene inicialmente menos instancias en buen estado que el número mínimo especificado. En ese caso, el proceso de implementación no reduce el número de instancias en buen estado.

#### Note

CodeDeploy intentará implementarse en todas las instancias de un grupo de implementación, incluso en las que se encuentran actualmente en estado de parada. En el cálculo de hosts mínimos en buen estado, una instancia detenida tiene el mismo efecto que una instancia errónea. Para resolver los errores de implementación debido a demasiadas instancias detenidas, reinicie las instancias o cambie sus etiquetas para excluirlas del grupo de implementaciones.

CodeDeploy inicia el proceso de despliegue intentando implementar la revisión de la aplicación en las instancias en mal estado del grupo de despliegue. Por cada implementación correcta, CodeDeploy cambia el estado de la instancia a correcto y la agrega a las instancias en buen estado del grupo de implementación. CodeDeploy a continuación, compara el número actual de instancias en buen estado con el número mínimo especificado de instancias en buen estado.

- Si la cantidad de instancias en buen estado es inferior o igual a la cantidad mínima especificada de instancias en buen estado, CodeDeploy cancela la implementación para garantizar que la cantidad de instancias en buen estado no disminuya con más implementaciones.
- Si el número de instancias en buen estado es superior en al menos una al número mínimo especificado de instancias en buen estado, CodeDeploy implementa la revisión de la aplicación en el conjunto original de instancias en buen estado.

Si se produce un error en la implementación de una instancia en buen estado, CodeDeploy cambia el estado de esa instancia a en mal estado. A medida que avanza la implementación, CodeDeploy actualiza el número actual de instancias en buen estado y lo compara con el número mínimo especificado de instancias en buen estado. Si el número de instancias en buen estado cae hasta el número mínimo especificado en algún momento del proceso de implementación, CodeDeploy se detiene la implementación. De este modo, se impide la posibilidad de que la próxima implementación produzca un error, haciendo que el número de instancias en buen estado sea inferior al número mínimo especificado.

#### Note

Asegúrese de que el número mínimo de instancias en buen estado que especifique sea inferior al número total de instancias del grupo de implementaciones. Si especifica un valor de porcentaje, recuerde que se redondeará. De lo contrario, cuando comience la implementación, la cantidad de instancias en buen estado ya será inferior o igual a la cantidad mínima especificada de instancias en buen estado y, de forma inmediata, CodeDeploy se producirá un error en toda la implementación.

CodeDeploy también utiliza el número mínimo especificado de instancias en buen estado y el número real de instancias en buen estado para determinar si se debe implementar la revisión de la aplicación en varias instancias y cómo hacerlo. De forma predeterminada, CodeDeploy implementa la revisión de la aplicación en tantas instancias como sea posible sin riesgo de que el número de instancias en buen estado caiga por debajo del número mínimo especificado de instancias en buen estado.

Para determinar el número de instancias que se deben implementar a la vez, CodeDeploy utilice el siguiente cálculo:

$$[\text{total-hosts}] - [\text{minimum-healthy-hosts}] =$$

```
[number-of-hosts-to-deploy-to-at-once]
```

Por ejemplo:

- Si su grupo de implementación tiene 10 instancias y ha establecido el número mínimo de instancias en buen estado en 9, se CodeDeploy implementa una instancia a la vez.
- Si su grupo de implementación tiene 10 instancias y ha establecido el número mínimo de instancias en buen estado en 3, se CodeDeploy despliega en 7 instancias al mismo tiempo en el primer lote y, después, en las 3 restantes en el segundo lote.
- Si el grupo de implementación tiene 10 instancias y estableces el número mínimo de instancias en buen estado en 0, se CodeDeploy despliega en 10 instancias al mismo tiempo.

## Ejemplos

En los siguientes ejemplos se presupone que existe un grupo de implementaciones con 10 instancias.

Mínimo de instancias en buen estado: 95%

CodeDeploy redondea el número mínimo de instancias en buen estado hasta 10 instancias, lo que equivale al número de instancias en buen estado. Toda la implementación produce inmediatamente un error sin que se implemente la revisión en ninguna de las instancias.

Mínimo de instancias en buen estado: 9

CodeDeploy implementa la revisión en una instancia a la vez. Si la implementación en alguna de las instancias falla, se interrumpe CodeDeploy inmediatamente la implementación general, ya que el número de instancias en buen estado es igual al número mínimo de instancias en buen estado. La excepción a esta regla es que si se produce un error en la última instancia, la implementación se realiza correctamente.

CodeDeploy continúa la implementación, instancia por vez, hasta que alguna implementación falle o se complete la implementación general. Si las 10 implementaciones se realizan correctamente, el grupo de implementación tendrá ahora 10 instancias en buen estado.

Mínimo de instancias en buen estado: 8

CodeDeploy implementa la revisión en dos instancias a la vez. Si dos de estas implementaciones fallan, la implementación general fracasa CodeDeploy inmediatamente. La excepción a esta regla

es que si la última instancia es la segunda instancia que da error, la implementación se realiza correctamente.

Mínimo de instancias en buen estado: 0

CodeDeploy implementa la revisión en todo el grupo de implementaciones a la vez. La implementación debe realizarse correctamente al menos en una instancia para que la implementación global se realice correctamente. Si hay 0 instancias en buen estado, la implementación no se realiza correctamente. Esto se debe al requisito de que, para considerar que una implementación global se ha realizado correctamente, al menos una instancia debe estar en buen estado cuando finalice la implementación global, incluso si el valor mínimo de instancias en buen estado es 0.

## Acerca del número mínimo de instancias en buen estado por zona de disponibilidad

### Note

En esta sección se utilizan los términos instancia y host indistintamente para hacer referencia a las instancias de Amazon EC2.

Si va a realizar la implementación en instancias de varias [zonas de disponibilidad](#), de forma opcional, puede habilitar la [zonal configuration](#) función, que CodeDeploy permite la implementación en una zona de disponibilidad a la vez.

Cuando esta función esté habilitada, se CodeDeploy asegurará de que la cantidad de hosts en buen estado se mantenga por encima de los valores de «hosts en buen estado mínimos por zona» y «hosts en buen estado». Si el número de hosts en buen estado cae por debajo de cualquiera de estos valores, CodeDeploy no se podrá realizar la implementación en todas las zonas de disponibilidad.

Para calcular la cantidad de hosts en los que realizar la implementación a la vez, CodeDeploy utiliza los valores de «número mínimo de hosts en buen estado por zona» y «número mínimo de hosts en buen estado». CodeDeploy utilizará el menor de los cálculos [A] y [B], donde y dónde están: [A]  
[B]

$$[A] = [\text{total-hosts}] - [\text{min-healthy-hosts}] =$$

$$[\text{number-of-hosts-to-deploy-to-at-once}]$$
$$[B] = [\text{total-hosts-per-AZ}] - [\text{min-healthy-hosts-per-AZ}] =$$
$$[\text{number-of-hosts-to-deploy-to-at-once-per-AZ}]$$

Tras determinar el número de hosts en los que realizar el despliegue a la vez, se CodeDeploy despliega en los hosts en lotes de ese número, una zona de disponibilidad a la vez, con una pausa opcional (o «hora de cocción») entre las zonas.

## Ejemplo

Si su implementación está configurada de la siguiente manera:

- `[total-hosts]` es 200
- `[minimum-healthy-hosts]` es 160
- `[total-hosts-per-AZ]` es 100
- `[minimum-healthy-hosts-per-AZ]` es 50

Entonces...

- $[A] = 200 - 160 = 40$
- $[B] = 100 - 50 = 50$
- 40 es menor que 50

Por lo tanto, CodeDeploy se implementará en los hosts a la vez. 40

En este caso, la implementación se desglosa de la siguiente manera:

1. CodeDeploy se despliega en la primera zona de disponibilidad:
  - a. CodeDeploy se despliega en los primeros 40 hosts.
  - b. CodeDeploy se despliega en los siguientes 40 hosts.
  - c. CodeDeploy se despliega en los hosts restantes 20.

La implementación en la primera zona de disponibilidad ya está completa.

2. (Opcional) CodeDeploy espera mientras se «hornea» el despliegue en la primera zona, según se define en la configuración de duración del monitor o Añadir una duración del monitor para la primera zona. Si no hay problemas, CodeDeploy continúa.

3. CodeDeploy se despliega en la segunda zona de disponibilidad:
  - a. CodeDeploy se despliega en los primeros 40 hosts.
  - b. CodeDeploy se despliega en los siguientes 40 hosts.
  - c. CodeDeploy se despliega en los hosts restantes 20.

La implementación en la segunda y final zona de disponibilidad ya está completa.

Para obtener información sobre la característica de configuración de zonas y cómo especificar el número mínimo de instancias en buen estado por zona de disponibilidad, consulte [zonal configuration](#).

# Trabajar con configuraciones de implementación en CodeDeploy

Una configuración de despliegue es un conjunto de reglas y condiciones de éxito y fracaso que se utilizan CodeDeploy durante un despliegue. Estas reglas y condiciones son diferentes en función de si realiza la implementación en una plataforma de informática EC2/en las instalaciones, una plataforma de informática AWS Lambda o una plataforma de informática Amazon ECS.

## Configuraciones de implementación en una plataforma de informática EC2/en las instalaciones

Cuando se realiza la implementación en una plataforma de computación EC2/en las instalaciones, la configuración de implementación específica, a través de un valor mínimo de hosts en buen estado, el número o porcentaje de instancias que deben estar disponibles en cualquier momento durante una implementación.

Puede utilizar una de las tres configuraciones de despliegue predefinidas que proporciona AWS o crear una configuración de despliegue personalizada. Para obtener más información sobre la creación de configuraciones de implementación personalizadas, consulte [Create a Deployment Configuration](#). Si no especifica una configuración de despliegue, CodeDeploy utiliza la CodeDeployDefault.OneAtConfiguration de despliegue aTime.

Para obtener más información sobre cómo CodeDeploy monitorea y evalúa el estado de la instancia durante una implementación, consulte [Instance Health](#). Para ver una lista de las configuraciones de despliegue que ya están registradas en su AWS cuenta, consulte [View Deployment Configuration Details](#).

## Configuraciones de implementación predefinidas para una plataforma de informática EC2/en las instalaciones

La siguiente tabla muestra las configuraciones de implementación predefinidas.

### Note


No hay configuraciones de implementación predefinidas que admitan la característica [zonal configuration](#) (que es la función que le permite especificar la cantidad de hosts en buen

estado por zona de disponibilidad). Si desea utilizar esta característica, debe [crear su propia configuración de implementación](#).

| Configuración de implementación | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.AllAtOnce     | <p>Implementaciones in situ:</p> <p>Intenta implementar una revisión de aplicación en tantas instancias a la vez como sea posible. El estado de la implementación global se muestra como Realizado correctamente cuando la revisión de la aplicación se haya implementado en una o más instancias. El estado de la implementación global se muestra como Error cuando la revisión de la aplicación no se ha implementado en ninguna instancia. Utilizando un ejemplo de nueve instancias, CodeDeployDefault.AllAtOnce intenta realizar la implementación en las nueve instancias a la vez. La implementación global tiene éxito si se realiza de forma correcta incluso la implementación en una sola instancia. Devuelve un error solo si se produce un error en las implementaciones de las nueve instancias.</p> <p>Implementaciones azul/verde:</p> <ul style="list-style-type: none"><li>• Implementación en un entorno de reemplazo : sigue las mismas reglas de implementación que CodeDeployDefault.AllAtOnce para despliegues in situ.</li><li>• Redirección del tráfico: dirige el tráfico a todas las instancias del entorno de sustitución a la vez. Tiene éxito si el tráfico se dirige correctamente a al menos una instancia.</li></ul> |



| Configuración de implementación | Descripción                                                            |
|---------------------------------|------------------------------------------------------------------------|
|                                 | Fracasa si la redirección del tráfico falla para todas las instancias. |

| Configuración de implementación | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault. HalfAtA Time | <p>Implementaciones in situ:</p> <p>Implementa hasta en la mitad de las instancias a la vez (redondeando a la baja). La implementación global tiene éxito si la revisión de la aplicación se implementa en al menos la mitad de las instancias (redondeando al alza). De lo contrario, la implementación produce un error. En el ejemplo de nueve instancias, se implementa hasta en cuatro instancias a la vez. La implementación global tiene éxito si la implementación es correcta en cinco o más instancias. De lo contrario, la implementación produce un error.</p> <div data-bbox="829 905 1507 1696"><p> <b>Note</b></p><p>Si está implementando en instancias de varios grupos de Auto Scaling, CodeDeploy implementará hasta la mitad de las instancias a la vez, independientemente del grupo de Auto Scaling en el que se encuentren. Por ejemplo, supongamos que tiene dos grupos de escalado automático, ASG1 y ASG2, cada uno con 10 instancias. En este escenario, CodeDeploy podría implementarse en 10 instancias en tan solo un momento ASG1 y considerarlo un éxito, ya que se ha implementado en al menos la mitad de las instancias.</p></div> <p>Implementaciones azul/verde:</p> |

| Configuración de implementación | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | <ul style="list-style-type: none"><li>• Implementación en un entorno de reemplazo : sigue las mismas reglas de implementación que CodeDeployDefault. HalfAtA Time para despliegues in situ.</li><li>• Redirección del tráfico: dirige el tráfico hacia la mitad de las instancias del entorno de sustitución a la vez, como máximo. Tiene éxito si la redirección es correcta para al menos la mitad de las instancias. De lo contrario, da un error.</li></ul> |

| Configuración de implementación | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.OneAtATime    | <p>Implementaciones in situ:</p> <p>Implementa la revisión de la aplicación en solo una instancia cada vez.</p> <p>En el caso de grupos de implementaciones que contengan más de una instancia:</p> <ul style="list-style-type: none"><li>• La implementación global tiene éxito si la revisión de la aplicación se implementa en todas las instancias. La excepción a esta regla se da cuando falla la implementación en la última instancia: la implementación global sigue teniendo éxito. Esto se debe a CodeDeploy que solo permite desconectar una instancia a la CodeDeployDefault vez con. OneAtUna configuración de tiempo.</li><li>• La implementación global fracasa si la implementación de la revisión de la aplicación falla en cualquier instancia, salvo la última.</li><li>• En el ejemplo con nueve instancias, la implementación se hace en una instancia cada vez. La implementación global tiene éxito si la implementación en las primeras ocho instancias se realiza de forma correcta. La implementación global genera un error si la implementación en alguna de las primeras ocho instancias devuelve un error.</li></ul> <p>En el caso de grupos de implementaciones que contengan una única instancia, la implementación global solo tiene éxito si la implementación en la única instancia es correcta.</p> <p>Implementaciones azul/verde:</p> |

| Configuración de implementación | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | <ul style="list-style-type: none"> <li>Implementación en un entorno de reemplazo : sigue las mismas reglas de implementación que CodeDeployDefault.OneAtATime para despliegues in situ.</li> <li>Redirección del tráfico: dirige el tráfico a una instancia del entorno de sustitución cada vez. Tiene éxito si el tráfico se redirige correctamente a todas las instancias de sustitución. Fracasa en cuanto la redirección falla. La excepción a esta regla se da cuando no puede registrarse la última instancia: la implementación global sigue teniendo éxito.</li> </ul> |

## Configuraciones de implementación en una plataforma de informática de Amazon ECS

Cuando la implementación se realiza en una plataforma de informática de Amazon ECS, la configuración de implementación especifica cómo se desvía el tráfico al conjunto de tareas actualizado de Amazon ECS. Puede cambiar el tráfico mediante una configuración canaria, lineal o all-at-once despliegue. Para obtener más información, consulte [Configuración de implementación](#).

También puede crear su propia configuración de implementación "canary" o "linear". Para obtener más información, consulte [Create a Deployment Configuration](#).

## Configuraciones de implementación predefinidas para una plataforma de informática de Amazon ECS

La siguiente tabla muestra las configuraciones predefinidas disponibles para implementaciones de Amazon ECS.

### Note

Cuando se utiliza un Equilibrador de carga de red, solo se admite la configuración de implementación predefinida `CodeDeployDefault.ECSAllAtOnce`.

| Configuración de implementación                         | Descripción                                                                                                                   |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.ECS Linear 10 PercentEvery (1 minuto) | Desvía el 10 por ciento del tráfico cada minuto hasta que se desvía todo el tráfico.                                          |
| CodeDeployDefault.ECS Linear 10 3 minutos PercentEvery  | Desvía el 10 por ciento del tráfico cada tres minutos hasta que se desvía todo el tráfico.                                    |
| CodeDeployDefault.ECS Canary 10 por ciento 5 minutos    | Desvía el 10 por ciento del tráfico en el primer incremento. El 90 por ciento restante se implementa cinco minutos más tarde. |
| CodeDeployDefault.ECSCANARY 10 por ciento 15 minutos    | Desvía el 10 por ciento del tráfico en el primer incremento. El 90 por ciento restante se implementa 15 minutos más tarde.    |
| CodeDeployDefault.ECS AllAtOnce                         | Desvía todo el tráfico al contenedor de Amazon ECS actualizado a la vez.                                                      |

## Configuraciones de implementación para implementaciones azul/verde de AWS CloudFormation (Amazon ECS)

Cuando se implementa en una plataforma informática de Amazon ECS mediante implementaciones AWS CloudFormation azul/verde, la configuración de la implementación especifica cómo se traslada el tráfico al contenedor de Amazon ECS actualizado. Puede transferir el tráfico mediante una configuración canaria, lineal o all-at-once de implementación. Para obtener más información, consulte [Configuración de implementación](#).

Con las implementaciones AWS CloudFormation azules o verdes, no puede crear su propia configuración de implementación lineal o lineal personalizada. Para step-by-step obtener instrucciones sobre cómo administrar las implementaciones azul/verde de Amazon ECS, consulte [Automatizar las implementaciones azul/verde de ECS mediante CodeDeploy el uso AWS CloudFormation en la Guía del usuario](#). AWS CloudFormationAWS CloudFormation

**Note**

La administración de las implementaciones azul/verde de Amazon ECS no AWS CloudFormation está disponible en las regiones de Europa (Milán), África (Ciudad del Cabo) y Asia Pacífico (Osaka).

## Configuraciones de implementación en una plataforma de informática de AWS Lambda

Cuando se implementa en una plataforma de procesamiento AWS Lambda, la configuración de implementación especifica la forma en que el tráfico se traslada a las nuevas versiones de la función Lambda de la aplicación. Puede transferir el tráfico mediante una configuración canaria, lineal o de all-at-oncedespliegue. Para obtener más información, consulte [Configuración de implementación](#).

También puede crear su propia configuración de implementación "canary" o "linear". Para obtener más información, consulte [Create a Deployment Configuration](#).

## Configuraciones de implementación predefinidas para una plataforma de AWS Lambda cómputo

La siguiente tabla muestra las configuraciones predefinidas disponibles para implementaciones de AWS Lambda .

| Configuración de implementación                         | Descripción                                                                                                                   |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.LambdaCanary10 por ciento, 5 minutos  | Desvía el 10 por ciento del tráfico en el primer incremento. El 90 por ciento restante se implementa cinco minutos más tarde. |
| CodeDeployDefault.LambdaCanary10 por ciento, 10 minutos | Desvía el 10 por ciento del tráfico en el primer incremento. El 90 por ciento restante se implementa 10 minutos más tarde.    |
| CodeDeployDefault.LambdaCanary10 por ciento 15 minutos  | Desvía el 10 por ciento del tráfico en el primer incremento. El 90 por ciento restante se implementa 15 minutos más tarde.    |

| Configuración de implementación                          | Descripción                                                                                                                |
|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| CodeDeployDefault.LambdaCanary10 por ciento 30 minutos   | Desvía el 10 por ciento del tráfico en el primer incremento. El 90 por ciento restante se implementa 30 minutos más tarde. |
| CodeDeployDefault.LambdaLinear10, PercentEvery 1 minuto  | Desvía el 10 por ciento del tráfico cada minuto hasta que se desvía todo el tráfico.                                       |
| CodeDeployDefault.LambdaLinear10 (PercentEvery2 minutos) | Desvía el 10 por ciento del tráfico cada dos minutos hasta que se desvía todo el tráfico.                                  |
| CodeDeployDefault.LambdaLinear10 PercentEvery 3 minutos  | Desvía el 10 por ciento del tráfico cada tres minutos hasta que se desvía todo el tráfico.                                 |
| CodeDeployDefault.LambdaLinear10 PercentEvery 10 minutos | Desvía el 10 por ciento del tráfico cada 10 minutos hasta que se desvía todo el tráfico.                                   |
| CodeDeployDefault.LambdaAllAtOnce                        | Desvía todo el tráfico a las funciones Lambda actualizadas a la vez.                                                       |

## Temas

- [Create a Deployment Configuration](#)
- [View Deployment Configuration Details](#)
- [Delete a Deployment Configuration](#)

## Cree una configuración de despliegue con CodeDeploy

Si no desea utilizar una de las configuraciones de despliegue predeterminadas incluidas CodeDeploy, puede crear la suya propia siguiendo las instrucciones siguientes.

Puede usar la CodeDeploy consola AWS CLI, las CodeDeploy API o una AWS CloudFormation plantilla para crear configuraciones de implementación personalizadas.

Para obtener información sobre el uso AWS CloudFormation de una plantilla para crear una configuración de despliegue, consulte [AWS CloudFormation plantillas de CodeDeploy referencia](#).



## Temas

- [Creación de una configuración de implementación \(consola\)](#)
- [Crear una configuración de despliegue con CodeDeploy \(AWS CLI\)](#)

## Creación de una configuración de implementación (consola)

Siga estas instrucciones para crear una configuración de implementación con la consola AWS .

Para crear una configuración de despliegue CodeDeploy mediante la consola

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, elija Configuraciones de implementación.

Aparece una lista de las configuraciones de implementación integradas.

3. Elija Create deployment configuration (Crear configuración de implementación).
4. En Nombre de la configuración de implementación, introduzca un nombre para la configuración. Por ejemplo, **my-deployment-config**.
5. En Plataforma de computación, elija una de las siguientes opciones:
  - EC2/En las instalaciones
  - AWS Lambda
  - Amazon ECS
6. Realice una de las siguientes acciones siguientes:
  - Si ha elegido EC2/En las instalaciones:
    1. En Hosts mínimos en buen estado, especifique el número o porcentaje de instancias que deben permanecer disponibles en cualquier momento durante una implementación. Para obtener más información sobre cómo CodeDeploy monitorea y evalúa el estado de la instancia durante una implementación, consulte [Instance Health](#).

2. (Opcional) En Configuración zonal, seleccione Habilitar la configuración zonal para CodeDeploy implementar la aplicación en una [zona de disponibilidad](#) a la vez, dentro de una región. AWS Si se implementa en una zona de disponibilidad a la vez, puede exponer su implementación a un público cada vez mayor, a medida que aumenta la confianza en el rendimiento y la viabilidad de la implementación. Si no habilita una configuración zonal, CodeDeploy implementa la aplicación en una selección aleatoria de hosts de una región.

Si habilita la característica de configuración de zona, tenga en cuenta lo siguiente:

- La característica de configuración de zona solo es compatible con las implementaciones locales en las instancias de Amazon EC2. (No se admiten implementaciones azul/verde e instancias en las instalaciones). Para obtener más información acerca de las implementaciones locales, consulte [Tipo de implementación](#).
  - La característica de configuración de zona no es compatible con las [configuraciones de implementación predefinidas](#). Para usar una configuración de zona, debe crear una configuración de implementación personalizada, tal y como se describe aquí.
  - Si CodeDeploy necesita revertir una implementación, CodeDeploy realizará las operaciones de reversión en hosts aleatorios. (no CodeDeploy revertirá una zona a la vez, como era de esperar). Se ha elegido este comportamiento de reversión por motivos de rendimiento. Para obtener más información acerca de las reversiones, consulte [Vuelva a implementar y revertir una implementación con CodeDeploy](#).
3. Si ha seleccionado la casilla Habilitar configuración de zona, especifique opcionalmente las siguientes opciones:
    - (Opcional) En Supervisar la duración, especifique el período de tiempo, en segundos, que CodeDeploy debe esperar después de completar el despliegue en una zona de disponibilidad. CodeDeploy esperará este tiempo antes de iniciar un despliegue en la siguiente zona de disponibilidad. Plantéese la posibilidad de añadir una duración de monitor para que la implementación tenga tiempo de demostrar su eficacia (o “incorporación”) en una zona de disponibilidad antes de lanzarse en la siguiente zona. Si no especifica la duración del monitor, CodeDeploy comenzará la implementación en la siguiente zona de disponibilidad de forma inmediata. Para obtener más información sobre cómo funciona la configuración de Duración de monitor, consulte [Acerca del número mínimo de instancias en buen estado por zona de disponibilidad](#).
    - (Opcional) Seleccione Añadir una duración de monitor para la primera zona para establecer una duración de monitor que solo se aplique a la primera zona de disponibilidad. Puede configurar esta opción si quiere permitir más tiempo de

incorporación para la primera zona de disponibilidad. Si no especifica un valor en la duración del monitor de la primera zona, CodeDeploy utiliza el valor de duración del monitor para la primera zona de disponibilidad.

- (Opcional) En Hosts mínimos en buen estado por zona, especifique el número o porcentaje de instancias que deben permanecer disponibles por cada zona de disponibilidad durante una implementación. Elija `FLEET_PERCENT` para especificar un porcentaje o `HOST_COUNT` para especificar un número. Este campo funciona junto con el campo Hosts mínimos en buen estado. Para obtener más información, consulte [Acerca del número mínimo de instancias en buen estado por zona de disponibilidad](#).

Si no especificas un valor en el número mínimo de hosts en buen estado por zona, CodeDeploy utiliza un valor predeterminado de 0 porcentaje.

- Si ha elegido AWS Lambda o Amazon ECS:
  1. Para Tipo, elija Linear o Canary.
  2. En los campos Paso e Intervalo, realice una de las siguientes acciones:
    - Si elige Canary, para Paso, indique un porcentaje de tráfico que se debe desviar, comprendido entre 1 y 99. Este es el porcentaje de tráfico que se desvía en el primer incremento. El tráfico restante se desvía después del intervalo seleccionado en el segundo incremento.

En Intervalo, introduzca el número de minutos entre el primer y el segundo desvío de tráfico.

- Si elige Linear, para Paso, indique un porcentaje de tráfico que se debe desviar, comprendido entre 1 y 99. Este es el porcentaje de tráfico que se desvía al comienzo de cada intervalo.

En Intervalo, introduzca el número de minutos entre cada desvío incremental.

## 7. Elija Create deployment configuration (Crear configuración de implementación).

Ahora tiene una configuración de implementación que puede asociar a un grupo de implementación.

## Crear una configuración de despliegue con CodeDeploy (AWS CLI)

Para usar el AWS CLI para crear una configuración de despliegue, ejecute el [create-deployment-config](#) comando.

En el ejemplo siguiente se crea una configuración de implementación de EC2/en las instalaciones denominada `ThreeQuartersHealthy` que requiere que un 75 % de las instancias de destino se encuentren en buen estado durante la implementación:

```
aws deploy create-deployment-config --deployment-config-name ThreeQuartersHealthy --
minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

En el ejemplo siguiente se crea una configuración de implementación de EC2/en las instalaciones denominada `300Total150PerAZ` que requiere que 300 instancias de destino se encuentren en buen estado en total por implementación y que 50 se encuentren en buen estado por zona de disponibilidad. También establece una duración de monitor de 1 hora.

```
aws deploy create-deployment-config --deployment-config-name 300Total150PerAZ
--minimum-healthy-hosts type=HOST_COUNT,value=300 --zonal-config
'{"monitorDurationInSeconds":3600,"minimumHealthyHostsPerZone":
{"type":"HOST_COUNT","value":50}}'
```

En el siguiente ejemplo, se crea una configuración de despliegue de AWS Lambda denominada `Canary25Percent45Minutes`. Utiliza el cambio de tráfico de la versión Canary para desviar el 25 por ciento del tráfico en el primer incremento. El 75 por ciento restante se desviará 45 minutos más tarde:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
--traffic-routing-config
"type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform Lambda
```

En el ejemplo siguiente se crea una configuración de implementación Amazon ECS denominada `Canary25Percent45Minutes`. Utiliza el cambio de tráfico de la versión Canary para desviar el 25 por ciento del tráfico en el primer incremento. El 75 por ciento restante se desviará 45 minutos más tarde:

```
aws deploy create-deployment-config --deployment-config-name Canary25Percent45Minutes
--traffic-routing-config
"type="TimeBasedCanary",timeBasedCanary={canaryPercentage=25,canaryInterval=45}" --
compute-platform ECS
```

# Vea los detalles de la configuración de la implementación con CodeDeploy

Puede usar la CodeDeploy consola, las API o las CodeDeploy API para ver los detalles sobre las configuraciones de implementación asociadas a su AWS cuenta. AWS CLI Para obtener descripciones de las configuraciones de CodeDeploy implementación predefinidas, consulte [Configuraciones de implementación predefinidas para una plataforma de informática EC2/en las instalaciones](#).

## Temas

- [Consulta de los detalles de las configuraciones de implementación \(consola\)](#)
- [Consulta de la configuración de implementación \(CLI\)](#)

## Consulta de los detalles de las configuraciones de implementación (consola)

Para usar la CodeDeploy consola para ver una lista de nombres de configuraciones de implementación:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y después elija Configuraciones de implementación.

Aquí puede ver los nombres de las configuraciones de implementación y los criterios de cada configuración de implementación.

### Note

Si no se muestra ninguna entrada, asegúrese de seleccionar la región correcta. En la barra de navegación, en el selector de regiones, elija una de las regiones que aparecen

en [Región y puntos finales](#) en. Referencia general de AWS CodeDeploy solo se admite en estas regiones.

## Consulta de la configuración de implementación (CLI)

Para usar el AWS CLI para ver los detalles de la configuración de la implementación, llame al `get-deployment-config` comando o al `list-deployment-configs` comando.

Para ver los detalles de una configuración de implementación única, llame al [get-deployment-config](#) comando y especifique el nombre de la configuración de implementación única.

Para ver los detalles de varias configuraciones de implementación, llame al comando [list-deployments](#).

## Elimine una configuración de despliegue con CodeDeploy

Puede usar las API AWS CLI o las CodeDeploy API para eliminar las configuraciones de implementación personalizadas asociadas a su AWS cuenta. No puede eliminar las configuraciones de implementación integradas, como `CodeDeployDefault.AllAtOnce`, `CodeDeployDefault.HalfAtATime` y `CodeDeployDefault.OneAtATime`.

### Warning

No puede eliminar una configuración de implementación personalizadas que todavía está en uso. Si elimina una configuración de implementación personalizada no utilizada, ya no podrá asociarla con nuevas implementaciones y nuevos grupos de implementaciones. Esta acción no se puede deshacer.

Para usar la AWS CLI para eliminar una configuración de despliegue, llama al [delete-deployment-config](#) comando y especifica el nombre de la configuración de despliegue. Para ver una lista de los nombres de las configuraciones de despliegue, [list-deployment-config](#) ejecute el comando.

En el siguiente ejemplo, se elimina una configuración de despliegue denominada `ThreeQuartersHealthy`.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

# Trabajar con aplicaciones en CodeDeploy

Después de configurar las instancias, pero antes de poder implementar una revisión, debe crear una aplicación en CodeDeploy. Una aplicación es simplemente un nombre o un contenedor que se utiliza CodeDeploy para garantizar que se haga referencia a la revisión, la configuración de implementación y el grupo de implementación correctos durante una implementación.

Utilice la siguiente información de la tabla que se indica a continuación para realizar estos pasos:

| Plataforma de informática                         | Escenario                                                                                                  | Información para el paso siguiente                                                                        |
|---------------------------------------------------|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| EC2/En las instalaciones                          | Todavía no he creado las instancias.                                                                       | Consulte <a href="#">Trabajar con instancias para CodeDeploy</a> y, a continuación, vuelva a esta página. |
| EC2/En las instalaciones                          | He creado las instancias, pero no ha terminado de etiquetar las.                                           | Consulte <a href="#">Tagging Instances for Deployments</a> y, a continuación, vuelva a esta página.       |
| EC2/en las instalaciones, AWS Lambda y Amazon ECS | Todavía no he creado una aplicación.                                                                       | Consulte <a href="#">Cree una aplicación con CodeDeploy</a>                                               |
| EC2/en las instalaciones, AWS Lambda y Amazon ECS | Ya he creado una aplicación, pero no he creado un grupo de implementaciones.                               | Consulte <a href="#">Cree un grupo de despliegue con CodeDeploy</a> .                                     |
| EC2/en las instalaciones, AWS Lambda y Amazon ECS | Ya he creado una aplicación y un grupo de implementación, pero no he creado una revisión de la aplicación. | Consulte <a href="#">Trabajar con revisiones de aplicaciones para CodeDeploy</a> .                        |
| EC2/en las instalaciones, AWS Lambda y Amazon ECS | Ya he creado una aplicación y un grupo de implementación, y ya he cargado la revisión de                   | Consulte <a href="#">Cree una implementación con CodeDeploy</a> .                                         |

| Plataforma de informática | Escenario                                          | Información para el paso siguiente |
|---------------------------|----------------------------------------------------|------------------------------------|
|                           | mi aplicación. Estoy listo para la implementación. |                                    |

## Temas

- [Cree una aplicación con CodeDeploy](#)
- [Vea los detalles de la solicitud con CodeDeploy](#)
- [Creación de una regla de notificación](#)
- [Cambiar el nombre de una aplicación CodeDeploy](#)
- [Eliminar una aplicación en CodeDeploy](#)

## Cree una aplicación con CodeDeploy


Una aplicación es simplemente un nombre o un contenedor que se utiliza CodeDeploy para garantizar que se haga referencia a la revisión, la configuración y el grupo de implementación correctos durante una implementación. Puede usar la CodeDeploy consola AWS CLI, las CodeDeploy API o una AWS CloudFormation plantilla para crear aplicaciones.

El código, o la revisión de la aplicación, se instala en las instancias mediante un proceso denominado despliegue. CodeDeploy admite dos tipos de despliegues:

- **Implementación local:** la aplicación de cada instancia del grupo de implementación se para, se instala la última revisión de la aplicación, y se inicia y valida la nueva versión de la aplicación. Puede utilizar un equilibrador de carga de modo que se cancele el registro de cada instancia durante su implementación y, a continuación, vuelva a ponerse en servicio una vez completada la implementación. Solo las implementaciones que usan la plataforma de informática de EC2/en las instalaciones pueden utilizar implementaciones locales. Para obtener más información acerca de las implementaciones locales, consulte [Información general de la implementación local](#).
- **Implementación azul/verde:** el funcionamiento de la implementación depende de la plataforma de informática que utilice:
  - **Azul/verde en una plataforma de informática EC2/en las instalaciones:** las instancias de un grupo de implementación (el entorno original) se sustituyen por otro conjunto de instancias (el entorno de sustitución) siguiendo estos pasos:



- Las instancias se aprovisionan en el entorno de sustitución.
- La revisión de aplicación más reciente se instala en las instancias de sustitución.
- Existe un tiempo de espera opcional para actividades como las pruebas de aplicaciones y la verificación del sistema.
- Las instancias del entorno de sustitución se registran en uno o más equilibradores de carga de Elastic Load Balancing, lo que provoca que el tráfico se redirija a ellos. Las instancias del entorno original se anulan del registro y se pueden cerrar o mantener en ejecución para otros usos.

 Note

Si utiliza una plataforma de informática EC2/en las instalaciones, tenga en cuenta que las implementaciones azul/verde solo funcionan con instancias de Amazon EC2.

- Azul/verde en una plataforma informática o AWS Lambda Amazon ECS: el tráfico se desplaza en incrementos según una configuración canaria, lineal o all-at-once implementación.
- Implementaciones azules/verdes AWS CloudFormation: el tráfico pasa de los recursos actuales a los recursos actualizados como parte de una actualización de la pila. AWS CloudFormation Actualmente, solo se admiten implementaciones de ECS blue/green (azul/verde).

Para obtener más información acerca de las implementaciones blue/green, consulte [Información general de la implementación azul/verde](#).

Cuando usa la CodeDeploy consola para crear una aplicación, configura su primer grupo de implementación al mismo tiempo. Cuando se utiliza AWS CLI para crear una aplicación, se crea su primer grupo de despliegues en un paso independiente.

Para ver una lista de las aplicaciones que ya están registradas en su AWS cuenta, consulte [Vea los detalles de la solicitud con CodeDeploy](#). Para obtener información sobre el uso AWS CloudFormation de una plantilla para crear una aplicación, consulte [AWS CloudFormation plantillas de CodeDeploy referencia](#).

Los dos tipos de implementación no se aplican a todos los destinos. En la tabla siguiente se muestra una lista de los tipos de implementación que funcionan con implementaciones en los tres tipos de destinos de implementación.

| Destino de implementación         | In situ | Azul/verde |
|-----------------------------------|---------|------------|
| Amazon EC2                        | Sí      | Sí         |
| Implementación on-premise         | Sí      | No         |
| Funciones AWS Lambda sin servidor | No      | Sí         |
| Aplicaciones de Amazon ECS        | No      | Sí         |

## Temas

- [Creación de una aplicación para una implementación local \(consola\)](#)
- [Creación de una aplicación para una implementación azul/verde \(consola\)](#)
- [Creación de una aplicación para la implementación de un servicio de Amazon ECS \(consola\)](#)
- [Creación de una aplicación para la implementación de una función de AWS Lambda \(consola\)](#)
- [Creación de una aplicación \(CLI\)](#)

## Creación de una aplicación para una implementación local (consola)

Para usar la CodeDeploy consola a fin de crear una aplicación para una implementación local:

### Warning


No siga estos pasos si:

- No ha preparado las instancias para usarlas en CodeDeploy las implementaciones. Para configurar sus instancias, siga las instrucciones de [Trabajar con instancias para CodeDeploy](#) y, a continuación, siga los pasos de este tema.
- Desea crear una aplicación que utilice una configuración de implementación personalizada, pero todavía no ha creado dicha configuración de implementación. Siga las instrucciones de [Create a Deployment Configuration](#) y, a continuación, siga los pasos de este tema.

- No tiene un rol de servicio en el que confíe CodeDeploy con la confianza y los permisos mínimos requeridos. Para crear y configurar un rol de servicio con los permisos necesarios, siga las instrucciones de [Paso 2: Crear un rol de servicio para CodeDeploy](#) y, a continuación, vuelva a los pasos de este tema.
- Desea seleccionar un equilibrador de carga clásico, equilibrador de carga de aplicación o equilibrador de carga de red en Elastic Load Balancing para la implementación local, pero aún no lo ha creado.


Para crear una aplicación para una implementación local mediante la CodeDeploy consola:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, a continuación, elija Introducción.
3. Elija Crear aplicación.
4. En Application name (Nombre de la aplicación), especifique el nombre de la aplicación.
5. En Compute platform (Plataforma de informática) elija EC2/On-Premises.
6. Elija Crear aplicación.
7. En la página de la aplicación, en la pestaña Grupos de implementación, elija Crear grupo de implementación.
8. En Nombre de grupo de implementación, introduzca un nombre que describa el grupo de implementación.

 Note

Si desea utilizar la misma configuración empleada en otro grupo de implementación (incluido el nombre del grupo de implementación; las etiquetas, los nombres de grupo de Amazon EC2 Auto Scaling o ambos, y la configuración de implementación), especifique esos valores en esta página. Si bien este nuevo grupo de implementación y el grupo de implementación existente tienen el mismo nombre, los CodeDeploy trata como

grupos de implementación independientes, ya que cada uno de ellos está asociado a aplicaciones distintas.

9. En Función de servicio, elija una función de servicio que conceda CodeDeploy acceso a la instancia de destino.
10. En Tipo de implementación, elija Local.
11. En Environment configuration (Configuración del entorno), seleccione una de las opciones siguientes:
  - a. Grupos de Amazon EC2 Auto Scaling: introduzca o elija el nombre de un grupo de Amazon EC2 Auto Scaling para implementar la revisión de la aplicación. Cuando se lanzan nuevas instancias de Amazon EC2 como parte de un CodeDeploy grupo de Auto Scaling de Amazon EC2, puede implementar sus revisiones en las nuevas instancias automáticamente. Puede añadir hasta 10 grupos de Amazon EC2 Auto Scaling a un grupo de implementación.
  - b. Instancias de Amazon EC2 o Instancias en las instalaciones: en los campos Clave y Valor, escriba los valores del par de clave-valor que ha usado para etiquetar las instancias. Puede etiquetar hasta 10 pares de clave-valor en un único grupo de etiquetas.
    - i. Puede utilizar comodines en el campo Valor para identificar todas las instancias etiquetadas en determinados patrones, por ejemplo, instancias de Amazon EC2, centros de costes, nombres de grupo y otros elementos similares. Por ejemplo, si selecciona Nombre en el campo Clave e ingresa **GRP- \*a** en el campo Valor, CodeDeploy identifica todas las instancias que se ajustan a ese patrón, como **GRP-1aGRP-2a**, y **GRP-XYZ-a**
    - ii. El campo Value (Valor) distingue entre mayúsculas y minúsculas.
    - iii. Para eliminar un par clave-valor de la lista, elija Remove tag (Quitar etiqueta).

A medida que CodeDeploy busca instancias que coincidan con cada par clave-valor especificado o con el nombre del grupo de Auto Scaling de Amazon EC2, muestra el número de instancias coincidentes. Elija el número para obtener más información sobre las instancias.

Si desea redefinir los criterios de las instancias en las que se han realizado implementaciones, elija Add tag group para crear un grupo de etiquetas. Puede crear un máximo de tres grupos de etiquetas con hasta diez pares de clave-valor cada uno. Cuando utiliza varios grupos de etiquetas en un grupo de implementaciones, solo las

instancias identificadas por todos los grupos de etiquetas se incluyen en el grupo de implementaciones. Esto significa una instancia debe coincidir con al menos una de las etiquetas de cada uno de los grupos que se deben incluir en el grupo de implementaciones.

Para obtener información sobre el uso de grupos de etiquetas para redefinir el grupo de implementaciones, consulte [Tagging Instances for Deployments](#).

12. En Deployment settings (Configuración de la implementación), elija una configuración de implementación para controlar la velocidad en que la aplicación se implementa en las instancias, por ejemplo, una a una o todas a la vez. Para obtener más información acerca de las configuraciones de implementación, consulte [Trabajar con configuraciones de implementación en CodeDeploy](#).
13. (Opcional) En Load Balancer, selecciona Enable Load Balancing y, a continuación, en las listas, selecciona los grupos objetivo Classic Load Balancers, Application Load Balancer y Network Load Balancer para gestionar el tráfico a las instancias durante la implementación. CodeDeploy Puede seleccionar hasta 10 equilibradores de carga clásicos y 10 grupos de destino, para un total de 20 elementos. Asegúrese de que las instancias de Amazon EC2 en las que desea implementar estén registradas en los equilibradores de carga (equilibradores de carga clásicos) o grupos de destino (equilibradores de carga de aplicación y equilibradores de carga de red) seleccionados.

Durante una implementación, las instancias originales se cancelan del registro de los equilibradores de carga y los grupos de destino seleccionados para evitar que el tráfico se dirija a estas instancias durante la implementación. Cuando se completa la implementación, cada instancia se vuelve a registrar con todos los equilibradores de carga clásicos y los grupos de destino seleccionados.

Para obtener más información sobre los balanceadores de carga para las implementaciones, consulte. CodeDeploy [Integrating CodeDeploy with Elastic Load Balancing](#)

14. (Opcional) Amplíe Avanzado y configure las opciones que desee incluir en la implementación, como los activadores de notificaciones de Amazon SNS, CloudWatch las alarmas de Amazon o las reversiones automáticas.

Para obtener más información, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

15. Elija Crear grupo de implementación.

El siguiente paso consiste en preparar una revisión para implementar en la aplicación y el grupo de implementaciones. Para ver instrucciones, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).

## Creación de una aplicación para una implementación azul/verde (consola)

Para usar la CodeDeploy consola a fin de crear una aplicación para una implementación azul/verde:

### Note

Una implementación en la plataforma de cómputo AWS Lambda es siempre una implementación azul/verde. No especifique una opción de tipo de implementación.

### Warning

No siga estos pasos si:


- No tiene instancias con el CodeDeploy agente instalado que desee reemplazar durante el proceso de implementación azul/verde. Para configurar sus instancias, siga las instrucciones de [Trabajar con instancias para CodeDeploy](#) y, a continuación, siga los pasos de este tema.
- Desea crear una aplicación que utilice una configuración de implementación personalizada, pero todavía no ha creado dicha configuración de implementación. Siga las instrucciones de [Create a Deployment Configuration](#) y, a continuación, siga los pasos de este tema.
- No tiene un rol de servicio en el que se confíe, como mínimo, CodeDeploy con la confianza y los permisos descritos en. [Paso 2: Crear un rol de servicio para CodeDeploy](#) Para crear y configurar un rol de servicio, siga las instrucciones de [Paso 2: Crear un rol de servicio para CodeDeploy](#) y, a continuación, siga los pasos de este tema.
- No ha creado un equilibrador de carga clásico, equilibrador de carga de aplicación o equilibrador de carga de red en Elastic Load Balancing para el registro de las instancias en su entorno de reemplazo. Para obtener más información, consulte [Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2](#).

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, a continuación, elija Introducción.
3. En Application name (Nombre de la aplicación), especifique el nombre de la aplicación.
4. En Compute platform (Plataforma de informática) elija EC2/On-Premises.
5. Elija Crear aplicación.
6. En la página de la aplicación, en la pestaña Grupos de implementación, elija Crear grupo de implementación.
7. En Nombre de grupo de implementación, introduzca un nombre que describa el grupo de implementación.

 Note

Si desea utilizar la misma configuración empleada en otro grupo de implementación (incluido el nombre del grupo de implementación; las etiquetas, los nombres de grupo de Amazon EC2 Auto Scaling y la configuración de implementación), elija esos valores en esta página. Si bien este nuevo grupo de implementación y el grupo de implementación existente tienen el mismo nombre, CodeDeploy los trata como grupos de implementación independientes, ya que cada uno está asociado a una aplicación diferente.

8. En Función de servicio, elija una función de servicio que conceda CodeDeploy acceso a la instancia de destino.
9. En Deployment type (Tipo de implementación) elija Blue/green.
10. En Environment configuration, elija el método que desea utilizar para proporcionar instancias para su entorno de sustitución:
  - a. Copiar automáticamente el grupo Auto Scaling de Amazon EC2: CodeDeploy crea un grupo de Auto Scaling de Amazon EC2 copiando el que especifique.
  - b. Aprovisionar instancias manualmente: no especificará las instancias de su entorno de sustitución hasta que cree una implementación. Debe crear las instancias antes de empezar

la implementación. En su lugar, en este caso debe especificar las instancias que desea sustituir.

11. En función de su elección en el paso 10, lleve a cabo alguna de las siguientes operaciones:
  - Si eligió Copiar automáticamente el grupo Auto Scaling de Amazon EC2: en Grupo de Amazon EC2 Auto Scaling, elija o introduzca el nombre del grupo de Amazon EC2 Auto Scaling que desee utilizar como plantilla para el grupo de Amazon EC2 Auto Scaling para las instancias del entorno de reemplazo. El número de instancias actualmente en buen estado en el grupo de Amazon EC2 Auto Scaling que elija se crea en su entorno de sustitución.
  - Si eligió Aprovisionar instancias manualmente: habilite Grupos de Amazon EC2 Auto Scaling, Instancias de Amazon EC2 o ambos para especificar las instancias que se van a añadir a este grupo de implementación. Introduzca los valores de etiqueta de Amazon EC2 o los nombres de grupo de Amazon EC2 Auto Scaling para identificar las instancias del entorno original (es decir, las instancias que desea reemplazar o que ejecutan la revisión de la aplicación actual).
12. En Equilibrador de carga, seleccione Habilitar equilibrio de carga y, a continuación, en las listas, seleccione los equilibradores de carga clásicos, los grupos de destino de los equilibradores de carga de aplicación y los grupos de destino de los equilibradores de carga de red con los que desee registrar las instancias de Amazon EC2 de reemplazo. Cada instancia de reemplazo se registrará con todos los equilibradores de carga clásicos y los grupos de destino seleccionados. Puede seleccionar hasta 10 equilibradores de carga clásicos y 10 grupos de destino, para un total de 20 elementos.

El tráfico se redirigirá de las instancias originales a las de reemplazo según los ajustes de Redireccionamiento de tráfico y Configuración de implementación que haya elegido.

Para obtener más información sobre los balanceadores de carga para las CodeDeploy implementaciones, consulte [Integrating CodeDeploy with Elastic Load Balancing](#)


13. En Deployment settings, revise las opciones predeterminadas para redirigir el tráfico al entorno de sustitución, qué configuración de implementación desea utilizar para la implementación y cómo se gestionan las instancias en el entorno original después de la implementación.

Si desea cambiar la configuración, continúe con el paso siguiente. De lo contrario, vaya al paso 15.

14. Para cambiar la configuración de la implementación blue/green cambie cualquiera de los siguientes ajustes.



| Opción                                             | Opciones                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Traffic rerouting (Nuevo enrutamiento del tráfico) | <ul style="list-style-type: none"><li>• Redirigir el tráfico inmediatamente: tan pronto como se aprovisionan las instancias del entorno de sustitución y se instala en ellas la última revisión de la aplicación, se registran automáticamente en el equilibrador de carga, lo que hace que el tráfico se redirija a ellas. Se anula el registro de las instancias en el entorno original.</li><li>• Elegir si se debe redirigir el tráfico: las instancias del entorno de sustitución no se registran automáticamente en el equilibrador de carga a menos que redirija el tráfico manualmente. Si se agota el tiempo de espera especificado sin que se redirija el tráfico, el estado de la implementación cambia a Stopped.</li></ul> |

| Opción                                     | Opciones                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Configuración de implementación            | <p>Elija el modo en que las instancias del entorno de sustitución se registran en el entorno de sustitución con el equilibrador de carga y los grupos de destino, por ejemplo, una a una o todas a la vez.</p> <div data-bbox="862 495 1507 951" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Una vez que el tráfico se dirige al entorno de sustitución de forma satisfactoria, se anula el registro de todas las instancias del entorno original a la vez, independientemente de la configuración de implementación que haya seleccionado.</p> </div> <p>Para obtener más información, consulte <a href="#">Trabajar con configuraciones de implementación en CodeDeploy</a>.</p> |
| Original instances (Instancias originales) | <ul style="list-style-type: none"> <li>• Terminar las instancias originales en el grupo de implementación: una vez que el tráfico se redirija al entorno de sustitución, las instancias cuyo registro se anuló en los equilibradores de carga y los grupos de destino se terminan tras el periodo de espera especificado.</li> <li>• Mantener en ejecución las instancias originales del grupo de implementación: una vez que el tráfico se redirija al entorno de sustitución, las instancias cuyo registro se anuló en los equilibradores de carga y los grupos de destino se mantienen en ejecución.</li> </ul>                                                                                                                                                                                                                     |

15. (Opcional) En Avanzado, configure las opciones que desee incluir en la implementación, como los activadores de notificaciones de Amazon SNS, CloudWatch las alarmas de Amazon o las reversiones automáticas.

Para obtener información acerca de cómo especificar opciones avanzadas en grupos de implementaciones, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

16. Elija Crear grupo de implementación.

El siguiente paso consiste en preparar una revisión para implementar en la aplicación y el grupo de implementaciones. Para ver instrucciones, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).

## Creación de una aplicación para la implementación de un servicio de Amazon ECS (consola)

Puede usar la CodeDeploy consola para crear una aplicación para la implementación de un servicio de Amazon ECS.


1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Introducción.
3. En la página Crear aplicación, selecciona Usar CodeDeploy.
4. En Application name (Nombre de la aplicación), especifique el nombre de la aplicación.
5. En Plataforma de informática, elija Amazon ECS.
6. Elija Crear aplicación.
7. En la página de la aplicación, en la pestaña Grupos de implementación, elija Crear grupo de implementación. Para obtener más información sobre lo que necesita para crear un grupo de implementación para una implementación de Amazon ECS, consulte [Antes de empezar una implementación de Amazon ECS](#).

8. En Nombre de grupo de implementación, introduzca un nombre que describa el grupo de implementación.

 Note

Si desea utilizar la misma configuración empleada en otro grupo de implementación (incluido el nombre del grupo de implementación y la configuración de implementación), elija esos valores en esta página. Si bien este nuevo grupo y el grupo existente pueden tener el mismo nombre, los CodeDeploy trata como grupos de implementación independientes, ya que cada uno está asociado a una aplicación diferente.

9. En Función de servicio, elija una función de servicio que conceda CodeDeploy acceso a Amazon ECS. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).
10. En Nombre del equilibrador de carga, seleccione el nombre del equilibrador de carga que envía tráfico a su servicio de Amazon ECS.
11. En Puerto del oyente de producción, elija el puerto y el protocolo del oyente que envía el tráfico de producción a su servicio de ECS de Amazon.
12. (Opcional) En Puerto de oyente de prueba, elija el puerto y el protocolo de un oyente de prueba que envíe el tráfico a la tarea de reemplazo establecida en su servicio de Amazon ECS durante la implementación. Puede especificar una o más funciones Lambda en el AppSpec archivo que se ejecutarán durante el enlace. `AfterAllowTestTraffic` Las funciones pueden ejecutar pruebas de validación. Si la prueba de validación falla, se activa una restauración de la implementación. Si se superan las pruebas de validación, se desencadena el siguiente enlace del ciclo de vida de la implementación, `BeforeAllowTraffic`. Si no se especifica un puerto de oyente de prueba, no ocurre nada durante el enlace `AfterAllowTestTraffic`. Para obtener más información, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#).
13. Entre Nombre del grupo objetivo 1 y Nombre del grupo objetivo 2, elija los grupos de destino que se utilizarán para enrutar el tráfico durante la implementación. CodeDeploy vincula un grupo objetivo al conjunto de tareas original del servicio Amazon ECS y el otro al conjunto de tareas de reemplazo. Para obtener más información, consulte [Grupos de destino para sus equilibradores de carga de aplicación](#).
14. Elija Redirigir el tráfico inmediatamente o Especificar cuándo redirigir el tráfico para determinar cuándo redirigir el tráfico a su servicio de Amazon ECS actualizado.

Si elige Redirigir el tráfico inmediatamente, la implementación redireccionará automáticamente el tráfico una vez que se haya provisionado el conjunto de tareas de reemplazo.

Si elige Especificar cuándo redirigir el tráfico, elija el número de días, horas y minutos que debe esperar hasta que el conjunto de tareas de reemplazo se aprovisione correctamente. Durante este tiempo de espera, se ejecutan las pruebas de validación en las funciones Lambda especificadas en el AppSpec archivo. Si el tiempo de espera expira antes de que se redirija el tráfico, el estado de la implementación cambia a Stopped.

15. En Finalización de la revisión original, elija el número de días, horas y minutos que se deben esperar hasta que finalice la tarea original establecida en su servicio de Amazon ECS tras una implementación correcta.
16. (Opcional) En Avanzado, configure las opciones que desee incluir en la implementación, como los activadores de notificaciones de Amazon SNS, CloudWatch las alarmas de Amazon o las reversiones automáticas.

Para obtener más información, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

## Creación de una aplicación para la implementación de una función de AWS Lambda (consola)

Puede usar la CodeDeploy consola para crear una aplicación para el despliegue de una AWS Lambda función.


1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Introducción.
3. En la página Crear aplicación, selecciona Usar CodeDeploy.
4. Especifique el nombre de la aplicación en Application name (Nombre de la aplicación).
5. En Compute platform (Plataforma de informática), elija AWS Lambda.
6. Elija Crear aplicación.

7. En la página de la aplicación, en la pestaña Grupos de implementación, elija Crear grupo de implementación.
8. En Nombre de grupo de implementación, introduzca un nombre que describa el grupo de implementación.

 Note


Si desea utilizar la misma configuración empleada en otro grupo de implementación (incluido el nombre del grupo de implementación y la configuración de implementación), elija esos valores en esta página. Si bien este nuevo grupo de implementación y el grupo de implementación existente pueden tener el mismo nombre, los CodeDeploy trata como grupos de implementación independientes, ya que cada uno está asociado a una aplicación diferente.

9. En Función de servicio, elija una función de servicio a la que conceda CodeDeploy acceso AWS Lambda. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).
10. Si desea utilizar una configuración de implementación predefinida, elija una en Deployment configuration (Configuración de implementación) y, a continuación, vaya al paso 12. Para crear una configuración personalizada, continúe al paso siguiente.

Para obtener más información acerca de las configuraciones de implementación, consulte [Configuraciones de implementación en una plataforma de informática de AWS Lambda](#).

11. Para crear una configuración personalizada, elija Create deployment configuration (Crear configuración de implementación) y, a continuación, haga lo siguiente:
  - a. En Deployment configuration name (Nombre de la configuración de implementación), introduzca un nombre para la configuración.
  - b. En Type (Tipo), elija un tipo de configuración. Si elige Canary, el tráfico se desvía en dos incrementos. Si elige Linear, el tráfico se desvía en incrementos iguales con el mismo número de minutos entre incrementos.
  - c. En Step, introduzca el porcentaje de tráfico, comprendido entre 1 y 99, que se debe desviar. Si el tipo de configuración es Canary, este es el porcentaje de tráfico que se desvía en el primer incremento. El tráfico restante se desvía después del intervalo seleccionado en el segundo incremento. Si el tipo de configuración es Linear, este es el porcentaje de tráfico que se desvía al comienzo de cada intervalo.

- d. En Interval (Intervalo), introduzca el número de minutos. Si el tipo de configuración es Canary, este es el número de minutos entre el primer y el segundo desvío de tráfico. Si el tipo de configuración es Linear (Lineal), este es el número de minutos entre cada desvío incremental.

 Note

La duración máxima de una AWS Lambda implementación es de dos días o 2880 minutos. Por lo tanto, el valor máximo especificado para Interval en una configuración “canary” es de 2 800 minutos. EL valor máximo para una configuración “linear” depende del valor de Step. Por ejemplo, si el porcentaje de incremento de un desvío de tráfico “linear” es del 25%, hay cuatro desvíos de tráfico. El valor de intervalo máximo es de 2880 dividido entre cuatro, o 720 minutos.

- e. Elija Create deployment configuration (Crear configuración de implementación).
12. (Opcional) En Avanzado, configure las opciones que desee incluir en la implementación, como los activadores de notificaciones de Amazon SNS, CloudWatch las alarmas de Amazon o las reversiones automáticas.

Para obtener más información, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

13. Elija Crear grupo de implementación.

## Creación de una aplicación (CLI)

Para usar el AWS CLI para crear una aplicación, ejecute el comando [create-application](#) y especifique un nombre que represente a la aplicación de forma exclusiva. (En una AWS cuenta, el nombre de una CodeDeploy aplicación solo se puede usar una vez por región. Puede reutilizar el nombre de una aplicación en diferentes regiones).

Tras utilizar el AWS CLI para crear una aplicación, el siguiente paso es crear un grupo de despliegue que especifique las instancias en las que se van a implementar las revisiones. Para ver instrucciones, consulte [Cree un grupo de despliegue con CodeDeploy](#).

Después de crear el grupo de implementaciones, el siguiente paso consiste en preparar una revisión para implementar en la aplicación y el grupo de implementaciones. Para obtener instrucciones, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).

# Ve los detalles de la solicitud con CodeDeploy

Puedes usar la CodeDeploy consola AWS CLI, las API o las CodeDeploy API para ver los detalles de todas las aplicaciones asociadas a tu AWS cuenta.

## Temas

- [Visualización de los detalles de la aplicación \(consola\)](#)
- [Visualización de los detalles de la aplicación \(CLI\)](#)

## Visualización de los detalles de la aplicación (consola)

Para usar la CodeDeploy consola para ver los detalles de la aplicación:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Introducción.
3. Para ver información detallada adicional acerca de la aplicación, elija el nombre de la aplicación de la lista.

## Visualización de los detalles de la aplicación (CLI)

Para usar el AWS CLI para ver los detalles de la aplicación, `get-application` ejecuta el `batch-get-application` comando, el comando o el `list-applications` comando.

Para ver información detallada acerca de una sola aplicación, llame al comando [get-application](#), especificando el nombre de la aplicación.

Para ver los detalles de varias aplicaciones, [batch-get-application](#) ejecute el comando y especifique los nombres de varias aplicaciones.

Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).



# Creación de una regla de notificación

Puede utilizar reglas de notificación para notificar a los usuarios cuando hay cambios en las aplicaciones de implementación, como éxitos y errores de implementación. Las reglas de notificación especifican tanto los eventos como el tema de Amazon SNS que se utiliza para enviar notificaciones. Para obtener más información, consulte [¿Qué son las notificaciones?](#)

Puede usar la consola o la AWS CLI para crear reglas de notificación para AWS CodeDeploy.

Para crear una regla de notificación (consola)

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy/>.
2. Elija Application (Aplicación) y, a continuación, elija una aplicación donde desea agregar notificaciones.
3. En la página de la aplicación, elija Notify (Notificar) y, a continuación, elija Create notification rule (Crear regla de notificación). También puede ir a la página Settings (Configuración) para la aplicación y elegir Create notification rule (Crear regla de notificación).
4. En Nombre de la notificación, introduzca un nombre para la regla.
5. En Tipo de detalle, selecciona Básico si quieres que solo se EventBridge incluya en la notificación la información proporcionada a Amazon. Selecciona Completa si deseas incluir la información proporcionada a Amazon EventBridge y la información que pueda proporcionar el administrador de notificaciones CodeDeploy o el administrador de notificaciones.

Para obtener más información, consulte [Descripción del contenido y la seguridad de las notificaciones](#).

6. En Eventos que activan notificaciones, seleccione los eventos para los que desea enviar notificaciones.

| Categoría      | Eventos   |
|----------------|-----------|
| Implementación | Con error |
|                | Succeeded |
|                | Started   |

7. En Targets (Destinos), elija Create SNS topic (Crear tema SNS).

**Note**

Al crear el tema, se aplicará la política que CodeDeploy permite publicar eventos sobre el tema. El uso de un tema creado específicamente para CodeDeploy las notificaciones también ayuda a garantizar que solo añada usuarios a la lista de suscripciones de ese tema para los que desee ver las notificaciones sobre esta aplicación de despliegue.

Después del prefijo `codestar-notifications-` escriba un nombre para el tema y, a continuación, elija `Submit` (Enviar).

**Note**

Si desea utilizar un tema de Amazon SNS existente en lugar de crear uno nuevo, en `Targets` (Destinos), elija su ARN. Asegúrese de que el tema tiene la directiva de acceso adecuada y de que la lista de suscriptores contiene solo aquellos usuarios a los que se permite ver información sobre la aplicación de implementación. Para obtener más información, consulte [Configuración de temas existentes de Amazon SNS para notificaciones](#) y [Descripción del contenido y la seguridad de las notificaciones](#).

- Para terminar de crear la regla, elija `Enviar`.
- Debe suscribir a los usuarios al tema de Amazon SNS de la regla antes de que puedan recibir notificaciones. Para obtener más información, consulte [Suscribir usuarios a temas de Amazon SNS que son destinos](#). También puede configurar la integración entre las notificaciones y enviar notificaciones AWS Chatbot a las salas de chat de Amazon Chime o a los canales de Slack. Para obtener más información, consulte [Configurar la integración entre notificaciones y AWS Chatbot](#).

Para crear una regla de notificación (AWS CLI)

- En un terminal o símbolo del sistema, ejecute el comando `create-notification-rule` para generar el esqueleto JSON:

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton
> rule.json
```

Puede asignar al archivo el nombre que desee. En este ejemplo, el archivo se denomina *rule.json*.

- Abra el archivo JSON en un editor de texto sin formato y edítelo para incluir el recurso, los tipos de eventos y el destino de Amazon SNS que desea para la regla. En el siguiente ejemplo, se muestra una regla de notificación con **MyNotificationRule** el nombre de una aplicación nombrada *MyDeploymentApplication* en una AWS cuenta con el identificador *123456789012*. Las notificaciones se envían con todos los detalles a un tema de Amazon SNS denominado *codestar-notifications- MyNotificationTopic cuando las implementaciones se realizan correctamente*:

```
{
 "Name": "MyNotificationRule",
 "EventTypeId": [
 "codedeploy-application-deployment-succeeded"
],
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyDeploymentApplication",
 "Targets": [
 {
 "TargetType": "SNS",
 "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestar-
notifications-MyNotificationTopic"
 }
],
 "Status": "ENABLED",
 "DetailType": "FULL"
}
```

Guarde el archivo.

- Mediante el archivo que acaba de modificar, en el terminal o línea de comandos, vuelva a ejecutar el comando `create-notification-rule` para crear la regla de notificación:

```
aws codestar-notifications create-notification-rule --cli-input-json
file://rule.json
```

- Si se ejecuta correctamente, el comando devuelve el ARN de la regla de notificación, similar a lo siguiente:

```
{
```

```
"Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

## Cambiar el nombre de una aplicación CodeDeploy

Puede usar las API AWS CLI o las CodeDeploy API para cambiar el nombre de una aplicación.

Para ver una lista de nombres de aplicaciones, utilice el AWS CLI comando to call [list-applications](#).

[Para obtener información sobre cómo usar el AWS CLI para cambiar el nombre de una aplicación, consulte update-application.](#)

[Para obtener información sobre el uso de las CodeDeploy API para cambiar el nombre de una aplicación, consulte API\\_. UpdateApplication](#)

## Eliminar una aplicación en CodeDeploy

Puede usar la CodeDeploy consola AWS CLI, la o una acción de la CodeDeploy API para eliminar aplicaciones. Para obtener información sobre el uso de la acción de la CodeDeploy API, consulte [DeleteApplication](#).

### Warning

Al eliminar una aplicación, se elimina la información sobre la aplicación del CodeDeploy sistema, incluida toda la información relacionada con el grupo de implementación y los detalles de la implementación. La eliminación de una aplicación creada para una implementación de EC2/en las instalaciones no elimina las revisiones de la aplicación relacionadas de las instancias ni elimina la revisiones de los buckets de Amazon S3. La eliminación de una aplicación creada para una implementación de EC2/en las instalaciones no finaliza las instancias de Amazon EC2 ni anula el registro de las instancias en las instalaciones. Esta acción no se puede deshacer.

### Temas

- [Eliminar una aplicación \(consola\)](#)
- [Eliminar una aplicación \(AWS CLI\)](#)

## Eliminar una aplicación (consola)

Para usar la CodeDeploy consola para eliminar una aplicación:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la lista de aplicaciones, seleccione la aplicación que desee eliminar.  
  
Aparece una página con detalles sobre la aplicación.
4. Seleccione Eliminar aplicación en la esquina superior derecha.
5. Cuando se le solicite, introduzca **delete** para confirmar que desea eliminar la aplicación y seleccione Eliminar.

## Eliminar una aplicación (AWS CLI)

Para usar el AWS CLI para eliminar una aplicación, ejecute el comando [delete-application](#) y especifique el nombre de la aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).

# Trabajar con grupos de despliegue en CodeDeploy

Puede especificar uno o más grupos de despliegue para una CodeDeploy aplicación. Cada implementación de aplicación utiliza uno de sus grupos de implementaciones. El grupo de implementaciones contiene los ajustes y las configuraciones utilizadas durante la implementación. La mayoría de los ajustes del grupo de implementación dependen de la plataforma de informática que utiliza su aplicación. Algunas opciones de configuración, tales como restauraciones, desencadenadores y alarmas se puede configurar para grupos de implementación para cualquier plataforma de informática.

## Grupos de implementación en las implementaciones en la plataforma de informática Amazon ECS

En una implementación de Amazon ECS, un grupo de implementación especifica el servicio de Amazon ECS, el equilibrador de carga, el oyente de prueba opcional y dos grupos de destino. También especifica cuándo debe redirigirse el tráfico al conjunto de tareas de sustitución y cuándo terminar el conjunto de tareas original y la aplicación de Amazon ECS después de una implementación correcta.

## Grupos de despliegue en despliegues de plataformas AWS Lambda informáticas

En una implementación de AWS Lambda, un grupo de implementación define un conjunto de CodeDeploy configuraciones para las futuras implementaciones de una función. AWS Lambda Por ejemplo, el grupo de implementación especifica cómo dirigir el tráfico a una nueva versión de una función de Lambda. También podría especificar alarmas y restauraciones. Una sola implementación en un grupo de implementación de AWS Lambda puede anular una o más configuraciones de grupo.

## Grupos de implementación en las implementaciones en la plataforma de informática EC2/en las instalaciones

En una implementación EC2/en las instalaciones, un grupo de implementación es un conjunto de instancias individuales dirigidas a una implementación. Un grupo de implementación contiene instancias etiquetadas individualmente, instancias de Amazon EC2 en grupos Amazon EC2 Auto Scaling o ambas.

En una implementación local, las instancias del grupo de implementaciones se actualizan con la última revisión de la aplicación.

En una implementación azul/verde, el tráfico se redirige de un conjunto de instancias a otro desinstalando las instancias originales de uno o varios equilibradores de carga y registrando un conjunto de sustitución de instancias que suele tener instalada la última revisión de la aplicación.

Puede asociar más de un grupo de despliegues a una aplicación en CodeDeploy. Esto permite implementar una revisión de la aplicación en diferentes conjuntos de instancias en diferentes momentos. Por ejemplo, puede utilizar un grupo de implementaciones para implementar una revisión de la aplicación en un conjunto de instancias etiquetadas `Test`, en las que garantiza la calidad del código. A continuación, puede realizar una verificación adicional implementando la misma revisión de la aplicación en un grupo de implementaciones con instancias etiquetadas como `Staging`. Por último, cuando todo esté listo para lanzar la última aplicación y ponerla a disposición de los clientes, puede implementar a un grupo de implementaciones que incluya instancias etiquetadas como `Production`.

También puede utilizar varios grupos de etiquetas para afinar aún más los criterios para las instancias incluidas en un grupo de implementaciones. Para obtener más información, consulte [Tagging Instances for Deployments](#).

Cuando usa la CodeDeploy consola para crear una aplicación, configura su primer grupo de implementación al mismo tiempo. Cuando se utiliza AWS CLI para crear una aplicación, se crea su primer grupo de despliegues en un paso independiente.

Para ver una lista de los grupos de despliegue que ya están asociados a su AWS cuenta, consulte [Vea los detalles del grupo de implementación con CodeDeploy](#).

Para obtener información acerca de etiquetas de instancias de Amazon EC2, consulte [Uso de etiquetas mediante la consola](#). Para obtener más información acerca de las instancias en las instalaciones, consulte [Working with On-Premises Instances](#). Para obtener más información sobre Amazon EC2 Auto Scaling, consulte [Integración CodeDeploy con Amazon EC2 Auto Scaling](#).

## Temas

- [the section called “Creación de un grupo de implementación”](#)
- [the section called “Visualización de los detalles de un grupo de implementación”](#)
- [the section called “Cambio de la configuración de un grupo de implementación”](#)
- [the section called “Configuración de las opciones avanzadas de un grupo de implementación”](#)

- [the section called “Eliminación de un grupo de implementación”](#)

## Cree un grupo de despliegue con CodeDeploy

Puede usar la CodeDeploy consola AWS CLI, las CodeDeploy API o una AWS CloudFormation plantilla para crear grupos de despliegue. Para obtener información sobre el uso AWS CloudFormation de una plantilla para crear un grupo de despliegue, consulte [AWS CloudFormation plantillas de CodeDeploy referencia](#).

Cuando usa la CodeDeploy consola para crear una aplicación, configura su primer grupo de implementación al mismo tiempo. Cuando se utiliza AWS CLI para crear una aplicación, se crea su primer grupo de despliegues en un paso independiente.

Como parte de la creación de un grupo de implementaciones, debe especificar un rol de servicio. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).

### Temas

- [Creación de un grupo de implementación para una implementación local \(consola\)](#)
- [Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones \(consola\)](#)
- [Creación de un grupo de implementación para una implementación de Amazon ECS \(consola\)](#)
- [Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2](#)
- [Configure un balanceador de carga, grupos objetivo y agentes de escucha para las implementaciones de CodeDeploy Amazon ECS](#)
- [Creación de un grupo de implementación \(CLI\)](#)

## Creación de un grupo de implementación para una implementación local (consola)

Para usar la CodeDeploy consola a fin de crear un grupo de despliegues para un despliegue in situ:


### Warning

No siga estos pasos si:




- No ha preparado las instancias para usarlas en la primera CodeDeploy implementación de una aplicación. Para configurar sus instancias, siga las instrucciones de [Trabajar con instancias para CodeDeploy](#) y, a continuación, siga los pasos de este tema.
- Desea crear un grupo de implementaciones que utilice una configuración de implementación personalizada, pero todavía no ha creado dicha configuración de implementación. Siga las instrucciones de [Create a Deployment Configuration](#) y, a continuación, siga los pasos de este tema.
- No tiene un rol de servicio en el que se confíe, como mínimo, CodeDeploy con la confianza y los permisos descritos en [Paso 2: Crear un rol de servicio para CodeDeploy](#). Para crear y configurar un rol de servicio, siga las instrucciones de [Paso 2: Crear un rol de servicio para CodeDeploy](#) y, a continuación, siga los pasos de este tema.
- Desea seleccionar un equilibrador de carga clásico, equilibrador de carga de aplicación o equilibrador de carga de red en Elastic Load Balancing para la implementación local, pero aún no lo ha creado.

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la página Applications, elija el nombre de la aplicación para la que desea crear un grupo de implementaciones.
4. En la página de la aplicación, en la pestaña Grupos de implementación, elija Crear grupo de implementación.
5. En Nombre de grupo de implementación, introduzca un nombre que describa el grupo de implementación.

 Note

Si desea utilizar la misma configuración empleada en otro grupo de implementación (incluido el nombre del grupo de implementación; las etiquetas, los nombres de grupo de Amazon EC2 Auto Scaling o ambos, y la configuración de implementación), especifique

esos valores en esta página. Si bien este nuevo grupo de implementación y el grupo de implementación existente tienen el mismo nombre, los CodeDeploy trata como grupos de implementación independientes, ya que cada uno de ellos está asociado a aplicaciones distintas.

6. En Función de servicio, elija una función de servicio que conceda CodeDeploy acceso a la instancia de destino.
7. En Tipo de implementación, elija Local.
8. En Configuración del entorno, haga lo siguiente:
  - a. Si desea implementar la aplicación en un grupo de Amazon EC2 Auto Scaling, seleccione los grupos de Amazon EC2 Auto Scaling y, a continuación, elija el nombre de un grupo de Amazon EC2 Auto Scaling en el que se vaya a implementar la revisión de la aplicación. Cuando se lanzan nuevas instancias de Amazon EC2 como parte de un CodeDeploy grupo de Auto Scaling de Amazon EC2, puede implementar sus revisiones en las nuevas instancias automáticamente. Puede añadir hasta 10 grupos de Amazon EC2 Auto Scaling a un grupo de implementación. Para obtener más información, consulte [Integración CodeDeploy con Amazon EC2 Auto Scaling](#).
  - b. Si seleccionó grupos de Auto Scaling de Amazon EC2, si lo desea, seleccione Añadir un enlace de terminación a los grupos de Auto Scaling para CodeDeploy instalar un enlace de terminación en su grupo de Auto Scaling al crear o actualizar el grupo de implementación. Cuando este enlace esté instalado, CodeDeploy realizará los despliegues de terminación. Para obtener más información, consulte [Habilitación de implementaciones de terminación durante los eventos de reducción horizontal de Auto Scaling](#).
  - c. Si quiere etiquetar sus instancias, seleccione instancias Amazon EC2 o Instancias en las instalaciones. En los campos Clave y Valor, indique los valores de los pares clave-valor que ha utilizado para etiquetar las instancias. Puede etiquetar hasta 10 pares de clave-valor en un único grupo de etiquetas.
    - i. Puede utilizar comodines en el campo Valor para identificar todas las instancias etiquetadas en determinados patrones, por ejemplo, instancias de Amazon EC2, centros de costes, nombres de grupo y otros elementos similares. Por ejemplo, si elige Nombre en el campo Clave e ingresa **GRP- \*a** en el campo Valor, CodeDeploy identifica todas las instancias que se ajustan a ese patrón, como **GRP-1aGRP-2a**, y**GRP-XYZ-a**.
    - ii. El campo Value (Valor) distingue entre mayúsculas y minúsculas.

- iii. Para eliminar un par clave-valor de la lista, elija el icono de eliminación.

A medida que CodeDeploy busca instancias que coincidan con cada par clave-valor especificado o con el nombre del grupo de Auto Scaling de Amazon EC2, muestra el número de instancias coincidentes. Para obtener más información sobre las instancias, haga clic en el número.

Si desea redefinir los criterios de las instancias en las que se han realizado implementaciones, elija Add tag group para crear un grupo de etiquetas. Puede crear un máximo de tres grupos de etiquetas con hasta 10 pares de clave-valor cada uno. Cuando utiliza varios grupos de etiquetas en un grupo de implementaciones, solo las instancias identificadas por todos los grupos de etiquetas se incluyen en el grupo de implementaciones. Esto significa una instancia debe coincidir con al menos una de las etiquetas de cada uno de los grupos que se deben incluir en el grupo de implementaciones.


Para obtener información sobre el uso de grupos de etiquetas para redefinir el grupo de implementaciones, consulte [Tagging Instances for Deployments](#).

9. En Configuración del agente con Systems Manager, especifique cómo desea instalar y actualizar el CodeDeploy agente en las instancias de su grupo de implementación. Para obtener más información sobre el CodeDeploy agente, consulte [Trabajar con el CodeDeploy agente](#). Para obtener más información sobre Systems Manager, consulte [¿Qué es Systems Manager?](#)
  - a. Nunca: omite la configuración de la CodeDeploy instalación con Systems Manager. Las instancias deben tener el agente instalado para poder utilizarlas en las implementaciones, por lo que solo debe elegir esta opción si va a instalar el CodeDeploy agente de otra forma.
  - b. Solo una vez: Systems Manager instalará el CodeDeploy agente una vez en cada instancia del grupo de implementación.
  - c. Ahora y programe las actualizaciones: Systems Manager creará una asociación con State Manager que instalará el CodeDeploy agente según la programación que usted configure. Para obtener más información acerca del Administrador de estados y las asociaciones, consulte [Acerca del Administrador de estados](#).
10. En la lista Deployment configuration (Configuración de implementación), elija una configuración de implementación para controlar el modo en que se implementan las instancias, por ejemplo, una a una o todas a la vez. Para obtener más información acerca de las configuraciones de implementación, consulte [Trabajar con configuraciones de implementación en CodeDeploy](#).

11. (Opcional) En Load Balancer, selecciona Enable Load Balancing y, a continuación, en las listas, selecciona los grupos objetivo Classic Load Balancers, Application Load Balancer y Network Load Balancer para gestionar el tráfico a las instancias durante la implementación. CodeDeploy puede seleccionar hasta 10 equilibradores de carga clásicos y 10 grupos de destino, para un total de 20 elementos. Asegúrese de que las instancias de Amazon EC2 en las que desea implementar estén registradas en los equilibradores de carga (equilibradores de carga clásicos) o grupos de destino (equilibradores de carga de aplicación y equilibradores de carga de red) seleccionados.

Durante una implementación, las instancias originales se cancelan del registro de los equilibradores de carga y los grupos de destino seleccionados para evitar que el tráfico se dirija a estas instancias durante la implementación. Cuando se completa la implementación, cada instancia se vuelve a registrar con todos los equilibradores de carga clásicos y los grupos de destino seleccionados.

Para obtener más información sobre los balanceadores de carga para las implementaciones, consulte. CodeDeploy [Integrating CodeDeploy with Elastic Load Balancing](#)

 Warning

Si está configurando grupos de Auto Scaling y un balanceador de carga de Elastic Load Balancing en este grupo de despliegues y desea [adjuntar balanceadores de carga a los grupos de Auto Scaling](#), le recomendamos completar este adjunto antes de crear el CodeDeploy despliegue desde este grupo de despliegues. Si intenta completar la asociación después de crear la implementación, es posible que se cancele el registro de todas las instancias en los equilibradores de carga de forma inesperada.

12. (Opcional) Amplíe Avanzado y configure las opciones que desee incluir en la implementación, como los activadores de notificaciones de Amazon SNS, las CloudWatch alarmas de Amazon, las opciones de Auto Scaling o las reversiones automáticas.

Para obtener más información, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

13. Elija Crear grupo de implementación.

## Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones (consola)

Para usar la CodeDeploy consola a fin de crear un grupo de despliegues para un despliegue azul/verde:

### Warning

No siga estos pasos si:

- No tiene instancias con el CodeDeploy agente instalado que desee reemplazar durante el proceso de despliegue azul/verde. Para configurar sus instancias, siga las instrucciones de [Trabajar con instancias para CodeDeploy](#) y, a continuación, siga los pasos de este tema.
- Desea crear una aplicación que utilice una configuración de implementación personalizada, pero todavía no ha creado dicha configuración de implementación. Siga las instrucciones de [Create a Deployment Configuration](#) y, a continuación, siga los pasos de este tema.
- No tiene un rol de servicio en el que se confíe, como mínimo, CodeDeploy con la confianza y los permisos descritos en. [Paso 2: Crear un rol de servicio para CodeDeploy](#) Para crear y configurar un rol de servicio, siga las instrucciones de [Paso 2: Crear un rol de servicio para CodeDeploy](#) y, a continuación, siga los pasos de este tema.
- No ha creado un equilibrador de carga clásico o un equilibrador de carga de aplicación en Elastic Load Balancing para el registro de las instancias en su entorno de reemplazo. Para obtener más información, consulte [Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2](#).

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la página Applications, elija el nombre de la aplicación para la que desea crear un grupo de implementaciones.

4. En la página de la aplicación, en la pestaña Grupos de implementación, elija Crear grupo de implementación.
5. En Nombre de grupo de implementación, introduzca un nombre que describa el grupo de implementación.

 Note

Si desea utilizar la misma configuración empleada en otro grupo de implementación (incluido el nombre del grupo de implementación; las etiquetas, los nombres de grupo de Amazon EC2 Auto Scaling y la configuración de implementación), elija esos valores en esta página. Si bien este nuevo grupo de implementación y el grupo de implementación existente tienen el mismo nombre, los CodeDeploy trata como grupos de implementación independientes, ya que están asociados a aplicaciones distintas.

6. En Función de servicio, elija una función de servicio que conceda CodeDeploy acceso a la instancia de destino.
7. En Deployment type (Tipo de implementación) elija Blue/green.
8. En Configuración del entorno, haga lo siguiente:
  - Elija el método que desea utilizar para proporcionar instancias para su entorno de sustitución: Dispone de las opciones siguientes:
    - Copiar automáticamente el grupo Auto Scaling de Amazon EC2: CodeDeploy crea un grupo de Auto Scaling de Amazon EC2 copiando el que especifique.
    - Aprovisionar instancias manualmente: no especificará las instancias de su entorno de sustitución hasta que cree una implementación. Debe crear las instancias antes de empezar la implementación. En su lugar, en este caso debe especificar las instancias que desea sustituir.
  - Si seleccionó Copiar automáticamente el grupo de Auto Scaling de Amazon EC2, si lo desea, seleccione Añadir un enlace de terminación a los grupos de Auto Scaling para CodeDeploy instalar un enlace de terminación en su grupo de Auto Scaling al crear o actualizar el grupo de despliegue. Cuando este enlace esté instalado, CodeDeploy realizará los despliegues de terminación. Para obtener más información, consulte [Habilitación de implementaciones de terminación durante los eventos de reducción horizontal de Auto Scaling](#).
9. En Configuración del agente con Systems Manager, especifique cómo desea instalar y actualizar el CodeDeploy agente en las instancias de su grupo de implementación. Para obtener más

información sobre el CodeDeploy agente, consulte [Trabajar con el CodeDeploy agente](#). Para obtener más información sobre Systems Manager, consulte [¿Qué es Systems Manager?](#)

- a. Nunca: omita la configuración de la CodeDeploy instalación con Systems Manager. Las instancias deben tener el agente instalado para poder utilizarlas en las implementaciones, por lo que solo debe elegir esta opción si va a instalar el CodeDeploy agente de otra forma.
  - b. Solo una vez: Systems Manager instalará el CodeDeploy agente una vez en cada instancia del grupo de implementación.
  - c. Ahora y programe las actualizaciones: Systems Manager creará una asociación con State Manager que instalará el CodeDeploy agente según la programación que usted configure. Para obtener más información acerca del Administrador de estados y las asociaciones, consulte [Acerca del Administrador de estados](#).
10. En función de su elección en el paso 8, lleve a cabo alguna de las siguientes operaciones:
- Si eligió Copiar automáticamente el grupo Auto Scaling de Amazon EC2: en Grupo de Amazon EC2 Auto Scaling, elija o introduzca el nombre del grupo de Amazon EC2 Auto Scaling que desee utilizar como plantilla para el grupo de Amazon EC2 Auto Scaling que se crea para las instancias del entorno de reemplazo. El número de instancias actualmente en buen estado en el grupo de Amazon EC2 Auto Scaling que seleccione se crea en su entorno de sustitución.
  - Si eligió Aprovisionar instancias manualmente: seleccione Grupos de Amazon EC2 Auto Scaling, Instancias de Amazon EC2 o ambos para especificar las instancias que se van a añadir a este grupo de implementación. Introduzca los valores de etiqueta de Amazon EC2 Auto Scaling o los nombres de grupo de Amazon EC2 Auto Scaling para identificar las instancias del entorno original (es decir, las instancias que desea reemplazar o que ejecutan la revisión de la aplicación actual).
11. En Equilibrador de carga, seleccione Habilitar equilibrio de carga y, a continuación, en las listas, seleccione los equilibradores de carga clásicos, los grupos de destino de los equilibradores de carga de aplicación y los grupos de destino de los equilibradores de carga de red con los que desee registrar las instancias de Amazon EC2 de reemplazo. Cada instancia de reemplazo se registrará con todos los equilibradores de carga clásicos y los grupos de destino seleccionados. Puede seleccionar hasta 10 equilibradores de carga clásicos y 10 grupos de destino, para un total de 20 elementos.

El tráfico se redirigirá de las instancias originales a las de reemplazo según los ajustes de Redireccionamiento de tráfico y Configuración de implementación que haya elegido.

Para obtener más información sobre los balanceadores de carga para CodeDeploy las implementaciones, consulte [Integrating CodeDeploy with Elastic Load Balancing](#)

### Warning

Si está configurando grupos de Auto Scaling y balanceadores de carga de Elastic Load Balancing en este grupo de despliegues y desea [adjuntar los balanceadores de carga a los grupos de Auto Scaling](#), le recomendamos completar este adjunto antes de crear el CodeDeploy despliegue desde este grupo de despliegues. Si intenta completar la asociación después de crear la implementación, es posible que se cancele el registro de todas las instancias en los equilibradores de carga de forma inesperada.


- En Deployment settings, revise las opciones predeterminadas para redirigir el tráfico al entorno de sustitución, qué configuración de implementación desea utilizar para la implementación y cómo se gestionan las instancias en el entorno original después de la implementación.

Si desea cambiar la configuración, continúe con el paso siguiente. De lo contrario, vaya al paso 14.

- Para cambiar la configuración de la implementación blue/green elija cualquiera de los siguientes ajustes.

| Opción                                             | Opciones                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Traffic rerouting (Nuevo enrutamiento del tráfico) | <ul style="list-style-type: none"> <li>Redirigir el tráfico inmediatamente: tan pronto como se aprovisionan las instancias del entorno de sustitución y se instalan en ellas la última revisión de la aplicación, se registran automáticamente en el equilibrador de carga, lo que hace que el tráfico se redirija a ellas. Se anula el registro de las instancias en el entorno original.</li> <li>Elegir si se debe redirigir el tráfico: las instancias del entorno de sustitución no se registran automáticamente en el equilibrador de carga a menos que redirija el tráfico manualmente. Si se agota el tiempo de</li> </ul> |



| Opción                          | Opciones                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | <p>espera especificado sin que se redirija el tráfico, el estado de la implementación cambia a Stopped.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Configuración de implementación | <p>Elija el modo en que las instancias del entorno de sustitución se registran en el entorno de sustitución con el equilibrador de carga y los grupos de destino, por ejemplo, una a una o todas a la vez.</p> <div data-bbox="862 653 1507 1108" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Una vez que el tráfico se dirige al entorno de sustitución de forma satisfactoria, se anula el registro de todas las instancias del entorno original a la vez, independientemente de la configuración de implementación que haya seleccionado.</p></div> <p>Para obtener más información, consulte <a href="#">Trabajar con configuraciones de implementación en CodeDeploy</a>.</p> |

| Opción                                     | Opciones                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Original instances (Instancias originales) | <ul style="list-style-type: none"> <li>• Terminar las instancias originales en el grupo de implementación: una vez que el tráfico se redirija al entorno de sustitución, las instancias cuyo registro se anuló en los equilibradores de carga y los grupos de destino se terminan tras el periodo de espera especificado.</li> <li>• Mantener en ejecución las instancias originales del grupo de implementación: una vez que el tráfico se redirija al entorno de sustitución, las instancias cuyo registro se anuló en los equilibradores de carga y los grupos de destino se mantienen en ejecución.</li> </ul> |

14. (Opcional) En Avanzado, configure las opciones que desee incluir en la implementación, como los activadores de notificaciones de Amazon SNS, las CloudWatch alarmas de Amazon, las opciones de Auto Scaling o las reversiones automáticas.

Para obtener información acerca de cómo especificar opciones avanzadas en grupos de implementaciones, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

15. Elija Crear grupo de implementación.

## Creación de un grupo de implementación para una implementación de Amazon ECS (consola)


1. [Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en https://console.aws.amazon.com/codedeploy.](https://console.aws.amazon.com/codedeploy)

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.

- Desde la Applications table (Tabla de aplicaciones), seleccione el nombre de la aplicación asociada al grupo de implementaciones que desea editar.
- En la página de aplicación, en Deployment groups (Grupos de implementaciones), seleccione el nombre del grupo de implementaciones que desea editar.
- En la página de la aplicación, en la pestaña Grupos de implementación, elija Crear grupo de implementación. Para obtener más información sobre lo que necesita para crear un grupo de implementación para una implementación de Amazon ECS, consulte [Antes de empezar una implementación de Amazon ECS](#).
- En Nombre de grupo de implementación, introduzca un nombre que describa el grupo de implementación.

 Note

Si desea utilizar la misma configuración empleada en otro grupo de implementación (incluido el nombre del grupo de implementación y la configuración de implementación), elija esos valores en esta página. Si bien este nuevo grupo y el grupo existente pueden tener el mismo nombre, los CodeDeploy trata como grupos de implementación independientes, ya que cada uno está asociado a una aplicación diferente.

- En Función de servicio, elija una función de servicio que conceda CodeDeploy acceso a Amazon ECS. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).
- En Nombre del equilibrador de carga, seleccione el nombre del equilibrador de carga que envía tráfico a su servicio de Amazon ECS.
- En Puerto del oyente de producción, elija el puerto y el protocolo del oyente que envía el tráfico de producción a su servicio de ECS de Amazon.
- (Opcional) En Puerto de oyente de prueba, elija el puerto y el protocolo de un oyente de prueba que envíe el tráfico a la tarea de reemplazo establecida en su servicio de Amazon ECS durante la implementación. Puede especificar una o más funciones Lambda en el AppSpec archivo que se ejecutarán durante el enlace. `AfterAllowTestTraffic` Las funciones pueden ejecutar pruebas de validación. Si la prueba de validación falla, se activa una restauración de la implementación. Si se superan las pruebas de validación, se desencadena el siguiente enlace del ciclo de vida de la implementación, `BeforeAllowTraffic`. Si no se especifica un puerto de oyente de prueba, no ocurre nada durante el enlace `AfterAllowTestTraffic`. Para obtener más información, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#).

11. Entre Nombre del grupo objetivo 1 y Nombre del grupo objetivo 2, elija los grupos de destino que se utilizarán para enrutar el tráfico durante la implementación. CodeDeploy vincula un grupo objetivo al conjunto de tareas original del servicio Amazon ECS y el otro al conjunto de tareas de reemplazo. Para obtener más información, consulte [Grupos de destino para sus equilibradores de carga de aplicación](#).
12. Elija Redirigir el tráfico inmediatamente o Especificar cuándo redirigir el tráfico para determinar cuándo redirigir el tráfico a su servicio de Amazon ECS actualizado.

Si elige Redirigir el tráfico inmediatamente, la implementación redireccionará automáticamente el tráfico una vez que se haya aprovisionado el conjunto de tareas de reemplazo.

Si elige Especificar cuándo redirigir el tráfico, elija el número de días, horas y minutos que debe esperar hasta que el conjunto de tareas de reemplazo se aprovisiona correctamente. Durante este tiempo de espera, se ejecutan las pruebas de validación en las funciones Lambda especificadas en el AppSpec archivo. Si el tiempo de espera expira antes de que se redirija el tráfico, el estado de la implementación cambia a Stopped.

13. En Finalización de la revisión original, elija el número de días, horas y minutos que se deben esperar hasta que finalice la tarea original establecida en su servicio de Amazon ECS tras una implementación correcta.
14. (Opcional) En Avanzado, configure las opciones que desee incluir en la implementación, como los activadores de notificaciones de Amazon SNS, CloudWatch las alarmas de Amazon o las reversiones automáticas.

Para obtener más información, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

## Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2

Antes de ejecutar cualquier implementación azul/verde o una implementación local para la que desea especificar un equilibrador de carga opcional en el grupo de implementación, es necesario crear un equilibrador de carga clásico, un equilibrador de carga de aplicación o un equilibrador de carga de red en Elastic Load Balancing. Para las implementaciones blue/green, debe utilizar dicho balanceador de carga para registrar las instancias que componen su entorno de sustitución. Opcionalmente, las instancias de su entorno original pueden registrarse en este mismo balanceador de carga. En el caso de las implementaciones in situ, el balanceador de carga se usa para anular

el registro de las instancias en las que se está trabajando y volver a registrarlas cuando se haya CodeDeploy completado el trabajo.

CodeDeploy admite la implementación azul/verde y in situ en instancias de Amazon EC2 detrás de varios balanceadores de carga. Por ejemplo, supongamos que tiene 200 instancias de Amazon EC2, de las que 100 están registradas en 2 equilibradores de carga clásicos y otras 100 están registradas con 4 grupos de destino en 2 equilibradores de carga de aplicación. En este escenario, le CodeDeploy permitirá realizar despliegues azules o verdes e in situ en las 200 instancias, aunque estén distribuidas en 2 balanceadores de carga clásicos, 2 balanceadores de carga de aplicaciones y 4 grupos objetivo.

CodeDeploy admite hasta 10 balanceadores de carga clásicos y 10 grupos objetivo, con un total de 20 elementos.

Para configurar uno o más equilibradores de carga clásicos, siga las instrucciones de [Tutorial: Creación de un equilibrador de carga clásico](#) en la Guía del usuario de los equilibradores de carga clásicos. Tenga en cuenta lo siguiente:

- En el Paso 2: Definir el balanceador de carga, en Crear LB dentro, elija la misma VPC que seleccionó al crear las instancias.
- En el Paso 5: Registrar instancias EC2 en su balanceador de carga, seleccione las instancias que se encuentren actualmente en su grupo de implementaciones (implementaciones in situ) o que ha designado que estén en su entorno original (implementaciones blue/green).
- En el Paso 7: Crear y probar su balanceador de carga, anote la dirección DNS del balanceador de carga.

Por ejemplo, si denominó su balanceador de carga `my-load-balancer`, su dirección DNS aparecerá en un formato como `my-load-balancer-1234567890.us-east-2.elb.amazonaws.com`.

Para configurar uno o más equilibradores de carga de aplicación, siga las instrucciones de uno de los siguientes temas:

- [Creación de un equilibrador de carga de aplicación](#)
- [Tutorial: Creación de un Application Load Balancer con AWS CLI](#)

Para configurar uno o más equilibradores de carga de red, siga las instrucciones de uno de los siguientes temas:

- [Creación de un equilibrador de carga de red](#)
- [Tutorial: Creación de un Network Load Balancer con AWS CLI](#)

## Configure un balanceador de carga, grupos objetivo y agentes de escucha para las implementaciones de CodeDeploy Amazon ECS

Antes de ejecutar una implementación con la plataforma de informática de Amazon ECS, debe crear un equilibrador de carga de aplicación o un equilibrador de carga de red, dos grupos de destino y uno o dos oyentes. En este tema se muestra cómo crear un equilibrador de carga de aplicación. Para obtener más información, consulte [Antes de empezar una implementación de Amazon ECS](#).

Uno de los grupos de destino dirige el tráfico al conjunto de tareas original de la aplicación de Amazon ECS. El otro grupo de destino dirige el tráfico a su conjunto de tareas de sustitución. Durante la implementación, CodeDeploy crea un conjunto de tareas de reemplazo y redirige el tráfico del conjunto de tareas original al nuevo. CodeDeploy determina qué grupo objetivo se utiliza para cada conjunto de tareas.

El equilibrador de carga utiliza un oyente para dirigir el tráfico hacia los grupos de destino. Es obligatorio un agente de escucha de producción. Puede especificar un agente de escucha de prueba opcional que dirija el tráfico hacia conjunto de tareas de sustitución mientras ejecuta las pruebas de validación.

El equilibrador de carga debe utilizar una VPC con dos subredes públicas en diferentes zonas de disponibilidad. Los siguientes pasos muestran cómo confirmar la VPC predeterminada, crear un equilibrador de carga de aplicación de Amazon EC2 y, a continuación, crear dos grupos de destino para el equilibrador de carga. Para obtener más información, consulte [Grupos de destino de los equilibradores de carga de red](#).

### Verificación de la VPC predeterminada, las subredes públicas y el grupo de seguridad

En este tema se muestra cómo crear un equilibrador de carga de aplicación de Amazon EC2, dos grupos de destino y dos puertos que se pueden utilizar durante una implementación de Amazon ECS. Uno de los puertos es opcional y solo es necesario si dirige el tráfico a un puerto de prueba para las pruebas de validación durante la implementación.

1. [Inicie sesión en la consola de Amazon VPC AWS Management Console y ábrala en https://console.aws.amazon.com/vpc/.](https://console.aws.amazon.com/vpc/)

2. Compruebe la VPC predeterminada que se va a utilizar. En el panel de navegación, elija Your VPCs (Sus VPC). Fíjese en qué VPC muestra Yes (Sí) en la columna Default VPC (VPC predeterminada). Esta será su VPC predeterminada. Contiene las subredes predeterminadas que utiliza.
3. Elija Subnets (Subredes). Anote los ID de subred de dos subredes que muestran Yes (Sí) en la columna Default subnet (Subred predeterminada). Puede utilizar estos ID al crear el balanceador de carga.
4. Elija cada subred y, a continuación, elija la pestaña Description (Descripción). Compruebe que las subredes que desea utilizar se encuentran en diferentes zonas de disponibilidad.
5. Elija las subredes y, a continuación, elija la pestaña Route Table (Tabla de ruteo). Para verificar que cada subred que desea utilizar es una subred pública, confirme que la tabla de enrutamiento incluye una fila con un enlace a una gateway de Internet.
6. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
7. En el panel de navegación, elija Security Groups (Grupos de seguridad).
8. Compruebe que el grupo de seguridad que desee utilizar esté disponible y anote su ID de grupo (por ejemplo, sg-abcd1234). Puede utilizarlo al crear el balanceador de carga.

## Creación de un equilibrador de carga de aplicación de Amazon EC2, dos grupos de destino y oyentes (consola)

Para utilizar la consola de Amazon EC2 para crear un equilibrador de carga de aplicación de Amazon EC2:

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. En el panel de navegación, seleccione Equilibradores de carga.
3. Elija Create Load Balancer (Crear equilibrador de carga).
4. Elija Equilibrador de carga de aplicación y a continuación, Crear.
5. En Nombre, especifique el nombre del equilibrador de carga.
6. En Scheme (Esquema), elija Internet-facing (Expuesto a Internet).
7. En IP address type (Tipo de dirección IP), elija ipv4.


8. Configure un segundo puerto de agente de escucha para el balanceador de carga (opcional). Puede ejecutar pruebas de validación de implementación con el tráfico de prueba que se ofrece a este puerto.
  - a. En Load Balancer Protocol (Protocolo de balanceador de carga), elija Add listener (Añadir agente de escucha).
  - b. En Protocolo de equilibrador de carga del segundo agente de escucha, elija HTTP.
  - c. En Puerto del equilibrador de carga, escriba **8080**.
9. En Availability Zones (Zonas de disponibilidad), en VPC, elija la VPC predeterminada y, a continuación, seleccione las dos subredes predeterminadas que desea utilizar.
10. Elija Next: Configure Security Settings (Siguiendo: Establecer la configuración de seguridad).
11. Elija Next: Configure Security Groups (Siguiendo: configurar grupos de seguridad).
12. Elija Select an existing security group (Seleccionar un grupo de seguridad existente), después el grupo de seguridad predeterminado y, a continuación, anote su ID.
13. Elija Next: Configure Routing (Siguiendo: Configuración del enrutamiento).
14. En Target group (Grupo de destino), elija New target group (Nuevo grupo de destino) y configure su primer grupo de destino:
  - a. En Name (Nombre), introduzca un nombre de grupo de destino (por ejemplo, **target-group-1**).
  - b. En Target type (Tipo de destino), elija IP.
  - c. En Protocol (Protocolo), elija HTTP. En Port (Puerto), introduzca **80**.
  - d. Elija Next: Register Targets (Siguiendo: Registrar destinos).
15. Elija Next: Review (Siguiendo: análisis) y, a continuación, seleccione Create (Crear).

Para crear un segundo grupo de destino para el equilibrador de carga

1. Después de aprovisionar el equilibrador de carga, abra la consola de Amazon EC2. En el panel de navegación, elija Target Groups.
2. Elija Crear grupo de destino.
3. En Name (Nombre), introduzca un nombre de grupo de destino (por ejemplo, **target-group-2**).
4. En Target type (Tipo de destino), elija IP.
5. En Protocol (Protocolo), elija HTTP. En Port (Puerto), introduzca **80**.



6. En VPC, elija la VPC predeterminada.
7. Seleccione Crear.

 Note

Para que la implementación de Amazon ECS se ejecute, debe tener dos grupos de destino creados para el equilibrador de carga. Puede utilizar el ARN de uno de los grupos de destino al crear su servicio de Amazon ECS. Para obtener más información, consulte [Paso 4: Creación de un servicio de Amazon ECS](#) en la Guía del usuario de Amazon ECS.

## Creación de un equilibrador de carga de aplicación de Amazon EC2, dos grupos de destino y oyentes (CLI)

Para crear un equilibrador de carga de aplicación con la AWS CLI:

1. Utilice el [create-load-balancer](#) comando para crear un Application Load Balancer. Especifique dos subredes que no estén en la misma zona de disponibilidad y un grupo de seguridad.

```
aws elbv2 create-load-balancer --name bluegreen-alb \
--subnets subnet-abcd1234 subnet-abcd5678 --security-groups sg-abcd1234 --
region us-east-1
```

El resultado contiene el nombre de recurso de Amazon (ARN) del balanceador de carga con el siguiente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642
```

2. Utilice el [create-target-group](#) comando para crear su primer grupo objetivo. CodeDeploy dirige el tráfico de este grupo objetivo al conjunto de tareas original o de reemplazo de su servicio.

```
aws elbv2 create-target-group --name bluegreentarget1 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

El resultado contiene el ARN del primer grupo de destino con el siguiente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4
```

3. Usa el [create-target-group](#) comando para crear tu segundo grupo objetivo. CodeDeploy enruta el tráfico del grupo objetivo al conjunto de tareas que no atiende el primer grupo objetivo. Por ejemplo, si su primer grupo de destino dirige el tráfico al conjunto de tareas original, este grupo de destino dirige el tráfico al conjunto de tareas de sustitución.

```
aws elbv2 create-target-group --name bluegreentarget2 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

El resultado contiene el ARN del segundo grupo de destino con el siguiente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4
```

4. Utilice el comando [create-listener](#) para crear un agente de escucha con una regla predeterminada que reenvíe el tráfico de producción al puerto 80.

```
aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
--protocol HTTP --port 80 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4 --region us-east-1
```

El resultado incluye el ARN del agente de escucha con el siguiente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

5. Utilice el comando [create-listener](#) para crear un segundo agente de escucha con una regla predeterminada que reenvíe el tráfico de prueba al puerto 8080 (opcional). Puede ejecutar pruebas de validación de implementación con el tráfico de prueba que se ofrece a este puerto.

```
aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
```

```
--protocol HTTP --port 8080 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/209a844cd01825a4 --region us-east-1
```

El resultado incluye el ARN del agente de escucha con el siguiente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

## Creación de un grupo de implementación (CLI)

Para usar el AWS CLI para crear un grupo de despliegues, ejecuta el [create-deployment-group](#) comando y especifica:

- El nombre de la aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
- Un nombre para el grupo de implementaciones. Se creará un grupo de implementaciones con este nombre para la aplicación especificada. Un grupo de implementación solo se puede asociar a una aplicación.
- Información sobre las etiquetas, grupos de etiquetas o nombres de grupo de Amazon EC2 Auto Scaling que identifican las instancias que se deben incluir en el grupo de implementación.
- El identificador del nombre de recurso de Amazon (ARN) del rol de servicio que permite CodeDeploy actuar en nombre de su AWS cuenta al interactuar con otros AWS servicios. Para obtener el ARN del rol de servicio, consulte [Obtención del ARN del rol de servicio \(CLI\)](#). Para obtener más información sobre los roles de servicio, consulte [Términos y conceptos sobre los roles](#) en la Guía del usuario de IAM.
- Información sobre el tipo de implementación, ya sea una implementación in situ o blue/green, que se desea asociar al grupo de implementaciones.
- (Opcional) El nombre de una configuración de implementación existente. Para ver una lista de configuraciones de implementación, consulte [View Deployment Configuration Details](#). Si no se especifica, CodeDeploy utiliza una configuración de despliegue predeterminada.
- (Opcional) Comandos para crear un activador que inserta notificaciones sobre eventos de implementaciones e instancias a los que están suscritos a un tema de Amazon Simple Notification Service. Para obtener más información, consulte [Monitoring Deployments with Amazon SNS Event Notifications](#).

- (Opcional) Comandos para añadir CloudWatch las alarmas existentes al grupo de despliegue que se activan si una métrica especificada en una alarma cae por debajo o supera un umbral definido.
- (Opcional) Comandos para que una implementación vuelva a la última revisión válida conocida cuando una implementación falle o se active una CloudWatch alarma.
- (Opcional) Comandos para una implementación para generar enlaces de eventos del ciclo de vida durante un evento de escalado interno de Auto Scaling. Para obtener más información, consulte [Cómo funciona Amazon EC2 Auto Scaling con CodeDeploy](#).
- Para las implementaciones in situ:
  - (Opcional) Los nombres de los equilibradores de carga clásicos, equilibradores de carga de aplicaciones o equilibradores de carga de red de Elastic Load Balancing que administran el tráfico a las instancias durante los procesos de implementación.
- Para las implementaciones blue/green:
  - Configuración del proceso de implementación blue/green:
    - Cómo se aprovisionan las nuevas instancias en el entorno de sustitución.
    - Si se redirige el tráfico hacia el entorno de sustitución de inmediato o se espera un período de tiempo especificado para redirigir el tráfico manualmente.
    - Si se deben terminar las instancias del entorno original.
  - Los nombres de los equilibradores de carga clásicos, equilibradores de carga de aplicación o equilibradores de carga de red en Elastic Load Balancing que se utilizarán para las instancias registradas en el entorno de reemplazo.

#### Warning

Si está configurando un grupo de Auto Scaling y un balanceador de cargas de Elastic Load Balancing en su grupo de despliegue y desea [adjuntar el balanceador de carga al grupo de Auto Scaling](#), le recomendamos que complete este adjunto antes de crear el CodeDeploy despliegue desde este grupo de despliegues. Si intenta completar la asociación después de crear la implementación, es posible que se cancele el registro de todas las instancias en el equilibrador de carga de forma inesperada.

## Vea los detalles del grupo de implementación con CodeDeploy

Puede usar la CodeDeploy consola AWS CLI, las API o las CodeDeploy API para ver los detalles de todos los grupos de implementación asociados a una aplicación.

## Temas

- [Visualización de los detalles de un grupo de implementación \(consola\)](#)
- [Visualización de los detalles de un grupo de implementación \(CLI\)](#)

## Visualización de los detalles de un grupo de implementación (consola)

Para usar la CodeDeploy consola para ver los detalles del grupo de implementación:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la página Applications, elija el nombre de aplicación asociado con el grupo de implementaciones.

### Note

Si no se muestra ninguna entrada, asegúrese de seleccionar la región correcta. En la barra de navegación, en el selector de regiones, elija una de las regiones que aparecen en [Región y puntos finales](#) en. Referencia general de AWS CodeDeploy solo se admite en estas regiones.

4. Para ver los detalles de un solo grupo de implementación, en la pestaña Grupos de implementación, elija el nombre del grupo de implementación.

## Visualización de los detalles de un grupo de implementación (CLI)

Para usar el AWS CLI para ver los detalles del grupo de implementación, llame al `get-deployment-group` comando o al `list-deployment-groups` comando.

Para ver los detalles de un único grupo de despliegues, ejecuta el [get-deployment-group](#) comando y especifica:

- El nombre de la aplicación asociada con el grupo de implementaciones. Para obtener el nombre de la aplicación, llame al comando [list-applications](#).
- El nombre del grupo de implementaciones. Para obtener el nombre del grupo de implementación, [list-deployment-groups](#) ejecute el comando.

Para ver una lista de nombres de grupos de despliegue, ejecute el [list-deployment-groups](#) comando y especifique el nombre de la aplicación asociado a los grupos de despliegue. Para obtener el nombre de la aplicación, llame al comando [list-applications](#).

## Cambie la configuración del grupo de implementación con CodeDeploy

Puede usar la CodeDeploy consola AWS CLI, las API o las CodeDeploy API para cambiar la configuración de un grupo de implementación.

### Warning

No sigas estos pasos si quieres que el grupo de despliegues use un grupo de despliegues not-yet-created personalizado. En lugar de ello, siga las instrucciones en [Create a Deployment Configuration](#) y, a continuación, vuelva a este tema. No utilice estos pasos si desea que el grupo de implementación utilice un rol de not-yet-created servicio diferente. El rol de servicio debe confiar CodeDeploy, como mínimo, en los permisos descritos en [Paso 2: Crear un rol de servicio para CodeDeploy](#). Para crear y configurar un rol de servicio con los permisos correctos, siga las instrucciones de [Paso 2: Crear un rol de servicio para CodeDeploy](#) y, a continuación, vuelva a este tema.


### Temas

- [Cambie la configuración del grupo de implementación \(consola\)](#)
- [Cambie la configuración del grupo de implementación \(CLI\)](#)

## Cambie la configuración del grupo de implementación (consola)


Para usar la CodeDeploy consola para cambiar la configuración del grupo de implementación:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la lista de aplicaciones, elija el nombre de la aplicación que está asociada al grupo de implementaciones que desea cambiar.

 Note

Si no se muestra ninguna entrada, asegúrese de seleccionar la región correcta. En la barra de navegación, en el selector de regiones, elija una de las regiones que aparecen en [Región y puntos finales](#) en el Referencia general de AWS. CodeDeploy solo se admite en estas regiones.

4. Elija la pestaña Grupos de implementación y, a continuación, elija el nombre del grupo de implementación que desea modificar.
5. En la página Deployment group (Grupo de implementaciones), elija Edit (Editar).
6. Edite las opciones del grupo de implementaciones según sea necesario.

Para obtener más información sobre los componentes del grupo de implementaciones, consulte [Cree un grupo de despliegue con CodeDeploy](#).

7. Elija Guardar cambios.

## Cambie la configuración del grupo de implementación (CLI)

Para usar el AWS CLI para cambiar la configuración del grupo de implementación, [update-deployment-group](#) ejecute el comando y especifique:

- Para despliegues de EC2/on-premise y AWS Lambda:
  - El nombre de la aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).

- El nombre actual del grupo de implementación. Para ver una lista de los nombres de los grupos de implementación, ejecute el comando. [list-deployment-groups](#)
- (Opcional) Un nombre diferente para el grupo de implementaciones.
- (Opcional) Un nombre de recurso de Amazon (ARN) diferente que corresponda a un rol de servicio que permita CodeDeploy actuar en nombre de su AWS cuenta al interactuar con otros AWS servicios. Para obtener el ARN del rol de servicio, consulte [Obtención del ARN del rol de servicio \(CLI\)](#). Para obtener más información sobre los roles de servicio, consulte [Términos y conceptos sobre los roles](#) en la Guía del usuario de IAM.
- (Opcional) El nombre de la configuración de implementación. Para ver una lista de configuraciones de implementación, consulte [View Deployment Configuration Details](#). (Si no se especifica, CodeDeploy utiliza una configuración de despliegue predeterminada).
- (Opcional) Comandos para añadir una o más CloudWatch alarmas existentes al grupo de despliegue que se activan si una métrica especificada en una alarma cae por debajo o supera un umbral definido.
- (Opcional) Comandos para que una implementación vuelva a la última revisión válida conocida cuando una implementación falle o se active una CloudWatch alarma.
- (Opcional) Comandos para una implementación para generar enlaces de eventos del ciclo de vida durante un evento de escalado interno de Auto Scaling. Para obtener más información, consulte [Cómo funciona Amazon EC2 Auto Scaling con CodeDeploy](#).
- (Opcional) Comandos para crear o actualizar un disparador que publica en un tema en Amazon Simple Notification Service, para que los suscriptores de ese tema reciban notificaciones sobre eventos de implementaciones e instancias en este grupo de implementación. Para obtener más información, consulte [Monitoring Deployments with Amazon SNS Event Notifications](#).
- Solo para las implementaciones de EC2/en las instalaciones:
  - (Opcional) Etiquetas o grupos de etiquetas de sustitución que identifican de forma exclusiva las instancias que se deben incluir en el grupo de implementaciones.
  - (Opcional) Los nombres de los grupos de sustitución de Amazon EC2 Auto Scaling que se van a añadir al grupo de implementación.
- Solo para las implementaciones de Amazon ECS:
  - El servicio de Amazon ECS que se va a implementar.
  - Información del equilibrador de carga, incluido el Equilibrador de carga de aplicación o el Equilibrador de carga de red, los grupos de destino requeridos para una implementación de Amazon ECS e información del oyente de producción y de prueba opcional.



# Configuración de las opciones avanzadas de un grupo de implementación

Al crear o actualizar un grupo de implementación, puede configurar una serie de opciones para proporcionar mayor control y supervisión de las implementaciones de dicho grupo de implementación.

Utilice la información de esta página para ayudarle a configurar las opciones avanzadas cuando trabaje con grupos de implementaciones en los siguientes temas:

- [Cree una aplicación con CodeDeploy](#)
- [Cree un grupo de despliegue con CodeDeploy](#)
- [Cambie la configuración del grupo de implementación con CodeDeploy](#)

Activadores de notificaciones de Amazon SNS: puede añadir activadores a un grupo de CodeDeploy implementaciones para recibir notificaciones sobre eventos relacionados con las implementaciones de ese grupo de implementaciones. Estas notificaciones se envían a los destinatarios que están suscritos a un tema de Amazon SNS que forma parte de la acción del activador.

Debe haber configurado ya el tema de Amazon SNS al que apuntará este desencadenante y CodeDeploy debe tener permiso para publicar en el tema desde este grupo de implementaciones. Si aún no ha completado estos pasos de configuración, puede añadir disparadores al grupo de implementaciones más adelante.

Si desea crear un disparador ahora para recibir notificaciones sobre los eventos de implementación del grupo de implementaciones de esta aplicación, elija Create trigger.

Si la implementación es para una instancia de Amazon EC2, puede crear y recibir notificaciones sobre las instancias.

Para obtener más información, consulte [Monitoring Deployments with Amazon SNS Event Notifications](#).

CloudWatch Alarmas de Amazon: puedes crear una CloudWatch alarma que controle una única métrica durante un período de tiempo que especifiques y realice una o más acciones en función del valor de la métrica en relación con un umbral determinado durante varios períodos de tiempo. Para una implementación de Amazon EC2, puede crear una alarma para una instancia o grupo

de Auto Scaling de Amazon EC2 que esté utilizando CodeDeploy en sus operaciones. Para una implementación de AWS Lambda y Amazon ECS, puede crear una alarma para detectar errores en una función de Lambda.

Puedes configurar un despliegue para que se detenga cuando una CloudWatch alarma de Amazon detecte que una métrica ha caído por debajo o superado un umbral definido.

Debe haber creado ya la alarma para CloudWatch poder añadirla a un grupo de despliegues.

1. Para añadir la monitorización de alarmas al grupo de implementaciones, en Alarms (Alarmas), elija Add alarm (Añadir alarma).
2. Introduzca el nombre de una CloudWatch alarma que ya haya configurado para supervisar este despliegue.

Debe introducir la CloudWatch alarma exactamente como se creó en CloudWatch. Para ver una lista de alarmas, abra la CloudWatch consola en [yhttps://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/), a continuación, seleccione ALARM.

Opciones adicionales:

- Si quiere que las implementaciones procedan sin tener en cuenta las alarmas de cuenta que haya añadido, elija Ignore alarm configuration.

Esta opción resulta útil si desea desactivar temporalmente la monitorización de alarmas de un grupo de implementaciones sin tener que añadir las mismas alarmas de nuevo más tarde.

- (Opcional) Si quieres que las implementaciones continúen en caso de que CodeDeploy no puedas recuperar el estado de la alarma de Amazon CloudWatch, selecciona Continuar las implementaciones aunque el estado de la alarma no esté disponible.


#### Note

Esta opción corresponde al [AlarmConfiguration](#) objeto ignorePollAlarmFailure de la API. CodeDeploy

Para obtener más información, consulte [Supervisión de las implementaciones con CloudWatch alarmas en CodeDeploy](#).

Restauraciones automáticas: puede configurar una implementación o un grupo de implementaciones para que se restauren automáticamente si falla una implementación o si se supera un umbral de monitorización que haya especificado. En este caso, se implementa la última versión correcta conocida de una revisión de la aplicación. Puede configurar ajustes opcionales para un grupo de implementaciones cuando utilice la consola para crear una aplicación y crear o actualizar un grupo de implementaciones. Al crear una nueva implementación, también puede optar por anular la configuración de restauración automática especificada para el grupo de implementaciones.

- Puede habilitar la restauración de las implementaciones a la versión correcta conocida más reciente si ocurre algún problema eligiendo una o ambas de las opciones siguientes:
  - Retrocede cuando se produce un error en una implementación. CodeDeploy volverá a implementar la última revisión válida conocida como una nueva implementación.
  - Restaurar cuando se alcancen los umbrales de alarma. Si ha añadido una alarma a esta aplicación en el paso anterior, CodeDeploy volverá a implementar la última revisión válida conocida cuando se active una o más de las alarmas especificadas.

 Note

Para no tener en cuenta temporalmente una configuración de restauración, elija `Disable rollbacks`. Esta opción resulta útil si desea deshabilitar temporalmente las restauraciones automáticas sin tener que configurar la misma configuración de nuevo más tarde.

Para obtener más información, consulte [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

Actualizaciones automáticas de instancias desactualizadas: en determinadas circunstancias, CodeDeploy puede implementar una revisión desactualizada de su aplicación en sus instancias de Amazon EC2. Por ejemplo, si sus instancias EC2 se lanzan a un grupo de Auto Scaling (ASG) mientras se está realizando una CodeDeploy implementación, esas instancias reciben la versión anterior de la aplicación en lugar de la última. Para actualizar esas instancias, inicia CodeDeploy automáticamente una implementación de seguimiento (inmediatamente después de la primera) para actualizar las instancias desactualizadas. Si desea cambiar este comportamiento predeterminado para que las instancias EC2 desactualizadas permanezcan en la revisión anterior, puede hacerlo a través de la CodeDeploy API o la AWS Command Line Interface (CLI).

Para configurar las actualizaciones automáticas de instancias desactualizadas a través de la API, incluye el parámetro de solicitud `outdatedInstancesStrategy` en la acción `UpdateDeploymentGroup` o `CreateDeploymentGroup`. Para obtener información detallada, consulte la Referencia de la API de AWS CodeDeploy .

Para configurar las actualizaciones automáticas mediante AWS CLI, utilice uno de los siguientes comandos:

```
aws deploy update-deployment-group arguments --outdated-instances-strategy
UPDATE|IGNORE
```

O bien...

```
aws deploy create-deployment-group arguments --outdated-instances-strategy
UPDATE|IGNORE
```

... donde *argumentos* se sustituye por los argumentos necesarios para la implementación y *UPDATE|IGNORE* se sustituye por `UPDATE` para habilitar las actualizaciones automáticas o por `IGNORE` para deshabilitarlas.

Ejemplo:

```
aws deploy update-deployment-group --application-name "MyApp" --current-
deployment-group-name "MyDG" --region us-east-1 --outdated-instances-
strategy IGNORE
```

Para obtener más información sobre estos AWS CLI comandos, consulte la Referencia de AWS CLI comandos.

## Eliminar un grupo de despliegue con CodeDeploy

Puede usar la CodeDeploy consola AWS CLI, las API o las CodeDeploy API para eliminar los grupos de despliegue asociados a su AWS cuenta.

### Warning

Si eliminas un grupo de despliegues, también se eliminarán todos los detalles asociados a ese grupo de despliegues CodeDeploy. Las instancias utilizadas en el grupo de implementaciones no cambiarán. Esta acción no se puede deshacer.

## Temas

- [Eliminar un grupo de implementación \(consola\)](#)
- [Eliminar un grupo de implementación \(CLI\)](#)

## Eliminar un grupo de implementación (consola)

Para usar la CodeDeploy consola para eliminar un grupo de despliegues:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la lista de aplicaciones, seleccione el nombre de la aplicación asociada al grupo de implementaciones.
4. En la página Application details (Detalles de la aplicación), en la pestaña Deployment groups (Grupos de implementaciones), elija el nombre del grupo de implementaciones que desea eliminar.
5. En la página Deployment details (Detalles de la implementación), elija Delete (Eliminar).
6. Cuando se le solicite, escriba el nombre del grupo de implementaciones que desea eliminar y, a continuación, seleccione Delete.

## Eliminar un grupo de implementación (CLI)

Para usar el AWS CLI para eliminar un grupo de implementación, ejecuta el [delete-deployment-group](#) comando y especifica:

- El nombre de la aplicación asociada al grupo de implementaciones. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
- El nombre del grupo de implementación asociado a la aplicación. Para ver una lista de los nombres de los grupos de despliegue, [list-deployment-groups](#) ejecute el comando.

# Trabajar con revisiones de aplicaciones para CodeDeploy

En CodeDeploy, una revisión contiene una versión de los archivos fuente que CodeDeploy se implementarán en las instancias o los scripts CodeDeploy se ejecutarán en las instancias.

Planifica la revisión, añade un AppSpec archivo a la revisión y, a continuación, envía la revisión a Amazon S3 o GitHub. Una vez enviada la revisión, puede implementarla.

## Temas

- [Planifique una revisión para CodeDeploy](#)
- [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#)
- [Elige un tipo de CodeDeploy repositorio](#)
- [Enviar una revisión CodeDeploy a Amazon S3 \(solo para implementaciones de EC2/on-premise\)](#)
- [Vea los detalles de la revisión de la aplicación con CodeDeploy](#)
- [Registre una revisión de aplicación en Amazon S3 con CodeDeploy](#)

## Planifique una revisión para CodeDeploy

Con unos buenos planes es mucho más fácil implementar las revisiones.

Para las implementaciones en una plataforma informática AWS Lambda o Amazon ECS, una revisión es lo mismo AppSpec que el archivo. La siguiente información no es aplicable. Para obtener más información, consulte [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).

Para las implementaciones en una plataforma de informática EC2/en las instalaciones, comience por crear un directorio raíz (carpeta) vacío en la máquina de desarrollo. Ahí es donde almacenará los archivos de código fuente (como los archivos de texto y binarios, los ejecutables, los paquetes, etc.) que se van a implementan en las instancias o los scripts que se van a ejecutar en las instancias.

Por ejemplo, en la carpeta raíz /tmp/ en Linux, macOS o Unix o en la carpeta raíz c:\temp en Windows:

```
/tmp/ or c:\temp (root folder)
|--content (subfolder)
| |--myTextFile.txt
```

```
| |--mySourceFile.rb
| |--myExecutableFile.exe
| |--myInstallerFile.msi
| |--myPackage.rpm
| |--myImageFile.png
|--scripts (subfolder)
| |--myShellScript.sh
| |--myBatchScript.bat
| |--myPowerShellScript.ps1
|--appspec.yml
```

La carpeta raíz también debe incluir un archivo de especificaciones de la aplicación (AppSpec archivo), como se muestra aquí. Para obtener más información, consulte [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).

## Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy

En este tema se muestra cómo añadir un AppSpec archivo a la implementación. También incluye plantillas para crear un AppSpec archivo para una implementación de AWS Lambda y EC2/on-premise.

### Temas

- [Añadir un AppSpec archivo para una implementación de Amazon ECS](#)
- [Agregar un AppSpec archivo para una implementación de AWS Lambda](#)
- [Agregue un AppSpec archivo para una implementación local de EC2/](#)

## Añadir un AppSpec archivo para una implementación de Amazon ECS

Para una implementación en una plataforma de informática de Amazon ECS:

- El AppSpec archivo especifica la definición de tareas de Amazon ECS utilizada para la implementación, el nombre del contenedor y la asignación de puertos que se utilizan para enrutar el tráfico, y las funciones Lambda opcionales que se ejecutan después de los eventos del ciclo de vida de la implementación.
- Una revisión es lo mismo que un AppSpec archivo.
- Un AppSpec archivo se puede escribir con JSON o YAML.

- Un AppSpec archivo puede guardarse como archivo de texto o introducirse directamente en una consola al crear una implementación. Para obtener más información, consulte [Creación de una implementación en una plataforma de informática de Amazon ECS \(consola\)](#).

Para crear un AppSpec archivo

1. Copia la plantilla JSON o YAML en un editor de texto o en el AppSpec editor de la consola.
2. Modifique la plantilla como sea necesario.
3. Usa un validador de JSON o YAML para validar tu archivo. AppSpec Si utilizas el AppSpec editor, el archivo se valida al elegir Crear despliegue.
4. Si utiliza un editor de texto, guarde el archivo. Si utilizas el AWS CLI para crear tu implementación, consulta el AppSpec archivo si está en tu disco duro o en un bucket de Amazon S3. Si usa la consola, debe enviar el AppSpec archivo a Amazon S3.

## Plantilla de AppSpec archivo YAML para una implementación de Amazon ECS con instrucciones

La siguiente es una plantilla YAML de un AppSpec archivo para una implementación de Amazon ECS con todas las opciones disponibles. Para obtener más información acerca de los eventos del ciclo de vida que se deben utilizar en la sección hooks, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#).

```
This is an appspec.yml template file for use with an Amazon ECS deployment in
CodeDeploy.
The lines in this template that start with the hashtag are
comments that can be safely left in the file or
ignored.
For help completing this file, see the "AppSpec File Reference" in the
"CodeDeploy User Guide" at
https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
In the Resources section, you must specify the following: the Amazon ECS service,
task definition name,
and the name and port of the load balancer to route traffic,
target version, and (optional) the current version of your AWS Lambda function.
Resources:
 - TargetService:
 Type: AWS::ECS::Service
 Properties:
```



```

 TaskDefinition: "" # Specify the ARN of your task definition
 (arn:aws:ecs:region:account-id:task-definition/task-definition-family-name:task-
 definition-revision-number)
 LoadBalancerInfo:
 ContainerName: "" # Specify the name of your Amazon ECS application's
 container
 ContainerPort: "" # Specify the port for your container where traffic
 reroutes
Optional properties
 PlatformVersion: "" # Specify the version of your Amazon ECS Service
 NetworkConfiguration:
 AwsvpcConfiguration:
 Subnets: ["", ""] # Specify one or more comma-separated subnets in your
 Amazon ECS service
 SecurityGroups: ["", ""] # Specify one or more comma-separated security
 groups in your Amazon ECS service
 AssignPublicIp: "" # Specify "ENABLED" or "DISABLED"
(Optional) In the Hooks section, specify a validation Lambda function to run during
a lifecycle event.
Hooks:
Hooks for Amazon ECS deployments are:
- BeforeInstall: "" # Specify a Lambda function name or ARN
- AfterInstall: "" # Specify a Lambda function name or ARN
- AfterAllowTestTraffic: "" # Specify a Lambda function name or ARN
- BeforeAllowTraffic: "" # Specify a Lambda function name or ARN
- AfterAllowTraffic: "" # Specify a Lambda function name or ARN

```

## AppSpec Archivo JSON para una plantilla de despliegue de Amazon ECS

La siguiente es una plantilla JSON para un AppSpec archivo para una implementación de Amazon ECS con todas las opciones disponibles. Para ver las instrucciones de plantilla, consulte los comentarios de la versión YAML en la sección anterior. Para obtener más información acerca de los eventos del ciclo de vida que se deben utilizar en la sección hooks, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#).

```

{
 "version": 0.0,
 "Resources": [
 {
 "TargetService": {
 "Type": "AWS::ECS::Service",
 "Properties": {
 "TaskDefinition": "",

```

```
 "LoadBalancerInfo": {
 "ContainerName": "",
 "ContainerPort":
 },
 "PlatformVersion": "",
 "NetworkConfiguration": {
 "AwsvpcConfiguration": {
 "Subnets": [
 "",
 ""
],
 "SecurityGroups": [
 "",
 ""
],
 "AssignPublicIp": ""
 }
 }
 }
},
"Hooks": [
 {
 "BeforeInstall": ""
 },
 {
 "AfterInstall": ""
 },
 {
 "AfterAllowTestTraffic": ""
 },
 {
 "BeforeAllowTraffic": ""
 },
 {
 "AfterAllowTraffic": ""
 }
]
```

## Agregar un AppSpec archivo para una implementación de AWS Lambda

Para una implementación en una plataforma de procesamiento AWS Lambda:

- El AppSpec archivo contiene instrucciones sobre las funciones Lambda que se van a implementar y utilizar para la validación de la implementación.
- Una revisión es lo mismo que un AppSpec archivo.
- Un AppSpec archivo se puede escribir con JSON o YAML.
- Un AppSpec archivo puede guardarse como archivo de texto o introducirse directamente en un AppSpec editor de consola al crear una implementación. Para obtener más información, consulte [Creación de una implementación en la plataforma de informática de AWS Lambda \(consola\)](#).

Para crear un AppSpec archivo:

1. Copia la plantilla JSON o YAML en un editor de texto o en el AppSpec editor de la consola.
2. Modifique la plantilla como sea necesario.
3. Usa un validador de JSON o YAML para validar tu archivo. AppSpec Si utilizas el AppSpec editor, el archivo se valida al elegir Crear despliegue.
4. Si utiliza un editor de texto, guarde el archivo. Si utilizas el AWS CLI para crear tu implementación, consulta el AppSpec archivo si está en tu disco duro o en un bucket de Amazon S3. Si usa la consola, debe enviar el AppSpec archivo a Amazon S3.

### Plantilla de AppSpec archivo YAML para una AWS Lambda implementación con instrucciones

Para obtener más información acerca de los eventos del ciclo de vida que se deben utilizar en la sección de enlaces, consulte [AppSpec sección de «ganchos» para una implementación de AWS Lambda](#).

```
This is an appspec.yml template file for use with an AWS Lambda deployment in
CodeDeploy.
The lines in this template starting with the hashtag symbol are
instructional comments and can be safely left in the file or
ignored.
For help completing this file, see the "AppSpec File Reference" in the
```

```
"CodeDeploy User Guide" at
https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
In the Resources section specify the name, alias,
target version, and (optional) the current version of your AWS Lambda function.
Resources:
 - MyFunction: # Replace "MyFunction" with the name of your Lambda function
 Type: AWS::Lambda::Function
 Properties:
 Name: "" # Specify the name of your Lambda function
 Alias: "" # Specify the alias for your Lambda function
 CurrentVersion: "" # Specify the current version of your Lambda function
 TargetVersion: "" # Specify the version of your Lambda function to deploy
(Optional) In the Hooks section, specify a validation Lambda function to run during
a lifecycle event. Replace "LifeCycleEvent" with BeforeAllowTraffic
or AfterAllowTraffic.
Hooks:
 - LifeCycleEvent: "" # Specify a Lambda validation function between double-quotes.
```

## AppSpec Archivo JSON para una plantilla de AWS Lambda despliegue

En la siguiente plantilla, sustituya MyFunction "» por el nombre de la AWS Lambda función. En la sección Hooks opcional, sustituye los eventos del ciclo de vida por BeforeAllowTraffic o AfterAllowTraffic.

Para obtener más información acerca de los eventos del ciclo de vida que se deben utilizar en la sección Hooks, consulte [AppSpec sección de «ganchos» para una implementación de AWS Lambda](#).

```
{
 "version": 0.0,
 "Resources": [{
 "MyFunction": {
 "Type": "AWS::Lambda::Function",
 "Properties": {
 "Name": "",
 "Alias": "",
 "CurrentVersion": "",
 "TargetVersion": ""
 }
 }
]},
 "Hooks": [{
 "LifeCycleEvent": ""
```

```
}
]
}
```

## Agregue un AppSpec archivo para una implementación local de EC2/

Sin un AppSpec archivo, CodeDeploy no puede asignar los archivos de origen de la revisión de la aplicación a sus destinos ni ejecutar scripts para su implementación en una plataforma informática local de EC2/.

Cada revisión debe contener solo un archivo. AppSpec

Para añadir un AppSpec archivo a una revisión:

1. Copie la plantilla en un editor de texto.
2. Modifique la plantilla como sea necesario.
3. Usa un validador de YAML para comprobar la validez del archivo. AppSpec
4. Guarde el archivo como `appspec.yml` en el directorio raíz de la revisión.
5. Ejecute uno de los siguientes comandos para comprobar que ha colocado el AppSpec archivo en el directorio raíz:

- Para Linux, macOS o Unix:

```
find /path/to/root/directory -name appspec.yml
```

No habrá ningún resultado si el AppSpec archivo no se encuentra allí.

- Para Windows:

```
dir path\to\root\directory\appspec.yml
```

Si el archivo no está almacenado allí, aparecerá el mensaje de error AppSpec Archivo no encontrado.

6. Envíe la revisión a Amazon S3 o GitHub.

Para ver instrucciones, consulte [Enviar una revisión CodeDeploy a Amazon S3 \(solo para implementaciones de EC2/on-premise\)](#).

## AppSpec plantilla de archivo para una implementación local de EC2/con instrucciones

### Note

Las implementaciones en instancias de Windows Server no admiten el elemento `runas`. Si va a realizar la implementación en instancias de Windows Server, no la incluya en el archivo. AppSpec

```
This is an appspec.yml template file for use with an EC2/On-Premises deployment in
CodeDeploy.
The lines in this template starting with the hashtag symbol are
instructional comments and can be safely left in the file or
ignored.
For help completing this file, see the "AppSpec File Reference" in the
"CodeDeploy User Guide" at
https://docs.aws.amazon.com/codedeploy/latest/userguide/app-spec-ref.html
version: 0.0
Specify "os: linux" if this revision targets Amazon Linux,
Red Hat Enterprise Linux (RHEL), or Ubuntu Server
instances.
Specify "os: windows" if this revision targets Windows Server instances.
(You cannot specify both "os: linux" and "os: windows".)
os: linux
os: windows
During the Install deployment lifecycle event (which occurs between the
BeforeInstall and AfterInstall events), copy the specified files
in "source" starting from the root of the revision's file bundle
to "destination" on the Amazon EC2 instance.
Specify multiple "source" and "destination" pairs if you want to copy
from multiple sources or to multiple destinations.
If you are not copying any files to the Amazon EC2 instance, then remove the
"files" section altogether. A blank or incomplete "files" section
may cause associated deployments to fail.
files:
 - source:
 destination:
 - source:
 destination:
For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
you can specify a "permissions"
section here that describes special permissions to apply to the files
```

```
in the "files" section as they are being copied over to
the Amazon EC2 instance.
For more information, see the documentation.
If you are deploying to Windows Server instances,
then remove the
"permissions" section altogether. A blank or incomplete "permissions"
section may cause associated deployments to fail.
permissions:
 - object:
 pattern:
 except:
 owner:
 group:
 mode:
 acls:
 -
 context:
 user:
 type:
 range:
 type:
 -
If you are not running any commands on the Amazon EC2 instance, then remove
the "hooks" section altogether. A blank or incomplete "hooks" section
may cause associated deployments to fail.
hooks:
For each deployment lifecycle event, specify multiple "location" entries
if you want to run multiple scripts during that event.
You can specify "timeout" as the number of seconds to wait until failing the
deployment
if the specified scripts do not run within the specified time limit for the
specified event. For example, 900 seconds is 15 minutes. If not specified,
the default is 1800 seconds (30 minutes).
Note that the maximum amount of time that all scripts must finish executing
for each individual deployment lifecycle event is 3600 seconds (1 hour).
Otherwise, the deployment will stop and CodeDeploy will consider the deployment
to have failed to the Amazon EC2 instance. Make sure that the total number of
seconds
that are specified in "timeout" for all scripts in each individual deployment
lifecycle event does not exceed a combined 3600 seconds (1 hour).
For deployments to Amazon Linux, Ubuntu Server, or RHEL instances,
you can specify "runas" in an event to
run as the specified user. For more information, see the documentation.
If you are deploying to Windows Server instances,
```

```
remove "runas" altogether.
If you do not want to run any commands during a particular deployment
lifecycle event, remove that event declaration altogether. Blank or
incomplete event declarations may cause associated deployments to fail.
During the ApplicationStop deployment lifecycle event, run the commands
in the script specified in "location" starting from the root of the
revision's file bundle.
ApplicationStop:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the BeforeInstall deployment lifecycle event, run the commands
in the script specified in "location".
BeforeInstall:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the AfterInstall deployment lifecycle event, run the commands
in the script specified in "location".
AfterInstall:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the ApplicationStart deployment lifecycle event, run the commands
in the script specified in "location".
ApplicationStart:
 - location:
 timeout:
 runas:
 - location:
 timeout:
 runas:
During the ValidateService deployment lifecycle event, run the commands
in the script specified in "location".
ValidateService:
```



```

- location:
 timeout:
 runas:
- location:
 timeout:
 runas:

```

## Elige un tipo de CodeDeploy repositorio

La ubicación de almacenamiento de los archivos que necesita CodeDeploy se denomina repositorio. El uso de un repositorio depende de la plataforma de informática que se utilice en la implementación.

- **EC2/On-Premises:** para implementar el código de la aplicación en una o más instancias, el código debe estar empaquetado en un archivo comprimido y colocado en un repositorio al que CodeDeploy pueda acceder a él durante el proceso de implementación. Agrupa el contenido desplegable y un AppSpec archivo en un archivo y, a continuación, lo carga en uno de los tipos de repositorios compatibles. CodeDeploy
- **AWS Lambda y Amazon ECS:** las implementaciones requieren un AppSpec archivo al que se pueda acceder durante una implementación de una de las siguientes maneras:
  - Desde un bucket de Amazon S3.
  - Del texto escrito directamente en el AppSpec editor de la consola. Para obtener más información, consulte [Creación de una implementación en la plataforma de informática de AWS Lambda \(consola\)](#) y [Creación de una implementación en una plataforma de informática de Amazon ECS \(consola\)](#).
  - Si usa el AWS CLI, puede hacer referencia a un AppSpec archivo que esté en su disco duro o en una unidad de red. Para obtener más información, consulte [Creación de una implementación en la plataforma de informática de AWS Lambda \(CLI\)](#) y [Creación de una implementación en una plataforma de informática de Amazon ECS \(CLI\)](#).

CodeDeploy actualmente admite los siguientes tipos de repositorios:

| Tipo de repositorio | Detalles del repositorio                                                                | Plataforma de informática compatible                                        |
|---------------------|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Amazon S3           | <a href="#">Amazon Simple Storage Service</a> (Amazon S3) es la solución de AWS para el | Las implementaciones que utilizan las siguientes plataformas de informática |

almacenamiento seguro y escalable de objetos. Amazon S3 almacena los datos como objetos dentro de buckets. Un objeto consta de un archivo y, opcionalmente, cualquier metadato que describa ese archivo.

Para almacenar un objeto en Amazon S3, debe cargar el archivo en un bucket. Al cargar un archivo, puede configurar permisos y metadatos en el objeto.

Más información:

- [Creación de un bucket en Amazon S3](#)
- [Enviar una revisión CodeDeploy a Amazon S3 \(solo para implementaciones de EC2/on-premise\)](#)
- [Implemente automáticamente desde Amazon S3 mediante CodeDeploy](#)

pueden almacenar la revisión en un bucket de Amazon S3.

- EC2/en las instalaciones
- AWS Lambda
- Amazon ECS

## GitHub


Puede almacenar las revisiones de la aplicación en [GitHub](#) repositorios. Puedes activar un despliegue desde un GitHub repositorio siempre que se modifique el código fuente de ese repositorio.

Más información:

- [Integración CodeDeploy con GitHub](#)
- [Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub](#)

Solo las implementaciones locales de EC2/pueden almacenar la revisión en un repositorio. GitHub

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                       |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Bitbucket | <p><a href="#">Puedes implementar código en grupos de instancias de EC2 mediante la canalización de Bitbucket Pipelines . CodeDeploy</a> Bitbucket Pipelines ofrece funciones de integración continua e implementación continua (CI/CD), incluidas las <a href="#">implementaciones de Bitbucket</a>. La CodeDeploy tubería primero envía el artefacto a un depósito de S3 que hayas especificado y, a continuación, despliega el artefacto de código desde el depósito.</p> <p>Más información:</p> <ul style="list-style-type: none"><li>• <a href="#">Consulta la tubería de Bitbucket CodeDeploy</a></li></ul> | Solo las implementaciones de EC2/on-premise pueden almacenar la revisión en un repositorio. BitBucket |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|

 Note

Una AWS Lambda implementación solo funciona con un repositorio de Amazon S3.

## Enviar una revisión CodeDeploy a Amazon S3 (solo para implementaciones de EC2/on-premise)

Tras planificar la revisión tal y como se describe en [Planifique una revisión para CodeDeploy](#) y añadir un AppSpec archivo a la revisión tal y como se describe en [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#), estará listo para agrupar los archivos de los componentes y enviar la revisión a Amazon S3. Para las implementaciones en instancias de Amazon EC2, después de enviar la revisión, puede CodeDeploy utilizarla para implementar la revisión desde Amazon S3 en las instancias.

**Note**

CodeDeploy también se puede usar para implementar las revisiones que se han transferido. GitHub Para obtener más información, consulte GitHub la documentación.

Suponemos que ya ha seguido las instrucciones de [Empezar con CodeDeploy](#) para configurar la AWS CLI. Esto resulta de gran importancia para ejecutar el comando push que se describe más adelante.

Asegúrese de que tiene un bucket de Amazon S3. Siga las instrucciones de [Crear un bucket](#).

Si la implementación se realiza en instancias de Amazon EC2, el bucket de Amazon S3 de destino se debe crear o encontrar en la misma región que las instancias de destino. Por ejemplo, si quiere implementar una revisión en algunas instancias de la región Este de EE. UU. (Norte de Virginia) y otras instancias en la región Oeste de EE. UU. (Oregón), debe tener un bucket en la región Este de EE. UU. (Norte de Virginia) con una copia de la revisión y otro en la región Oeste de EE. UU. (Oregón) con otra copia de la misma revisión. En este caso, debería crear dos implementaciones independientes, una en la región Este de EE. UU. (Norte de Virginia) y otra en la región Oeste de EE. UU. (Oregón), aunque la revisión es la misma en ambas regiones y en ambos buckets.

Debe tener permisos para cargar en el bucket de Amazon S3. Puede especificar estos permisos mediante una política de bucket de Amazon S3. Por ejemplo, en la siguiente política de bucket de Amazon S3, el uso del carácter comodín (\*) permite 111122223333 a la AWS cuenta cargar archivos en cualquier directorio del bucket de Amazon S3 denominado `codeploydemobucket`:

```
{
 "Statement": [
 {
 "Action": [
 "s3:PutObject"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::codeploydemobucket/*",
 "Principal": {
 "AWS": [
 "111122223333"
]
 }
 }
]
}
```

```
}
```

Para ver el ID de su AWS cuenta, consulte [Cómo encontrar el ID de su AWS cuenta](#).

Para aprender a generar y vincular una política de bucket de Amazon S3, consulte [Ejemplos de políticas de bucket](#).

El usuario que llama al comando push debe disponer, como mínimo, de permisos para cargar la revisión en cada bucket de Amazon S3 de destino. Por ejemplo, la siguiente política permite que el usuario cargue revisiones en cualquier lugar del bucket de Amazon S3 denominado codedeploydemobucket:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:PutObject"
],
 "Resource": "arn:aws:s3:::codedeploydemobucket/*"
 }
]
}
```

Para obtener información sobre cómo crear y vincular una política de IAM, consulte [Trabajo con políticas](#).

## Impulsa una revisión mediante el AWS CLI

### Note

El push comando agrupa los artefactos de la aplicación y un AppSpec archivo en una revisión. El formato de archivo de esta revisión es un archivo ZIP comprimido. El comando no se puede usar con una implementación de AWS Lambda o Amazon ECS porque cada una espera una revisión que sea un archivo con formato JSON o con formato YAML. AppSpec

Ejecute el comando push para agrupar y enviar la revisión de una implementación. Sus parámetros son:

- `--application-name`: (cadena) obligatorio. El nombre de la aplicación que se va a asociar a la CodeDeploy revisión de la aplicación.
- `--s3-location`: (cadena) obligatorio. Información sobre la ubicación de la revisión de la aplicación que se cargará en Amazon S3. Debe especificar un bucket de Amazon S3 y una clave. La clave es el nombre de la revisión. CodeDeploy comprime el contenido antes de cargarlo. Utilice el formato `s3://your-S3-bucket-name/your-key.zip`.
- `--ignore-hidden-files` o `--no-ignore-hidden-files`: (booleano) opcional. Utilice la marca `--no-ignore-hidden-files` (el valor predeterminado) para empaquetar y cargar archivos ocultos en Amazon S3. Utilice la marca `--ignore-hidden-files` para no empaquetar y cargar archivos ocultos en Amazon S3.
- `--source` (cadena) opcional. La ubicación del contenido que se va a implementar y el AppSpec archivo de la máquina de desarrollo que se va a comprimir y cargar en Amazon S3. La ubicación se especifica como una ruta relativa al directorio actual. Si no se especifica la ruta relativa o si se utiliza un único punto para la ruta ("."), se utiliza el directorio actual.
- `--description` (cadena) opcional. Un comentario que resume la revisión de la aplicación. Si no se especifica, se utiliza la cadena predeterminada «Cargado por AWS CLI hora UTC», donde «hora» es la hora actual del sistema en hora universal coordinada (UTC).

Puede utilizar el AWS CLI para enviar una revisión para una implementación de Amazon EC2. Un comando de envío de ejemplo tiene este aspecto:

En Linux, macOS o Unix:

```
aws deploy push \
 --application-name WordPress_App \
 --description "This is a revision for the application WordPress_App" \
 --ignore-hidden-files \
 --s3-location s3://codedeploydemobucket/WordPressApp.zip \
 --source .
```

En Windows:

```
aws deploy push --application-name WordPress_App --description "This is a revision
for the application WordPress_App" --ignore-hidden-files --s3-location s3://
codedeploydemobucket/WordPressApp.zip --source .
```

Este comando hace lo siguiente:

- Asocia los archivos agrupados con una aplicación denominada `WordPress_App`.
- Adjunta una descripción a la revisión.
- Ignora los archivos ocultos.
- Asigna a la revisión el nombre `WordPressApp.zip` y la envía a un bucket llamado `codedeploydemobucket`.
- Agrupa todos los archivos del directorio raíz en la revisión.

Cuando la transferencia se realice correctamente, puede utilizar la consola AWS CLI o la CodeDeploy consola para implementar la revisión desde Amazon S3. Para implementar esta revisión con AWS CLI:

En Linux, macOS o Unix:

```
aws deploy create-deployment \
 --application-name WordPress_App \
 --deployment-config-name your-deployment-config-name \
 --deployment-group-name your-deployment-group-name \
 --s3-location bucket=codedeploydemobucket,key=WordPressApp.zip,bundleType=zip
```

En Windows:

```
aws deploy create-deployment --application-name WordPress_App --deployment-config-
name your-deployment-config-name --deployment-group-name your-deployment-group-name --
s3-location bucket=codedeploydemobucket,key=WordPressApp.zip,bundleType=zip
```

Para obtener más información, consulte [Cree una implementación con CodeDeploy](#).

## Vea los detalles de la revisión de la aplicación con CodeDeploy

Puede usar la CodeDeploy consola, las API o las CodeDeploy API para ver los detalles de todas las revisiones de la aplicación que están registradas en su AWS cuenta para una aplicación específica.

AWS CLI

Para obtener información sobre el registro de una revisión, consulte [Registre una revisión de aplicación en Amazon S3 con CodeDeploy](#).

Temas

- [Visualización de los detalles de una revisión de aplicación \(consola\)](#)



- [Visualización de los detalles de una revisión de aplicación \(CLI\)](#)

## Visualización de los detalles de una revisión de aplicación (consola)

Para ver los detalles de una revisión de aplicación:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y elija Aplicaciones.

### Note

Si no se muestra ninguna entrada, asegúrese de seleccionar la región correcta. En la barra de navegación, en el selector de regiones, elija una de las regiones que aparecen en [Región y puntos finales](#) en el Referencia general de AWS. CodeDeploy solo se admite en estas regiones.

3. Elija el nombre de la aplicación que tiene la revisión que desea ver.
4. En la página Application details (Detalles de la aplicación), elija la pestaña Revisions (Revisiones) y examine la lista de revisiones registradas para la aplicación. Elija una revisión y, a continuación, elija View details (Ver detalles).

## Visualización de los detalles de una revisión de aplicación (CLI)

Para usar el AWS CLI para ver la revisión de una aplicación, `get-application-revision` ejecute el comando o el `list-application-revisions` comando.

### Note

Las referencias GitHub se aplicarán únicamente a las implementaciones de EC2/on-premise. Las revisiones de las AWS Lambda implementaciones no funcionan con GitHub.

Para ver los detalles de una sola revisión de la aplicación, ejecute el [get-application-revision](#) comando y especifique:

- El nombre de la aplicación. Para obtener el nombre de la aplicación, llame al comando [list-applications](#).
- En el caso de una revisión almacenada en GitHub, el nombre del GitHub repositorio y el identificador de la confirmación que hace referencia a la revisión de la aplicación que se envió al repositorio.
- En el caso de una revisión almacenada en Amazon S3, el nombre del bucket de Amazon S3 que contiene la revisión, el nombre y tipo del archivo de almacenamiento y, opcionalmente, el identificador de versión de Amazon S3 y la ETag del archivo de almacenamiento. Si el identificador de la versión, la ETag o ambos se especificaron durante una llamada a [register-application-revision](#), deben especificarse aquí.

Para ver los detalles sobre las revisiones de varias aplicaciones, ejecute el [list-application-revisions](#) comando y especifique:

- El nombre de la aplicación. Para obtener el nombre de la aplicación, llame al comando [list-applications](#).
- Si solo desea ver los detalles de las revisiones de la aplicación en Amazon S3, el nombre del bucket de Amazon S3 que contiene las revisiones.
- Si solo desea ver los detalles de las revisiones de la aplicación en Amazon S3, una cadena de prefijo para limitar la búsqueda a las revisiones de la aplicación en Amazon S3. (Si no se especifica, CodeDeploy mostrará una lista de todas las revisiones de la aplicación Amazon S3 que coincidan).
- Opcionalmente, si se deben mostrar los detalles de las revisiones en función de si son revisiones de destino de un grupo de implementaciones. (Si no se especifica, CodeDeploy mostrará una lista de todas las revisiones coincidentes).
- Opcionalmente, el nombre de la columna y el orden en que deben mostrarse la lista de detalles de las revisiones. (Si no se especifica, CodeDeploy mostrará los resultados en un orden arbitrario).

Es posible incluir todas las revisiones, o solo las revisiones almacenadas en Amazon S3. No puede enumerar solo las revisiones almacenadas en GitHub.

# Registre una revisión de aplicación en Amazon S3 con CodeDeploy

Si ya ha ejecutado el comando [push](#) para enviar una revisión de la aplicación a Amazon S3, no es necesario registrar la revisión. Sin embargo, si sube una revisión a Amazon S3 por otros medios y desea que la revisión aparezca en la CodeDeploy consola o a través de AWS CLI, siga estos pasos para registrar primero la revisión.

Si ha subido una revisión de una aplicación a un GitHub repositorio y desea que aparezca en la CodeDeploy consola o a través de AWS CLI ella, también debe seguir estos pasos.

Puede usar solo las API AWS CLI o las CodeDeploy API para registrar las revisiones de la aplicación en Amazon S3 o GitHub.

## Temas

- [Registrar una revisión en Amazon S3 con CodeDeploy \(CLI\)](#)
- [Registrar una revisión GitHub con CodeDeploy \(CLI\)](#)

## Registrar una revisión en Amazon S3 con CodeDeploy (CLI)

1. Cargue la revisión en Amazon S3.
2. Ejecute el comando [register-application-revision](#) especificando lo siguiente:
  - El nombre de la aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
  - Información acerca de la revisión que desea registrar:
    - Nombre del bucket de Amazon S3 que contiene la revisión.
    - Nombre y tipo de archivo de la revisión cargada. En el AWS Lambda caso de las implementaciones, la revisión es un AppSpec archivo escrito en JSON o YAML. Para las implementaciones de EC2/on-premise, la revisión contiene una versión de los archivos fuente que CodeDeploy se implementarán en las instancias o los scripts que CodeDeploy se ejecutarán en las instancias.

**Note**

Los formatos de archivo tar y tar comprimido (.tar y .tar.gz) no son compatibles con las instancias de Windows Server.

- (Opcional) Identificador de versión de Amazon S3 correspondiente a la versión. (Si no se especifica el identificador de la versión, se CodeDeploy utilizará la versión más reciente).
- (Opcional) ETag de la revisión. (Si no se especifica la ETag, CodeDeploy omitirá la validación del objeto).
- (Opcional) Una descripción que desee asociar a la revisión.

La información sobre las revisiones de Amazon S3 puede especificarse en la línea de comandos utilizando la sintaxis siguiente como parte de la llamada a `register-application-revision`. (`version` y `eTag` son opcionales).

Para un archivo de revisión para una implementación de EC2/en las instalaciones:

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|zip,version=string,eTag=string
```

Para un archivo de revisión para una implementación de AWS Lambda:

```
--s3-location bucket=string,key=string,bundleType=JSON|YAML,version=string,eTag=string
```

## Registrar una revisión GitHub con CodeDeploy (CLI)

**Note**

AWS Lambda las implementaciones no funcionan con GitHub.

1. Sube la revisión a tu GitHub repositorio.
2. Ejecute el comando [register-application-revision](#) especificando lo siguiente:
  - El nombre de la aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).

- Información acerca de la revisión que desea registrar:
  - El nombre de GitHub usuario o grupo asignado al repositorio que contiene la revisión, seguido de una barra diagonal (/) seguida del nombre del repositorio.
  - El ID de la confirmación correspondiente a la revisión en el repositorio.
  - (Opcional) Una descripción que desee asociar a la revisión.

La información sobre una revisión se GitHub puede especificar en la línea de comandos, utilizando esta sintaxis como parte de la `register-application-revision` llamada:

```
--github-location repository=string,commitId=string
```

# Trabajar con despliegues en CodeDeploy

En CodeDeploy, una implementación es el proceso, y los componentes involucrados en el proceso, de instalar contenido en una o más instancias. Este contenido puede consistir en archivos de código, web y de configuración, ejecutables, paquetes, scripts, etc. CodeDeploy despliega el contenido almacenado en un repositorio de origen, de acuerdo con las reglas de configuración que especifique.

Si utiliza la plataforma de informática de EC2/en las instalaciones, pueden ejecutarse simultáneamente dos implementaciones en el mismo conjunto de instancias.

CodeDeploy ofrece dos opciones de tipos de despliegue: los despliegues locales y los despliegues azul/verde.

- **Implementación local:** la aplicación de cada instancia del grupo de implementación se para, se instala la última revisión de la aplicación, y se inicia y valida la nueva versión de la aplicación. Puede utilizar un equilibrador de carga de modo que se cancele el registro de cada instancia durante su implementación y, a continuación, vuelva a ponerse en servicio una vez completada la implementación. Solo las implementaciones que usan la plataforma de informática de EC2/en las instalaciones pueden utilizar implementaciones locales. Para obtener más información acerca de las implementaciones locales, consulte [Información general de la implementación local](#).
- **Implementación azul/verde:** el funcionamiento de la implementación depende de la plataforma de informática que utilice:
  - **Azul/verde en una plataforma de informática EC2/en las instalaciones:** las instancias de un grupo de implementación (el entorno original) se sustituyen por otro conjunto de instancias (el entorno de sustitución) siguiendo estos pasos:
    - Las instancias se aprovisionan en el entorno de sustitución.
    - La revisión de aplicación más reciente se instala en las instancias de sustitución.
    - Existe un tiempo de espera opcional para actividades como las pruebas de aplicaciones y la verificación del sistema.
    - Las instancias del entorno de sustitución se registran en uno o más equilibradores de carga de Elastic Load Balancing, lo que provoca que el tráfico se redirija a ellos. Las instancias del entorno original se anulan del registro y se pueden cerrar o mantener en ejecución para otros usos.

**Note**

Si utiliza una plataforma de informática EC2/en las instalaciones, tenga en cuenta que las implementaciones azul/verde solo funcionan con instancias de Amazon EC2.

- Azul/verde en una plataforma informática o AWS Lambda Amazon ECS: el tráfico se desplaza en incrementos según una configuración canaria, lineal o all-at-once implementación.
- Implementaciones azules/verdes AWS CloudFormation: el tráfico pasa de los recursos actuales a los recursos actualizados como parte de una actualización de la pila. AWS CloudFormation Actualmente, solo se admiten implementaciones de ECS blue/green (azul/verde).

Para obtener más información acerca de las implementaciones blue/green, consulte [Información general de la implementación azul/verde](#).

Para obtener información sobre la implementación automática desde Amazon S3, consulte [Implementación automática desde Amazon S3 mediante CodeDeploy](#).

## Temas

- [Cree una implementación con CodeDeploy](#)
- [Ver detalles CodeDeploy de la implementación](#)
- [Vea los datos de registro de las implementaciones locales CodeDeploy de EC2/](#)
- [Detenga un despliegue con CodeDeploy](#)
- [Vuelva a implementar y revertir una implementación con CodeDeploy](#)
- [Implementación de una aplicación en otra cuenta de AWS](#)
- [Utilice el CodeDeploy agente para validar un paquete de despliegue en una máquina local](#)

## Cree una implementación con CodeDeploy

Puede utilizar la CodeDeploy consola, las API o las CodeDeploy API para crear una implementación que instale las revisiones de aplicaciones que ya haya enviado a Amazon S3 o, si la implementación se realiza en una plataforma informática local de EC2 GitHub, en las instancias de un grupo de implementaciones. AWS CLI

El proceso para crear una implementación depende de la plataforma de informática usada en la implementación.

## Temas

- [Requisitos previos para una implementación](#)
- [Creación de una implementación en una plataforma de informática de Amazon ECS \(consola\)](#)
- [Creación de una implementación en la plataforma de informática de AWS Lambda \(consola\)](#)
- [Creación de una implementación en una plataforma de informática de EC2/en las instalaciones \(consola\)](#)
- [Creación de una implementación en una plataforma de informática de Amazon ECS \(CLI\)](#)
- [Creación de una implementación en la plataforma de informática de AWS Lambda \(CLI\)](#)
- [Creación de una implementación en una plataforma de informática EC2/en las instalaciones \(CLI\)](#)
- [Cree una implementación azul/verde de Amazon ECS mediante AWS CloudFormation](#)

## Requisitos previos para una implementación

Asegúrese de seguir estos pasos antes de iniciar una implementación.

### Requisitos previos para una implementación en una plataforma de informática AWS Lambda

- Cree una aplicación que incluya al menos un grupo de implementación. Para obtener más información, consulte [Cree una aplicación con CodeDeploy](#) y [Cree un grupo de despliegue con CodeDeploy](#).
- Prepare la revisión de la aplicación, también conocida como AppSpec archivo, que especifica la versión de la función Lambda que desea implementar. El AppSpec archivo también puede especificar las funciones de Lambda para validar la implementación. Para más información, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).
- Si desea utilizar una configuración de implementación personalizada para su implementación, créela antes de iniciar el proceso de implementación. Para obtener más información, consulte [Create a Deployment Configuration](#).

### Requisitos previos para una implementación en una plataforma de informática EC2/en las instalaciones

- Si se trata de una implementación "in-place", cree o configure las instancias en las que desea realizar la implementación. Para obtener más información, consulte [Trabajar con instancias para](#)



[CodeDeploy](#). Si se trata de una implementación azul/verde, dispone de un grupo de Amazon EC2 Auto Scaling para utilizarlo como plantilla para el entorno de sustitución, o de una o varias instancias o grupos de Amazon EC2 Auto Scaling que deberá especificar como entorno original. Para obtener más información, consulte [Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling](#) y [Integración CodeDeploy con Amazon EC2 Auto Scaling](#).

- Cree una aplicación que incluya al menos un grupo de implementación. Para obtener más información, consulte [Cree una aplicación con CodeDeploy](#) y [Cree un grupo de despliegue con CodeDeploy](#).
- Prepare la revisión de la aplicación que desea implementar en las instancias del grupo de implementaciones. Para obtener más información, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).
- Si desea utilizar una configuración de implementación personalizada para su implementación, créela antes de iniciar el proceso de implementación. Para obtener más información, consulte [Create a Deployment Configuration](#).
- Si está implementando la revisión de la aplicación desde un bucket de Amazon S3, el bucket se encuentra en la misma AWS región que las instancias de su grupo de implementación.
- Si está implementando la revisión de la aplicación desde un bucket de Amazon S3, debe haberse aplicado al bucket una política de bucket Amazon S3. Esta política concede a las instancias los permisos necesarios para descargar la revisión de la aplicación.

Por ejemplo, la siguiente política de bucket de Amazon S3 permite a cualquier instancia de Amazon EC2 asociada a un perfil de instancia de IAM que contenga el ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` descargar desde cualquier lugar al bucket de Amazon S3 llamado `codedeploydemobucket`:

```
{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::codedeploydemobucket/*",
 "Principal": {
 "AWS": [
 "arn:aws:iam::444455556666:role/CodeDeployDemo"
]
 }
 }
]
}
```

```

 }
 }
]
}

```

La siguiente política de bucket de Amazon S3 permite a cualquier instancia en las instalaciones con un usuario de IAM asociado que contenga el ARN `arn:aws:iam::444455556666:user/CodeDeployUser` descargar desde cualquier lugar al bucket de Amazon S3 denominado `codedeploydemobucket`:

```

{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::codedeploydemobucket/*",
 "Principal": {
 "AWS": [
 "arn:aws:iam::444455556666:user/CodeDeployUser"
]
 }
 }
]
}

```

Para obtener información acerca de cómo generar y adjuntar una política de bucket de Amazon S3, consulte [Ejemplos de política de bucket](#).

- Si está creando una implementación azul/verde, o si ha especificado un Equilibrador de carga clásico, un Equilibrador de carga de aplicación o un Equilibrador de carga de red en el grupo de implementación para una implementación local, haber creado mediante Amazon VPC una VPC que contenga al menos dos subredes. (CodeDeploy usa Elastic Load Balancing, que requiere que todas las instancias de un grupo de balanceadores de carga estén en una sola VPC).

Si aún no ha creado una VPC, consulte la [Guía de introducción a Amazon VPC](#).

- Si está creando una implementación azul/verde, haber configurado un Equilibrador de carga clásico, un Equilibrador de carga de aplicación o un Equilibrador de carga de red en Elastic Load Balancing y haberlo utilizado para registrar las instancias que componen el entorno original.

**Note**

Las instancias del entorno de sustitución se registrarán con el equilibrador de carga en otro momento.

Para obtener más información sobre la configuración de un equilibrador de carga, consulte [Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2](#) y [Configure un balanceador de carga, grupos objetivo y agentes de escucha para las implementaciones de CodeDeploy Amazon ECS](#).

## Requisitos previos de implementación para una implementación azul/verde hasta AWS CloudFormation

- La plantilla no necesita modelar los recursos para una CodeDeploy aplicación o un grupo de implementación.
- La plantilla debe incluir recursos para una VPC usando Amazon VPC que contenga al menos dos subredes.
- La plantilla debe incluir recursos para uno o más equilibradores de carga clásicos, equilibradores de carga de aplicación o equilibradores de carga de red en Elastic Load Balancing que se utilizan para dirigir el tráfico a sus grupos objetivo.

## Creación de una implementación en una plataforma de informática de Amazon ECS (consola)

En este tema se muestra cómo implementar un servicio de Amazon ECS mediante la consola. Para obtener más información, consulte [Tutorial: Implementación de una aplicación en Amazon ECS](#) y [Tutorial: Implementación de un servicio de Amazon ECS con una prueba de validación](#).

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.


 Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. Realice una de las siguientes acciones siguientes:

- Si desea implementar una aplicación, en el panel de navegación, expanda Deploy (Implementar) y, a continuación, elija Applications (Aplicaciones). Elija el nombre de la aplicación que desea implementar. Asegúrese de que la columna Plataforma de informática de la aplicación sea Amazon ECS.
- Si desea volver a implementar una implementación, en el panel de navegación, expanda Implementar y, a continuación, elija Implementaciones. Seleccione la implementación que desea volver a implementar y, en la columna Application (Aplicación), elija el nombre de su aplicación. Asegúrese de que la columna Plataforma de informática de la implementación sea Amazon ECS.

3. En la pestaña Implementaciones, elija Crear implementación.

 Note

La aplicación debe tener un grupo de implementación antes de poder implementarse. Si la aplicación no tiene un grupo de implementación, en la pestaña Grupos de implementación, elija Crear grupo de implementación. Para obtener más información, consulte [Cree un grupo de despliegue con CodeDeploy](#).

4. En Grupo de implementación, elija el grupo de implementación que va a utilizar para esta implementación.

5. Junto a Ubicación de la revisión, elija el lugar donde se encuentra la revisión:

- Mi aplicación está almacenada en Amazon S3: para obtener información, consulte [Especificar información acerca de una revisión almacenada en un bucket de Amazon S3](#) y vuelva al paso 6.
- Utilizar el AppSpec editor: selecciona JSON o YAML y, a continuación, introduce el AppSpec archivo en el editor. Para guardar el AppSpec archivo, selecciona Guardar como archivo de texto. Cuando elija Implementar al final de estos pasos, recibirá un error si el código JSON o YAML no es válido. Para obtener más información sobre la creación de un AppSpec archivo, consulte [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).

6. (Opcional) En el cuadro Descripción de la implementación, introduzca una descripción para la implementación.
7. (Opcional) En Rollback configuration overrides, puede especificar opciones de restauración automática de esta implementación distintas de las que se hayan especificado para el grupo de implementaciones, si las hay.

Para obtener información sobre las reversiones CodeDeploy, consulte [Reimplementaciones y restauración de implementaciones](#) y [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

Elija una de las siguientes opciones:

- Se revierte cuando se produce un error en una implementación CodeDeploy: se vuelve a implementar la última revisión válida conocida como una implementación nueva.
  - CodeDeploy Retirar cuando se alcanzan los umbrales de alarma: si se agregaron alarmas al grupo de implementación, vuelve a implementar la última revisión válida conocida cuando se activan una o más de las alarmas especificadas.
  - Deshabilitar restauraciones: no se realizan restauraciones de esta implementación.
8. Elija Crear implementación.

Para realizar un seguimiento del estado de la implementación, consulte [Ver detalles CodeDeploy de la implementación](#).

## Creación de una implementación en la plataforma de informática de AWS Lambda (consola)

En este tema se muestra cómo implementar una función de Lambda mediante la consola.


1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. Realice una de las siguientes acciones siguientes:

- Si desea implementar una aplicación, en el panel de navegación, expanda Deploy (Implementar) y, a continuación, elija Applications (Aplicaciones). Elija el nombre de la aplicación que desea implementar. Asegúrese de que la columna Plataforma de informática de la aplicación sea AWS Lambda.
  - Si desea volver a implementar una implementación, en el panel de navegación, expanda Implementar y, a continuación, elija Implementaciones. Seleccione la implementación que desea volver a implementar y, en la columna Application (Aplicación), elija el nombre de su aplicación. Asegúrese de que la columna Plataforma de informática de la implementación sea AWS Lambda.
3. En la pestaña Implementaciones, elija Crear implementación.

 Note

La aplicación debe tener un grupo de implementación antes de poder implementarse. Si la aplicación no tiene un grupo de implementación, en la pestaña Grupos de implementación, elija Crear grupo de implementación. Para obtener más información, consulte [Cree un grupo de despliegue con CodeDeploy](#).

4. En Grupo de implementación, elija el grupo de implementación que va a utilizar para esta implementación.
5. Junto a Ubicación de la revisión, elija el lugar donde se encuentra la revisión:
  - Mi aplicación está almacenada en Amazon S3: para obtener información, consulte [Especificar información acerca de una revisión almacenada en un bucket de Amazon S3](#) y vuelva al paso 6.
  - Utilizar el AppSpec editor: selecciona JSON o YAML y, a continuación, introduce el AppSpec archivo en el editor. Para guardar el AppSpec archivo, selecciona Guardar como archivo de texto. Cuando elija Implementar al final de estos pasos, recibirá un error si el código JSON o YAML no es válido. Para obtener más información sobre la creación de un AppSpec archivo, consulte [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#).
6. (Opcional) En el cuadro Descripción de la implementación, introduzca una descripción para la implementación.
7. (Opcional) Expanda Anulaciones del grupo de implementación para elegir una configuración de implementación que controle cómo se desvía el tráfico a una versión de la función de Lambda distinta de la que se especifica en el grupo de implementación.

Para obtener más información, consulte [Configuraciones de implementación en una plataforma de informática de AWS Lambda](#).

8. (Opcional) En Rollback configuration overrides, puede especificar opciones de restauración automática de esta implementación distintas de las que se hayan especificado para el grupo de implementaciones, si las hay.

Para obtener información sobre las reversiones CodeDeploy, consulte [Reimplementaciones y restauración de implementaciones](#) y [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

Elija una de las siguientes opciones:

- Se revierte cuando se produce un error en una implementación CodeDeploy: se vuelve a implementar la última revisión válida conocida como una implementación nueva.
  - CodeDeploy Retirar cuando se alcanzan los umbrales de alarma: si se agregaron alarmas al grupo de implementación, vuelve a implementar la última revisión válida conocida cuando se activan una o más de las alarmas especificadas.
  - Deshabilitar restauraciones: no se realizan restauraciones de esta implementación.
9. Elija Crear implementación.

Para realizar un seguimiento del estado de la implementación, consulte [Ver detalles CodeDeploy de la implementación](#).

## Creación de una implementación en una plataforma de informática de EC2/ en las instalaciones (consola)


En este tema se muestra cómo implementar una aplicación un servidor Amazon EC2 o en las instalaciones mediante la consola.

1. [Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en https://console.aws.amazon.com/codedeploy](https://console.aws.amazon.com/codedeploy).

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. Realice una de las siguientes acciones siguientes:
  - Si desea implementar una aplicación, en el panel de navegación, expanda Deploy (Implementar) y, a continuación, elija Applications (Aplicaciones). Elija el nombre de la aplicación que desea implementar. Asegúrese de que la columna Plataforma de informática de la aplicación sea EC2/en las instalaciones.
  - Si desea volver a implementar una implementación, en el panel de navegación, expanda Implementar y, a continuación, elija Implementaciones. Localice la implementación que desea volver a implementar y, a continuación, elija el nombre de su aplicación en la columna Application (Aplicación). Asegúrese de que la columna Plataforma de informática de la implementación sea EC2/en las instalaciones.
3. En la pestaña Implementaciones, elija Crear implementación.

 Note

La aplicación debe tener un grupo de implementación antes de poder implementarse. Si la aplicación no tiene un grupo de implementación, en la pestaña Grupos de implementación, elija Crear grupo de implementación. Para obtener más información, consulte [Cree un grupo de despliegue con CodeDeploy](#).

4. En Grupo de implementación, elija el grupo de implementación que va a utilizar para esta implementación.
5. Junto a Repository type (Tipo de repositorio), seleccione el tipo de repositorio en el que está almacenada la revisión:
  - Mi aplicación está almacenada en Amazon S3: para obtener información, consulte [Especificar información acerca de una revisión almacenada en un bucket de Amazon S3](#) y vuelva al paso 6.
  - Mi aplicación está guardada en GitHub: para obtener más información, consulte y [Especifique la información sobre una revisión almacenada en un GitHub repositorio](#), a continuación, vuelva al paso 6.
6. (Opcional) En el cuadro Descripción de la implementación, introduzca una descripción para la implementación.
7. (Opcional) Expanda Anular la configuración de la implementación para elegir una configuración de implementación que controle cómo se desvía el tráfico al servidor de Amazon EC2, o a uno en las instalaciones, distinto del que se especifica en el grupo de implementación.



Para obtener más información, consulte [Trabajar con configuraciones de implementación en CodeDeploy](#).

8. a. Seleccione No fallar la implementación si el evento del ApplicationStop ciclo de vida falla si desea que la implementación en una instancia se realice correctamente si el evento del ApplicationStop ciclo de vida falla.
- b. Amplíe la configuración adicional del comportamiento de despliegue para especificar CodeDeploy cómo se gestionan los archivos en una ubicación de destino de despliegue que no formaban parte de la anterior implementación exitosa.

Elija una de las siguientes opciones:

- Dar como errónea la implementación: se notifica un error y el estado de la implementación cambia a Failed.
- Sobrescribir el contenido: si hay un archivo del mismo nombre en la ubicación de destino, la versión de la revisión de la aplicación lo sustituye.
- Retener el contenido: si hay un archivo del mismo nombre en la ubicación de destino, este se conserva y la versión de la revisión de la aplicación no se copia en la instancia.

Para obtener más información, consulte [Comportamiento de la restauración con contenido existente](#).

9. (Opcional) En Rollback configuration overrides, puede especificar opciones de restauración automática de esta implementación distintas de las que se hayan especificado para el grupo de implementaciones, si las hay.

Para obtener información sobre las reversiones CodeDeploy, consulte [Reimplementaciones y restauración de implementaciones](#) y [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

Elija una de las siguientes opciones:

- Se revierte cuando se produce un error en una implementación CodeDeploy: se vuelve a implementar la última revisión válida conocida como una implementación nueva.
- Retirar cuando se alcanzan los umbrales de alarma: si se agregaron alarmas al grupo de implementación, CodeDeploy implementa la última revisión válida conocida cuando se activan una o más de las alarmas especificadas.
- Deshabilitar restauraciones: no se realizan restauraciones de esta implementación.

## 10. Elija Iniciar la implementación.

Para realizar un seguimiento del estado de la implementación, consulte [Ver detalles CodeDeploy de la implementación](#).

### Temas

- [Especificar información acerca de una revisión almacenada en un bucket de Amazon S3](#)
- [Especifique la información sobre una revisión almacenada en un GitHub repositorio](#)

## Especificar información acerca de una revisión almacenada en un bucket de Amazon S3

Si está siguiendo los pasos descritos en [Creación de una implementación en una plataforma de informática de EC2/en las instalaciones \(consola\)](#), siga estos también para añadir detalles acerca de una revisión de la aplicación almacenada en un bucket de Amazon S3.

1. Copie el enlace de Amazon S3 de la revisión en el cuadro Ubicación de la revisión. Para encontrar el valor del enlace:

- a. En una pestaña diferente del navegador:

Inicie sesión en la consola de Amazon S3 AWS Management Console y ábrala en <https://console.aws.amazon.com/s3/>.

Busque y seleccione la revisión.

- b. Si el panel Properties no está visible, elija el botón Properties.
- c. En el panel de propiedades, copie el valor del campo Enlace en el cuadro Ubicación de la revisión de la CodeDeploy consola.

Para especificar una ETag (una suma de comprobación de archivo) como parte de la ubicación de la revisión:

- Si el valor del campo Link (Enlace) termina en **?versionId=*versionId***, añada **&etag=** y la ETag al final del valor del campo Link (Enlace).
- Si el valor del campo Link (Enlace) no especifica un ID de versión, añada **?etag=** y la ETag al final del valor del campo Link (Enlace).

**Note**

Aunque no es tan fácil como copiar el valor del campo Link, también puede escribir la ubicación de la revisión en uno de los siguientes formatos:

**s3://*bucket-name*/*folders*/*objectName***

**s3://*bucket-name*/*folders*/*objectName*?versionId=*versionId***

**s3://*bucket-name*/*folders*/*objectName*?etag=*etag***

**s3://*bucket-name*/*folders*/*objectName*?versionId=*versionId*&etag=*etag***

***bucket-name*.s3.amazonaws.com/*folders*/*objectName***


2. Si aparece un mensaje en la lista File type que indique que no se ha podido detectar el tipo de archivo, seleccione el tipo de archivo de la revisión. En caso contrario, acepte el tipo de archivo detectado.

Especifique la información sobre una revisión almacenada en un GitHub repositorio

Si sigue los pasos descritos [Creación de una implementación en una plataforma de informática de EC2/en las instalaciones \(consola\)](#), siga estos pasos para añadir detalles sobre una revisión de la aplicación almacenada en un GitHub repositorio.

1. En Conectarse a GitHub, realice una de las siguientes acciones:
  - Para crear una conexión de CodeDeploy aplicaciones a una GitHub cuenta, cierre sesión en otra pestaña del navegador web GitHub. En la GitHub cuenta, introduce un nombre para identificar esta conexión y, a continuación, selecciona Conectar a GitHub. La página web le pide que autorice CodeDeploy la interacción con GitHub su aplicación. Continúe en el paso 2.
  - Para usar una conexión que ya haya creado, en la GitHub cuenta, seleccione su nombre y, a continuación, elija Conectar a GitHub. Continúe en el paso 4.
  - Para crear una conexión a otra GitHub cuenta, cierra sesión en otra pestaña del navegador web GitHub. Selecciona Conectarse a una GitHub cuenta diferente y, a continuación, selecciona Conectar a GitHub. Continúe en el paso 2.
2. Si se te pide que inicies sesión GitHub, sigue las instrucciones de la página de inicio de sesión. Inicie sesión con su nombre GitHub de usuario o correo electrónico y contraseña.
3. Si aparece la página Authorize application, elija Authorize application.

4. En la página Crear despliegue, en el cuadro Nombre del repositorio, introduzca el nombre del GitHub usuario o la organización que contiene la revisión, seguido de una barra diagonal (/) seguida del nombre del repositorio que contiene la revisión. Si no está seguro del valor que debe especificar:
  - a. En otra pestaña del navegador web, ve a tu [GitHubpanel](#) de control.
  - b. En Your repositories, pase el cursor sobre el nombre del repositorio de destino. Aparece una descripción emergente en la que se muestra el nombre del GitHub usuario o la organización, seguido de una barra diagonal (/) seguida del nombre del repositorio. Escriba el valor mostrado en el cuadro Repository name (Nombre de repositorio).

 Note

Si el nombre del repositorio de destino no está visible en Tus repositorios, usa el GitHub cuadro de búsqueda para buscar el nombre del repositorio de destino y el nombre del GitHub usuario u organización.

5. En el campo Commit ID (ID de confirmación), escriba el ID de la confirmación que hace referencia a la revisión en el repositorio. Si no está seguro del valor que debe especificar:
  - a. En otra pestaña del navegador web, ve a tu [GitHubpanel](#) de control.
  - b. En Your repositories, seleccione el nombre del repositorio que contiene la confirmación de destino.
  - c. En la lista de confirmaciones, busque y copie el ID de confirmación que hace referencia a la revisión en el repositorio. Este ID suele tener 40 caracteres y se compone de letras y números. No utilice la versión corta del ID de confirmación, que suele ser los primeros 10 caracteres de la versión larga.
  - d. Pegue el ID de confirmación en el cuadro Commit ID.

## Creación de una implementación en una plataforma de informática de Amazon ECS (CLI)

Una vez creada la aplicación y la revisión (en las implementaciones de Amazon ECS, este es el AppSpec archivo):

Llame al comando [create-deployment](#) especificando lo siguiente:

- Un nombre de aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
- El nombre del grupo de implementaciones. Para ver una lista de los nombres de los grupos de implementación, ejecute el [list-deployment-groups](#) comando.
- Información acerca de la revisión a implementar:

Para revisiones almacenadas en Amazon S3:

- Nombre del bucket de Amazon S3 que contiene la revisión.
- El nombre de la revisión cargada.
- (Opcional) El identificador de versión de Amazon S3 de la revisión. (Si no se especifica el identificador de versión, CodeDeploy utiliza la versión más reciente).
- (Opcional) La ETag de la revisión. (Si no se especifica la ETag, CodeDeploy omite la validación del objeto).

Para las revisiones almacenadas en un archivo que no está en Amazon S3, necesita el nombre de archivo y su ruta. El archivo de revisión está escrito en JSON o YAML, por lo que probablemente tenga una extensión `.json` o `.yaml`.

- Una descripción para la implementación (opcional).

El archivo de revisión se puede especificar como un archivo cargado en un bucket de Amazon S3 o como una cadena. La sintaxis de cada uno cuando se usa como parte del comando `create-deployment` es:

- Bucket de Amazon S3:

Los parámetros `version` y `eTag` son opcionales.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Cadena:

```
--revision '{"revisionType": "String", "string": {"content": "revision-as-string"}}'
```

**Note**

El comando `create-deployment` puede cargar una revisión desde un archivo. Para obtener más información, consulte [Carga de parámetros desde un archivo](#).

Para ver las plantillas AWS Lambda de revisión de despliegue, consulte [Agregar un AppSpec archivo para una implementación de AWS Lambda](#). Para ver una revisión de ejemplo, consulte [AppSpec Ejemplo de archivo para una implementación de AWS Lambda](#).

Para realizar un seguimiento del estado de la implementación, consulte [Ver detalles CodeDeploy de la implementación](#).

## Creación de una implementación en la plataforma de informática de AWS Lambda (CLI)

Una vez que haya creado la aplicación y la revisión (en las implementaciones de AWS Lambda, este es el AppSpec archivo):

Llame al comando [create-deployment](#) especificando lo siguiente:

- Un nombre de aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
- El nombre del grupo de implementaciones. Para ver una lista de los nombres de los grupos de implementación, ejecute el [list-deployment-groups](#) comando.
- Información acerca de la revisión a implementar:

Para revisiones almacenadas en Amazon S3:

- Nombre del bucket de Amazon S3 que contiene la revisión.
- El nombre de la revisión cargada.
- (Opcional) El identificador de versión de Amazon S3 de la revisión. (Si no se especifica el identificador de versión, CodeDeploy utiliza la versión más reciente).
- (Opcional) La ETag de la revisión. (Si no se especifica la ETag, CodeDeploy omite la validación del objeto).

Para las revisiones almacenadas en un archivo que no está en Amazon S3, necesita el nombre de archivo y su ruta. El archivo de revisión está escrito en JSON o YAML, por lo que probablemente tenga una extensión `.json` o `.yaml`.

- El nombre de la configuración de implementación a utilizar (opcional). Para ver una lista de las configuraciones de despliegue, ejecute el [list-deployment-configs](#) comando. (Si no se especifica, CodeDeploy utiliza una configuración de despliegue predeterminada específica).
- Una descripción para la implementación (opcional).

El archivo de revisión se puede especificar como un archivo cargado en un bucket de Amazon S3 o como una cadena. La sintaxis de cada uno cuando se usa como parte del comando `create-deployment` es:

- Bucket de Amazon S3:

Los parámetros `version` y `eTag` son opcionales.

```
--s3-location bucket=string,key=string,bundleType=JSON|
YAML,version=string,eTag=string
```

- Cadena:

```
--revision '{"revisionType": "String", "string": {"content": "revision-as-string"}}'
```

#### Note

El comando `create-deployment` puede cargar una revisión desde un archivo. Para obtener más información, consulte [Carga de parámetros desde un archivo](#).

Para ver las plantillas de revisión de AWS Lambda despliegue, consulte [Agregar un AppSpec archivo para una implementación de AWS Lambda](#). Para ver una revisión de ejemplo, consulte [AppSpec Ejemplo de archivo para una implementación de AWS Lambda](#).

Para realizar un seguimiento del estado de la implementación, consulte [Ver detalles CodeDeploy de la implementación](#).

## Creación de una implementación en una plataforma de informática EC2/en las instalaciones (CLI)

Para usar el AWS CLI para implementar una revisión en la plataforma informática local de EC2:

1. Una vez preparadas las instancias, creada la aplicación y enviada la revisión, realice alguna de las siguientes acciones:
  - Si desea implementar una revisión de un bucket de Amazon S3, continúe con el paso 2.
  - Si desea implementar una revisión desde un GitHub repositorio, primero complete los pasos del paso 2 y, a continuación [Conectar una CodeDeploy aplicación a un GitHub repositorio](#), continúe con el paso 2.
2. Llame al comando [create-deployment](#) especificando lo siguiente:
  - `--application-name`: un nombre de aplicación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
  - `--deployment-group-name`: un nombre de grupo de implementación de Amazon EC2. Para ver una lista de los nombres de los grupos de implementación, [list-deployment-groups](#) ejecute el comando.
  - `--revision`: información acerca de la revisión a implementar:

Para revisiones almacenadas en Amazon S3:

- `s3Location`: nombre del bucket de Amazon S3 que contiene la revisión.
- `s3Location --> key`: nombre de la revisión cargada.
- `s3Location --> bundleType`: nombre y tipo de archivo de la revisión cargada.

### Note

Los formatos de archivo tar y tar comprimido (.tar y .tar.gz) no son compatibles con las instancias de Windows Server.

- `s3Location --> version`: (Opcional) El identificador de versión de Amazon S3 de la revisión. (Si no se especifica el identificador de versión, CodeDeploy utiliza la versión más reciente).
- `s3Location --> eTag`: la ETag de la revisión (opcional). (Si no se especifica la ETag, CodeDeploy omite la validación del objeto).

Para las revisiones almacenadas en: GitHub



- `githubLocation --> repository`: el nombre de GitHub usuario o grupo asignado al repositorio que contiene la revisión, seguido de una barra diagonal (/) seguida del nombre del repositorio.
- `githubLocation --> commitId`: ID de confirmación de la revisión.
- `--deployment-config-name`: (Opcional) El nombre de la configuración de implementación a utilizar. Para ver una lista de las configuraciones de despliegue, ejecute el [list-deployment-configs](#) comando. (Si no se especifica, CodeDeploy utiliza una configuración de despliegue predeterminada específica).
- `--ignore-application-stop-failures` | `--no-ignore-application-stop-failures`: si desea que la implementación en una instancia continúe y pase al evento del ciclo de vida `BeforeInstall` en caso de que se produzca un error en el evento del ciclo de vida de la implementación `ApplicationStop` (opcional).
- `--description`: una descripción para la implementación (opcional).
- `--file-exists-behavior`: (Opcional) Como parte del proceso de implementación, el CodeDeploy agente elimina de cada instancia todos los archivos instalados en la implementación más reciente. Elija qué ocurre cuando los archivos que no formaban parte de una implementación anterior aparecen en las ubicaciones de implementación de destino.
- `--target-instances`: en el caso de las implementaciones azul/verde, información sobre las instancias que pertenecen al entorno de sustitución en una implementación azul/verde, incluidos los nombres de uno o varios grupos de Amazon EC2 Auto Scaling, o la clave, el tipo y el valor del filtro de etiquetas utilizado para identificar las instancias de Amazon EC2.

#### Note

Utilice esta sintaxis como parte de la llamada a `create-deployment` para especificar información sobre una revisión en Amazon S3 directamente en la línea de comandos. Los parámetros `version` y `eTag` son opcionales.

```
--s3-location bucket=string,key=string,bundleType=tar|tgz|zip,version=string,eTag=string
```

Utilice esta sintaxis como parte de la `create-deployment` llamada para especificar la información sobre una revisión GitHub directamente en la línea de comandos:

```
--github-location repository=string,commitId=string
```

Para obtener información sobre las revisiones que ya se han publicado, ejecute el [list-application-revisions](#) comando.

Para realizar un seguimiento del estado de la implementación, consulte [Ver detalles CodeDeploy de la implementación](#).

## referencia del comando create-deployment

A continuación se muestran la estructura y las opciones del comando `create-deployment`. Para obtener más información, consulte la referencia de [create-deployment](#) en la Referencia del comando AWS CLI.

```
create-deployment
--application-name <value>
[--deployment-group-name <value>]
[--revision <value>]
[--deployment-config-name <value>]
[--description <value>]
[--ignore-application-stop-failures | --no-ignore-application-stop-failures]
[--target-instances <value>]
[--auto-rollback-configuration <value>]
[--update-outdated-instances-only | --no-update-outdated-instances-only]
[--file-exists-behavior <value>]
[--s3-location <value>]
[--github-location <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

## Conectar una CodeDeploy aplicación a un GitHub repositorio


Antes de poder implementar una aplicación desde un GitHub repositorio por primera vez utilizando el AWS CLI, primero debes dar CodeDeploy permiso para interactuar con ella GitHub en nombre de tu GitHub cuenta. Este paso debe completarse una vez para cada aplicación que utilice la CodeDeploy consola.

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

 Note


Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. Elija Aplicaciones.
3. En Aplicaciones, elija la aplicación que desee vincular a su cuenta de GitHub usuario y elija Implementar aplicación.

 Note

No está creando una implementación. Actualmente, esta es la única forma de conceder CodeDeploy permiso para interactuar con ella GitHub en nombre de su cuenta de GitHub usuario.

4. Junto a Tipo de repositorio, selecciona En qué se guarda la revisión de mi aplicación GitHub.
5. Selecciona Conectar a GitHub.

 Note

Si ves un enlace de Conectarse a una GitHub cuenta diferente:  
Es posible que ya tengas autorización CodeDeploy para interactuar GitHub en nombre de otra GitHub cuenta de la aplicación.  
Es posible que hayas revocado la autorización CodeDeploy para interactuar GitHub en nombre de la GitHub cuenta en la que has iniciado sesión en todas las aplicaciones vinculadas a ella. CodeDeploy  
Para obtener más información, consulte [GitHub autenticación con aplicaciones en CodeDeploy](#).

6. Si aún no ha iniciado sesión GitHub, siga las instrucciones de la página de inicio de sesión.
7. En la página Authorize application, elija Authorize application.
8. Ahora CodeDeploy que tienes permiso, selecciona Cancelar y continúa con los pasos indicados [Creación de una implementación en una plataforma de informática EC2/en las instalaciones \(CLI\)](#).

# Cree una implementación azul/verde de Amazon ECS mediante AWS CloudFormation

Puede utilizarlo AWS CloudFormation para gestionar las implementaciones azul/verde de Amazon ECS mediante CodeDeploy. Para generar la implementación, defina los recursos verdes y azules y especifique la configuración de enrutamiento y estabilización del tráfico que se va a utilizar en AWS CloudFormation. En este tema se describen las diferencias entre las implementaciones azul/verde de Amazon ECS gestionadas por CodeDeploy y las gestionadas por AWS CloudFormation.

Para step-by-step obtener instrucciones sobre cómo administrar las implementaciones azul/verde de Amazon ECS, consulte [Automatizar las implementaciones azul/verde de ECS mediante CodeDeploy el uso AWS CloudFormation en la Guía del usuario](#). AWS CloudFormation

## Note

La administración de las implementaciones azul/verde de Amazon ECS con no AWS CloudFormation está disponible en la región de Asia Pacífico (Osaka).

## Diferencias entre las implementaciones azul/verde de Amazon ECS hasta y CodeDeploy AWS CloudFormation

La plantilla de AWS CloudFormation pila modela los recursos y la infraestructura relacionados con las tareas de Amazon ECS, así como las opciones de configuración para las implementaciones. Por lo tanto, existen diferencias entre las implementaciones azul/verde estándar de Amazon ECS y las implementaciones azul/verde que se crean mediante AWS CloudFormation.

A diferencia de las implementaciones azul/verde estándar de Amazon ECS, usted no modela ni crea manualmente lo siguiente:

- No se crea una AWS CodeDeploy aplicación especificando un nombre que represente de forma exclusiva lo que se quiere implementar.
- No se crea un grupo AWS CodeDeploy de implementación.
- No se especifica un archivo de especificaciones de la aplicación (AppSpec archivo). La información que normalmente se gestiona con el AppSpec archivo, como las opciones de configuración ponderadas o los eventos del ciclo de vida, se gestiona mediante un `AWS::CodeDeploy::BlueGreen` enlace.

En esta tabla se resumen las diferencias en el flujo de trabajo de alto nivel entre los tipos de implementación.

| Función                                                                                                                                                                                                           | Implementaciones blue/green (azul/verde) estándar                        | Implementaciones azul/verde mediante AWS CloudFormation                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Especifique el clúster de Amazon ECS, el servicio Amazon ECS, el Equilibrador de carga de aplicación o el Equilibrador de carga de red, el Oyente de producción, el oyente de prueba y los dos grupos de destino. | Cree un grupo de CodeDeploy y despliegue que especifique estos recursos. | Cree una AWS CloudFormation plantilla para modelar estos recursos.                                                                                  |
| Especifique el cambio que se va a implementar.                                                                                                                                                                    | Cree una CodeDeploy aplicación.                                          | Cree una AWS CloudFormation plantilla que especifique la imagen del contenedor.                                                                     |
| Especifique la definición de tarea de Amazon ECS, el nombre del contenedor y el puerto del contenedor.                                                                                                            | Cree un AppSpec archivo que especifique estos recursos.                  | Cree una AWS CloudFormation plantilla para modelar estos recursos.                                                                                  |
| Especifique las opciones de cambio de tráfico de implementación y los enlaces de eventos de ciclo de vida.                                                                                                        | Cree un AppSpec archivo que especifique estas opciones.                  | Cree una AWS CloudFormation plantilla que utilice los parámetros de <code>AWS::CodeDeploy::BlueGreen</code> gancho para especificar estas opciones. |
| CloudWatch alarmas.                                                                                                                                                                                               | Cree una CloudWatch alarma que desencadene una reversión.                | Configure una CloudWatch alarma a nivel de AWS CloudFormation pila que active una reversión.                                                        |

| Función                        | Implementaciones blue/green (azul/verde) estándar            | Implementaciones azul/verde mediante AWS CloudFormation    |
|--------------------------------|--------------------------------------------------------------|------------------------------------------------------------|
| Restauración/reimplementación. | Especifique las opciones de restauración y reimplementación. | Cancela la actualización de la pila en AWS CloudFormation. |

## Supervisión de las implementaciones azul/verde de Amazon ECS mediante AWS CloudFormation

Puede monitorear las implementaciones azul/verde mediante y. AWS CloudFormation CodeDeploy Para obtener información sobre la supervisión integral AWS CloudFormation, consulte [Supervisión de eventos azules/verdes en AWS CloudFormation](#) la Guía del usuario.AWS CloudFormation

Para ver el estado de despliegue de los despliegues azul/verde en CodeDeploy

1. [Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en https://console.aws.amazon.com/codedeploy.](https://console.aws.amazon.com/codedeploy)

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).


2. En Implementaciones, aparece la implementación que se activó por la actualización de la AWS CloudFormation pila. Elija la implementación para ver el Deployment history (Historial de implementaciones).


| Deployment Id | Status    | Deployment type | Compute platform | Application | Deployment group | Revision location | Initiating event            | Start      |
|---------------|-----------|-----------------|------------------|-------------|------------------|-------------------|-----------------------------|------------|
| d-H8IKM2571   | Succeeded | Blue/green      | Amazon ECS       | -           | -                | stack/dkst...     | CloudFormation stack update | Nov 1 3:41 |


3. Elija la implementación para ver el estado de cambio de tráfico. Tenga en cuenta que el grupo de aplicación e implantación no se han creado.

**d-H8IKMZ571**

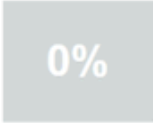

### Deployment status

Step 1:  
Deploying replacement task set Completed  
 ✔ Succeeded

Step 2:  
Rerouting production traffic to replacement task set 100% traffic shifted  
 ✔ Succeeded

Step 3:  
Terminate original task set Completed  
 ✔ Succeeded

### Traffic shifting progress

|                                                                                     |                                                                                     |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Original                                                                            | Replacement                                                                         |
|  |  |
| Original task set not serving traffic                                               | Replacement task set                                                                |

### Deployment details

|                                                                                                           |                  |                             |
|-----------------------------------------------------------------------------------------------------------|------------------|-----------------------------|
| Application                                                                                               | Deployment ID    | Status                      |
| -                                                                                                         | d-H8IKMZ571      | ✔ Succeeded                 |
| Deployment configuration                                                                                  | Deployment group | Initiated by                |
| -                                                                                                         | -                | CloudFormation stack update |
| Deployment description                                                                                    |                  |                             |
| This deployment is triggered by a stack update for CloudFormation stackId arn:aws:cloudformation:eu-west- |                  |                             |

4. Lo siguiente se aplica para restaurar o detener la implementación:

- La implementación correcta aparece en CodeDeploy y muestra que la implementación fue iniciada por AWS CloudFormation.
- Si quiere detener y revertir la implementación, debe cancelar la actualización de la pila en AWS CloudFormation.

## Ver detalles CodeDeploy de la implementación

Puede usar la CodeDeploy consola, las API o las CodeDeploy API para ver los detalles de las implementaciones asociadas a su AWS cuenta. AWS CLI

### Note

Puede ver los registros de las implementaciones de EC2/en las instalaciones de las instancias en las siguientes ubicaciones:

- Para Amazon Linux, RHEL y Ubuntu Server: `/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log`
- Servidor Windows: `C:\ProgramData\Amazon\CodeDeploy <DEPLOYMENT-GROUP-ID>\logs\scripts.log`

Para obtener más información, consulte [Análisis de archivos de registro para investigar los errores de implementación en instancias](#).

## Temas

- [Visualización de los detalles de la implementación \(consola\)](#)
- [Visualización de los detalles de la implementación \(CLI\)](#)

## Visualización de los detalles de la implementación (consola)

Para usar la CodeDeploy consola para ver los detalles de la implementación:

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y después elija Aplicaciones.

### Note

Si no se muestra ninguna entrada, asegúrese de seleccionar la región correcta. En la barra de navegación, en el selector de regiones, elija una de las regiones que aparecen en [Región y puntos finales](#) en. Referencia general de AWS CodeDeploy solo se admite en estas regiones.

3. Para ver más detalles sobre una única implementación, en Deployment history (Historial de implementaciones), elija el ID de la implementación o seleccione el botón situado junto al ID de implementación y, a continuación, elija View (Ver).



## Visualización de los detalles de la implementación (CLI)

Para usar el AWS CLI para ver los detalles de la implementación, llame al `get-deployment` comando o al `batch-get-deployments` comando. Puede llamar al comando `list-deployments` para obtener una lista de ID exclusivos de las implementaciones para utilizar como entradas del comando `get-deployment` y del comando `batch-get-deployments`.

Para ver los detalles de una única implementación, ejecute el comando [get-deployment](#), especificando el identificador exclusivo de la implementación. Para obtener el ID de implementación, llame al comando [list-deployments](#).

Para ver los detalles de varios despliegues, llama al [batch-get-deployments](#) comando y especifica varios identificadores de despliegue únicos. Para obtener los ID de implementación, llame al comando [list-deployments](#).

Para ver una lista de los ID de implementación, llame al comando [list-deployments](#) especificando lo siguiente:

- El nombre de la aplicación asociada a la implementación. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
- El nombre del grupo de implementaciones asociado a la implementación. Para ver una lista de los nombres de los grupos de despliegue, ejecute el [list-deployment-groups](#) comando.
- Si lo prefiere, puede incluir detalles acerca de las implementaciones por su estado de implementación. (Si no se especifican, se enumerarán todas las implementaciones coincidentes, independientemente de su estado de implementación).
- Si lo prefiere, puede incluir detalles acerca de las implementaciones por la hora de inicio o final de creación o ambas. (Si no se especifican, se enumerarán todas las implementaciones coincidentes, independientemente de sus horas de creación).

## Vea los datos de registro de las implementaciones locales CodeDeploy de EC2/

Puedes ver los datos de registro creados por una CodeDeploy implementación configurando el CloudWatch agente de Amazon para que vea los datos agregados en la CloudWatch consola o iniciando sesión en una instancia individual para revisar el archivo de registro.

**Note**

Los registros no son compatibles con las AWS implementaciones de Lambda o Amazon ECS. Solamente se pueden crear para implementaciones de EC2/en las instalaciones.

**Temas**

- [Ver los datos de los archivos de registro en la CloudWatch consola de Amazon](#)
- [Visualización de los archivos de registro de una instancia](#)

## Ver los datos de los archivos de registro en la CloudWatch consola de Amazon

Cuando el CloudWatch agente de Amazon está instalado en una instancia, los datos de despliegue de todas las implementaciones de esa instancia están disponibles para su visualización en la CloudWatch consola. Para simplificar, le recomendamos que lo utilice CloudWatch para supervisar de forma centralizada los archivos de registro en lugar de verlos instancia por instancia. Para obtener más información, consulte [Enviar los registros CodeDeploy de los agentes a CloudWatch](#).

## Visualización de los archivos de registro de una instancia

Para ver los datos del log de implementación de una instancia individual, puede iniciar sesión en la instancia y examinar la información sobre errores u otros eventos de implementación.

**Temas**

- [Para ver los archivos de registro de implementación en las instancias de Amazon Linux, RHEL y Ubuntu Server](#)
- [Para ver los archivos de registro de implementación en las instancias de Windows Server](#)

## Para ver los archivos de registro de implementación en las instancias de Amazon Linux, RHEL y Ubuntu Server

En las instancias de Amazon Linux, RHEL y Ubuntu Server, los registros de implementación se encuentran en la siguiente ubicación:

```
/opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log
```

Para ver o analizar los registros de implementación en las instancias de Amazon Linux, RHEL y Ubuntu Server, inicie sesión en la instancia y, a continuación, escriba el siguiente comando para abrir el archivo de registro del CodeDeploy agente:

```
less /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

Ejecute los comandos siguientes para buscar mensajes de error en el archivo de registro:

| Comando            | Resultado                                                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>&amp; ERROR</b> | Muestra solo los mensajes de error del archivo de registro. Utilice un único espacio delante y detrás de la palabra <b>ERROR</b> . |
| <b>/ ERROR</b>     | Busque el mensaje de error siguiente. <sup>1</sup>                                                                                 |
| <b>? ERROR</b>     | Busque el mensaje de error anterior. <sup>2</sup> Utilice un solo espacio antes y después de la palabra <b>ERROR</b> .             |
| <b>G</b>           | Ir al final del archivo de registro.                                                                                               |
| <b>g</b>           | Ir al inicio del archivo de registro.                                                                                              |
| <b>q</b>           | Salir del archivo de registro.                                                                                                     |
| <b>h</b>           | Conozca otros comandos.                                                                                                            |

<sup>1</sup> Después de escribir **/ ERROR** , escriba **n** para ir al mensaje de error siguiente. Escriba **N** para ir al mensaje de error anterior.

<sup>2</sup> Después de escribir **? ERROR** , escriba **n** para ir al mensaje de error siguiente o **N** para ir al mensaje de error anterior.

También puedes escribir el siguiente comando para abrir un archivo de registro de CodeDeploy scripts:

```
less /opt/codedeploy-agent/deployment-root/deployment-group-ID/deployment-ID/logs/scripts.log
```

Ejecute los comandos siguientes para buscar mensajes de error en el archivo de registro:

| Comando            | Resultado                                                   |
|--------------------|-------------------------------------------------------------|
| <b>&amp;stderr</b> | Muestra solo los mensajes de error del archivo de registro. |
| <b>/stderr</b>     | Busque el mensaje de error siguiente. <sup>1</sup>          |
| <b>?stderr</b>     | Busca el mensaje de error anterior. <sup>2</sup>            |
| <b>G</b>           | Ir al final del archivo de registro.                        |
| <b>g</b>           | Ir al inicio del archivo de registro.                       |
| <b>q</b>           | Salir del archivo de registro.                              |
| <b>h</b>           | Conozca otros comandos.                                     |

Después de escribir **/stderr**, escriba **n** para ir al mensaje de error siguiente hacia delante. Escriba **N** para ir al mensaje de error anterior hacia atrás.

<sup>2</sup> Después de escribir **?stderr**, escriba **n** para ir al mensaje de error siguiente hacia atrás. Escriba **N** para ir al mensaje de error anterior hacia delante.

## Para ver los archivos de registro de implementación en las instancias de Windows Server

CodeDeploy archivo de registro del agente: en las instancias de Windows Server, el archivo de registro del CodeDeploy agente se almacena en la siguiente ubicación:

```
C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Para ver o analizar el archivo de registro del CodeDeploy agente en una instancia de Windows Server, inicie sesión en la instancia y, a continuación, escriba el siguiente comando para abrir el archivo:

```
notepad C:\ProgramData\Amazon\CodeDeploy\log\codedeploy-agent-log.txt
```

Para buscar mensajes de error en el archivo de registro, pulse CTRL+F, escriba **ERROR** [ y, a continuación, pulse Intro para encontrar el primer error.

CodeDeploy archivos de registro de scripts: en las instancias de Windows Server, los registros de implementación se almacenan en la siguiente ubicación:

```
C:\ProgramData\Amazon\CodeDeploy\deployment-group-id\deployment-id\logs\scripts.log
```

Donde:

- *deployment-group-id* es una cadena como `examplebf3a9c7a-7c19-4657-8684-b0c68d0cd3c4`
- *deployment-id* es un identificador similar a `d-12EXAMPLE`

Escriba el siguiente comando para abrir un archivo de registro de CodeDeploy scripts:

```
notepad C:\ProgramData\Amazon\CodeDeploy\deployment-group-ID\deployment-ID\logs\scripts.log
```

Para buscar mensajes de error en el archivo de registro, pulse CTRL+F, escriba **stderr** y, a continuación, pulse Intro para encontrar el primer error.

## Detenga un despliegue con CodeDeploy

Puede usar la CodeDeploy consola AWS CLI, las API o las CodeDeploy API para detener las implementaciones asociadas a su AWS cuenta.

### Warning

Si se detiene una implementación de EC2/en las instalaciones, es posible que algunas o todas las instancias de los grupos de implementación queden en un estado de implementación indeterminado. Para obtener más información, consulte [Implementaciones detenidas y que producen error](#).

Puede detener una implementación o detener y restaurar una implementación.

- [Detención de una implementación \(consola\)](#)
- [Detener una implementación \(CLI\)](#)

**Note**

Si su implementación es de tipo azul/verde AWS CloudFormation, no podrá realizar esta tarea en la consola. CodeDeploy Vaya a la AWS CloudFormation consola para realizar esta tarea.

## Detención de una implementación (consola)

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

**Note**

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y después elija Aplicaciones.

**Note**

Si no se muestra ninguna entrada, asegúrese de seleccionar la región correcta. En la barra de navegación, en el selector de regiones, elija una de las regiones que aparecen en [Región y puntos finales](#) en. Referencia general de AWS CodeDeploy solo se admite en estas regiones.

3. Elija la implementación que desea que deje de hacer una de las siguientes acciones:
  1. Elija Stop deployment (Detener la implementación) para detener la implementación sin realizar una restauración.
  2. Elija Stop and roll back deployment (Detener y restaurar la implementación) para detener y restaurar la implementación.

Para obtener más información, consulte [Vuelva a implementar y revertir una implementación con CodeDeploy](#).

**Note**

Si Stop deployment (Detener la implementación) y Stop and roll back deployment (Detener y restaurar la implementación) aparecen no disponibles, la implementación ha progresado hasta un punto en el que no puede detenerse.

## Detener una implementación (CLI)

Llame al comando [stop-deployment](#) y especifique el ID de la implementación. Para ver una lista de los ID de implementación, llame al comando [list-deployments](#).

## Vuelva a implementar y revertir una implementación con CodeDeploy

CodeDeploy revierte las implementaciones al volver a implementar una revisión de una aplicación previamente implementada como una nueva implementación. Estas implementaciones restauradas son, desde el punto de vista técnico, nuevas implementaciones, con nuevos ID de implementación, en lugar de versiones recuperadas de una implementación anterior.

Las implementaciones se pueden restaurar de forma automática o manual.

### Temas

- [Restauraciones automáticas](#)
- [Restauraciones manuales](#)
- [Flujo de trabajo de la restauración y reimplementación](#)
- [Comportamiento de la restauración con contenido existente](#)

## Restauraciones automáticas

Puede configurar una implementación o un grupo de implementaciones para que se restauren automáticamente si falla una implementación o si se supera un umbral de monitorización que haya especificado. En este caso, se implementa la última versión correcta conocida de una revisión de la aplicación. Puede configurar restauraciones automáticas al crear una aplicación o al crear o actualizar un grupo de implementaciones.

Al crear una nueva implementación, también puede optar por anular la configuración de restauración automática especificada para el grupo de implementaciones.

#### Note

Puede utilizar Amazon Simple Notification Service para recibir una notificación cuando se restaure automáticamente una implementación. Para obtener más información, consulte [Monitoring Deployments with Amazon SNS Event Notifications](#).

Para obtener más información acerca de cómo configurar restauraciones automáticas, consulte [Configuración de las opciones avanzadas de un grupo de implementación](#).

## Restauraciones manuales

Si no ha configurado restauraciones automáticas, puede restaurar manualmente una implementación creando una nueva implementación que utilice cualquier revisión de una aplicación implementada anteriormente y siguiendo los pasos para volver a implementar una revisión. Puede hacerlo si una aplicación se ha quedado en un estado desconocido. En lugar de dedicar mucho tiempo a resolver problemas, puede volver a implementar la aplicación a un estado operativo conocido. Para obtener más información, consulte [Cree una implementación con CodeDeploy](#).

#### Note

Si eliminas una instancia de un grupo de despliegues, CodeDeploy no desinstala nada que pueda estar ya instalado en esa instancia.

## Flujo de trabajo de la restauración y reimplementación

Cuando se inicia la reversión automática, o cuando se inicia manualmente una redistribución o reversión manual, CodeDeploy primero intenta eliminar de cada instancia participante todos los archivos que se instalaron correctamente por última vez. CodeDeploy lo hace comprobando el archivo de limpieza:

Archivo `/opt/codedeploy-agent/deployment-root/deployment-instructions/deployment-group-ID-cleanup` (para las instancias de Amazon Linux, Ubuntu Server y RHEL)



Archivo C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\*deployment-group-ID*-cleanup (para las instancias de Windows Server)

Si existe, CodeDeploy utiliza el archivo de limpieza para eliminar de la instancia todos los archivos de la lista antes de iniciar la nueva implementación.

Por ejemplo, los dos primeros archivos de texto y los dos archivos de script ya estaban implementados una instancia de Amazon EC2 que ejecutaba Windows Server y los scripts crearon dos archivos de texto más durante los eventos del ciclo de vida de la implementación:

```
c:\temp\a.txt (previously deployed by CodeDeploy)
c:\temp\b.txt (previously deployed by CodeDeploy)
c:\temp\c.bat (previously deployed by CodeDeploy)
c:\temp\d.bat (previously deployed by CodeDeploy)
c:\temp\e.txt (previously created by c.bat)
c:\temp\f.txt (previously created by d.bat)
```

El archivo de limpieza enumerará únicamente los dos primeros archivos de texto y dos archivos de script:

```
c:\temp\a.txt
c:\temp\b.txt
c:\temp\c.bat
c:\temp\d.bat
```

Antes de la nueva implementación, CodeDeploy eliminará solo los dos primeros archivos de texto y los dos archivos de script, dejando los dos últimos archivos de texto intactos:

```
c:\temp\a.txt will be removed
c:\temp\b.txt will be removed
c:\temp\c.bat will be removed
c:\temp\d.bat will be removed
c:\temp\e.txt will remain
c:\temp\f.txt will remain
```

Como parte de este proceso, no CodeDeploy intentará revertir ni conciliar ninguna acción realizada por ningún script en implementaciones anteriores durante las redistribuciones posteriores, ya sean reversiones manuales o automáticas. Por ejemplo, si los d.bat archivos c.bat y contienen la lógica de no volver a crear los f.txt archivos e.txt y si ya existen, las versiones anteriores f.txt

permanecerán intactas cada vez que se ejecuten `e.txt` y en las implementaciones posteriores. `CodeDeploy c.bat d.bat` Puede añadir lógica `c.bat` y `d.bat` para comprobar y eliminar siempre las versiones anteriores de `e.txt` y `f.txt` antes de crear otras nuevas.

## Comportamiento de la restauración con contenido existente

Como parte del proceso de implementación, el CodeDeploy agente elimina de cada instancia todos los archivos instalados en la implementación más reciente. Si los archivos que no formaban parte de una implementación anterior aparecen en las ubicaciones de implementación de destino, puede elegir CodeDeploy qué pasará con ellos durante la siguiente implementación:

- Dar como errónea la implementación: se notifica un error y el estado de la implementación cambia a `Error`.
- Sobrescribir el contenido: la versión del archivo de la revisión de la aplicación sustituye a la versión ya incluida en la instancia.
- Conservar el contenido: el archivo de la ubicación de destino se conserva y la versión de la revisión de la aplicación no se copia en la instancia.

Puede elegir este comportamiento si crea una implementación. Si está creando una implementación en la consola, consulte [Creación de una implementación en una plataforma de informática de EC2/en las instalaciones \(consola\)](#). Si va a crear una implementación con AWS CLI, consulte [Creación de una implementación en una plataforma de informática EC2/en las instalaciones \(CLI\)](#).

Puede optar por conservar los archivos que desea que formen parte de la próxima implementación sin tener que añadirlos al paquete de revisión de la aplicación. Por ejemplo, podría cargar directamente en la instancia los archivos que son necesarios para la implementación pero que no se añadieron al paquete de revisión de la aplicación. O bien, puede cargar archivos a la instancia si sus aplicaciones ya están en su entorno de producción, pero quiere utilizarlos CodeDeploy por primera vez para implementarlas.

En el caso de las restauraciones, en las que se vuelve a implementar la última revisión de la aplicación implementada correctamente debido a un error de implementación, se aplica la opción de gestión de contenido de la última implementación correcta a la implementación de restauración.

Sin embargo, si la implementación errónea se configuró para sobrescribir, en lugar de para conservar los archivos, podría producirse un resultado inesperado durante la restauración. En concreto, los archivos que esperaba conservar puede que los elimine la implementación errónea. Los archivos no están en la instancia cuando se ejecuta la implementación de restauración.

En el siguiente ejemplo, existen tres implementaciones. Cualquier archivo que se sobrescribe (elimina) durante la segunda implementación errónea ya no estará disponible (no se puede conservar) cuando se implemente de nuevo la 1ª revisión de la aplicación durante la 3ª implementación:

| Implementación    | Revisión de la aplicación    | Opción de sobrescritura del contenido | Estado de la implementación | Comportamiento y resultado                                                                                                                                                                                                                                                                        |
|-------------------|------------------------------|---------------------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1ª implementación | 1ª revisión de la aplicación | RETAIN                                | Succeeded                   | CodeDeploy detecta archivos en las ubicaciones de destino que no se desplegaron en la implementación anterior. Estos archivos podrían colocarse allí deliberadamente para que formen parte de la implementación actual. Se conservan y registran como parte del paquete de implementación actual. |
| 2ª implementación | 2ª revisión de la aplicación | OVERWRITE                             | Con error                   | Durante el proceso de implementación, CodeDeploy elimina todos                                                                                                                                                                                                                                    |

| Implementación | Revisión de la aplicación | Opción de sobrescritura del contenido | Estado de la implementación | Comportamiento y resultado                                                                                                                                                                                           |
|----------------|---------------------------|---------------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                |                           |                                       |                             | <p>los archivos que forman parte de la implementación anterior exitosa. Esto incluye los archivos que se conservan durante la 1ª implementación.</p> <p>Sin embargo, la implementación falla por motivos ajenos.</p> |

| Implementación    | Revisión de la aplicación    | Opción de sobrescritura del contenido | Estado de la implementación | Comportamiento y resultado                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|------------------------------|---------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3ª implementación | 1ª revisión de la aplicación | RETAIN                                |                             | <p>Como la reversión automática está habilitada para la implementación o el grupo de implementaciones, CodeDeploy y implementa a la última revisión válida de la aplicación conocida, la revisión 1 de la aplicación.</p> <p>Sin embargo, los archivos que quería conservar en la implementación 1 se eliminarán antes de que la implementación 2 fallara y no se pueden recuperar. AWS CodeDeploy Puede añadirlos usted mismo a la instancia si son necesarios para</p> |

| Implementación | Revisión de la aplicación | Opción de sobrescritura del contenido | Estado de la implementación | Comportamiento y resultado                                                          |
|----------------|---------------------------|---------------------------------------|-----------------------------|-------------------------------------------------------------------------------------|
|                |                           |                                       |                             | la 1ª revisión de la aplicación, o puede crear una nueva revisión de la aplicación. |

## Implementación de una aplicación en otra cuenta de AWS

Las organizaciones suelen tener varias AWS cuentas que utilizan para distintos fines (por ejemplo, una para las tareas de administración del sistema y otra para las tareas de desarrollo, pruebas y producción, o una asociada a los entornos de desarrollo y prueba y otra asociada al entorno de producción).

Aunque puede realizar trabajos relacionados en diferentes cuentas, los grupos de CodeDeploy despliegue y las instancias de Amazon EC2 en las que se despliegan están estrictamente vinculados a las cuentas en las que se crearon. Por ejemplo, no se puede añadir una instancia que lanzó en una cuenta a un grupo de implementaciones en otra.

Suponga que tiene dos AWS cuentas: la cuenta de desarrollo y la cuenta de producción. Trabaja principalmente en la cuenta de desarrollo, pero desea poder iniciar implementaciones en su cuenta de producción sin un conjunto completo de credenciales allí o sin tener que salir de la cuenta de desarrollo e iniciar sesión en la cuenta de producción.

Después de seguir los pasos de configuración entre cuentas, puede iniciar implementaciones que pertenecen a otra cuenta de la organización, sin necesitar un conjunto completo de credenciales para esa otra cuenta. Esto es posible, en parte, utilizando una capacidad proporcionada en el AWS Security Token Service (AWS STS) que le concede acceso temporal a esa cuenta.

### Paso 1: Crear un bucket de S3 en cualquiera de las cuentas

En la cuenta de desarrollo o en la cuenta de producción:

- Si aún no lo ha hecho, cree un bucket de Amazon S3 en el cual se almacenarán las revisiones de la aplicación para la cuenta de producción. Para obtener información, consulte [Crear un bucket](#)

[en Amazon S3](#). Incluso, puede utilizar el mismo bucket y revisiones de la aplicación para ambas cuentas, implementando los mismos archivos a su entorno de producción que ya probó y verificó en la cuenta de desarrollo.

## Paso 2: Conceder permisos de bucket de Amazon S3 al perfil de instancia de IAM de la cuenta de producción

Si el bucket de Amazon S3 que creó en el paso 1 está en la cuenta de producción, este paso no es necesario. El rol que asuma después ya tendrá acceso a este bucket porque también estará en la cuenta de producción.

Si creó el bucket de Amazon S3 en la cuenta de desarrollo, haga lo siguiente:

- En la cuenta de producción, cree un perfil de instancia de IAM. Para obtener más información, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#).

### Note

Anote el ARN de este perfil de instancia de IAM. Deberá añadirlo a la política de buckets cruzados que cree a continuación.

- En la cuenta de desarrollo, conceda acceso al bucket de Amazon S3 que creó en la cuenta de desarrollo al perfil de instancia de IAM que acaba de crear en la cuenta de producción. Para obtener más información, consulte [Ejemplo 2: Propietario del bucket que concede permisos de bucket entre cuentas](#).

Tenga en cuenta lo siguiente cuando complete el proceso de conceder permisos de bucket entre cuentas:

- En el ejemplo del tutorial, la Cuenta A representa la cuenta de desarrollo y la Cuenta B representa la cuenta de producción.
- Cuando [realice las tareas de la cuenta A \(cuenta de desarrollo\)](#), modifique la siguiente política de bucket para conceder permisos entre cuentas en lugar de utilizar la política de ejemplo del tutorial.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
```

```
"Sid": "Cross-account permissions",
"Effect": "Allow",
"Principal": {
 "AWS": "arn:aws:iam::account-id:role/role-name"
},
"Action": [
 "s3:Get*",
 "s3:List*"
],
"Resource": [
 "arn:aws:s3:::bucket-name/*"
]
}
]
```

*account-id* representa el número de cuenta de la cuenta de producción en la que acaba de crear el perfil de instancia de IAM.

*role-name* representa el nombre del perfil de instancia de IAM que acaba de crear.

*bucket-name* representa el nombre del bucket que creó en el paso 1. Asegúrese de incluir `/*` después del nombre de su bucket para conceder acceso a cada uno de los archivos dentro del bucket.

### Paso 3: Crear recursos y un rol entre cuentas en la cuenta de producción

En su cuenta de producción:

- Cree sus CodeDeploy recursos (aplicación, grupo de implementación, configuración de implementación, instancias de Amazon EC2, perfil de instancia de Amazon EC2, rol de servicio, etc.) siguiendo las instrucciones de esta guía.
- Cree una función adicional, una función de IAM multicuenta, que un usuario de su cuenta de desarrollo pueda asumir para realizar CodeDeploy operaciones en esta cuenta de producción.

Usa el [tutorial: Delega el acceso entre AWS cuentas utilizando las funciones de IAM](#) como guía para ayudarte a crear la función multicuenta. En lugar de añadir a tu documento de política los permisos de ejemplo que aparecen en el tutorial, deberías adjuntar al rol, como mínimo, las dos políticas AWS siguientes:



- `AmazonS3FullAccess`: necesaria solamente si el bucket de S3 está en la cuenta de desarrollo. Proporciona el rol asumido de la cuenta de producción con acceso total a los servicios y recursos de Amazon S3 en la cuenta de desarrollo donde se almacena la revisión.
- `AWSCodeDeployDeployerAccess`: permite a un usuario registrar e implementar revisiones.

Si desea crear y administrar grupos de implementación y no solamente iniciar implementaciones, añada la política `AWSCodeDeployFullAccess` en lugar de la política `AWSCodeDeployDeployerAccess`. Para obtener más información sobre el uso de las políticas gestionadas de IAM para conceder permisos para CodeDeploy las tareas, consulte. [AWS políticas gestionadas \(predefinidas\) para CodeDeploy](#)

Puede adjuntar políticas adicionales si desea ejecutar tareas en otros servicios de AWS mientras utiliza este rol entre cuentas.

#### Important

Cuando cree el rol de IAM entre cuentas, anote los detalles que necesitará para acceder a la cuenta de producción.

Para utilizarlas AWS Management Console para cambiar de rol, tendrás que proporcionar uno de los siguientes datos:

- Una URL para acceder a la cuenta de producción con las credenciales del rol asumido. Encontrará la URL en la página Review, que aparece al final del proceso de creación de rol entre cuentas.
- El nombre del rol entre cuentas y el número del ID de la cuenta o el alias.

Para usar el AWS CLI para cambiar de rol, necesitará proporcionar lo siguiente:

- El ARN del rol entre cuentas que asumirá.

## Paso 4: Cargar la revisión de la aplicación al bucket de Amazon S3

En la cuenta en la que creó el bucket de Amazon S3:

- Cargue la revisión de la aplicación al bucket de Amazon S3. Para obtener más información, consulte [Enviar una revisión CodeDeploy a Amazon S3 \(solo para implementaciones de EC2/on-premise\)](#).

## Paso 5: Asumir el rol entre cuentas e implementar las aplicaciones

En la cuenta de desarrollo, puede utilizar la AWS CLI o la AWS Management Console para asumir la función multicuenta e iniciar el despliegue en la cuenta de producción.

Para obtener instrucciones sobre cómo usarlo AWS Management Console para cambiar de rol e iniciar despliegues, consulte [Cambiar a un rol \(AWS Management Console\)](#) y [Creación de una implementación en una plataforma de informática de EC2/en las instalaciones \(consola\)](#)

Para obtener instrucciones sobre cómo utilizarla AWS CLI para asumir la función multicuenta e iniciar las implementaciones, consulte [Cambiar a una función de IAM \(\)](#) y [AWS Command Line Interface Creación de una implementación en una plataforma de informática EC2/en las instalaciones \(CLI\)](#)

[Para obtener más información sobre cómo asumir un rol AWS STS, consulte la Guía del AWS Security Token Service usuario y asume-role AssumeRole en la Referencia de comandos.AWS CLI](#)

Tema relacionado:

- [CodeDeploy: Implementación desde una cuenta de desarrollo a una cuenta de producción](#)

## Utilice el CodeDeploy agente para validar un paquete de despliegue en una máquina local

Con el CodeDeploy agente, puede implementar contenido en una instancia en la que haya iniciado sesión. Esto le permite probar la integridad del archivo de especificaciones de la aplicación (AppSpec archivo) que pretende utilizar en una implementación y el contenido que pretende implementar.

No necesita crear una aplicación ni un grupo de implementaciones. Si desea implementar contenido almacenado en la instancia local, ni siquiera necesita una AWS cuenta. Para realizar las pruebas más sencillas, puede ejecutar el `codedeploy-local` comando, sin especificar ninguna opción, en un directorio que contenga el AppSpec archivo y el contenido que se va a implementar. Existen opciones para otros escenarios de prueba en la herramienta.

Mediante la validación de un paquete de implementación en una máquina local, puede:

- Probar la integridad de una revisión de la aplicación
- Pruebe el contenido de un AppSpec archivo.
- Pruébalo CodeDeploy por primera vez con el código de su aplicación actual.
- Implementar contenido rápidamente cuando ya ha iniciado sesión en una instancia.

Puede utilizar el contenido de despliegue almacenado en la instancia local o en un tipo de repositorio remoto compatible (depósitos de Amazon S3 o GitHub repositorios públicos).

## Requisitos previos

Antes de iniciar una implementación local, siga los pasos que se indican a continuación:

- Cree o utilice un tipo de instancia compatible con el CodeDeploy agente. Para obtener más información, consulte [Sistemas operativos compatibles con el agente CodeDeploy](#).
- Instale la versión 1.0.1.1352 o posterior del agente. CodeDeploy Para obtener más información, consulte [Instala el CodeDeploy agente](#).
- Si va a implementar su contenido desde un GitHub depósito o repositorio de Amazon S3, provisione un usuario para usarlo con él CodeDeploy. Para obtener más información, consulte [Paso 1: Configurar](#).
- Si va a implementar la revisión de la aplicación desde un bucket de Amazon S3, cree un bucket de Amazon S3 en la región en la que está trabajando y aplique una política de bucket de Amazon S3 al bucket. Esta política concede a las instancias los permisos necesarios para descargar la revisión de la aplicación.

Por ejemplo, la siguiente política de bucket de Amazon S3 permite a cualquier instancia de Amazon EC2 asociada a un perfil de instancia de IAM que contenga el ARN `arn:aws:iam::444455556666:role/CodeDeployDemo` descargar desde cualquier lugar al bucket de Amazon S3 llamado `codedeploydemobucket`:

```
{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
```

```

 "Resource": "arn:aws:s3:::codedeploydemobucket/*",
 "Principal": {
 "AWS": [
 "arn:aws:iam::444455556666:role/CodeDeployDemo"
]
 }
]
}

```

La siguiente política de bucket de Amazon S3 permite a cualquier instancia en las instalaciones con un usuario de IAM asociado que contenga el ARN `arn:aws:iam::444455556666:user/CodeDeployUser` descargar desde cualquier lugar al bucket de Amazon S3 denominado `codedeploydemobucket`:

```

{
 "Statement": [
 {
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::codedeploydemobucket/*",
 "Principal": {
 "AWS": [
 "arn:aws:iam::444455556666:user/CodeDeployUser"
]
 }
 }
]
}

```

Para obtener información acerca de cómo generar y adjuntar una política de bucket de Amazon S3, consulte [Ejemplos de política de bucket](#).

- Si va a implementar la revisión de la aplicación desde un GitHub depósito o repositorio de Amazon S3, configure un perfil de instancia de IAM y adjúntelo a la instancia. Para obtener información, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#), [Cree una instancia de Amazon EC2 para CodeDeploy \(AWS CLI o una consola Amazon EC2\)](#) y [Crear una instancia de Amazon EC2 para CodeDeploy \(plantilla\)AWS CloudFormation](#).

- Si va a implementar su contenido desde GitHub, cree una GitHub cuenta y un repositorio público. Para crear una GitHub cuenta, consulta [Unirse GitHub](#). Para crear un GitHub repositorio, consulta [Crear un repositorio](#).

#### Note

Actualmente no se admiten repositorios privados. Si tu contenido está almacenado en un GitHub repositorio privado, puedes descargarlo en la instancia y usar la `--bundle-location` opción para especificar su ruta local.

- Prepare el contenido (incluido un AppSpec archivo) que desee implementar en la instancia y colóquelo en la instancia local, en su bucket de Amazon S3 o en su GitHub repositorio. Para obtener más información, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).
- Si desea utilizar valores distintos de los predeterminados para otras opciones de configuración, cree el archivo de configuración y colóquelo en la instancia (`/etc/codedeploy-agent/conf/codedeployagent.yml` para instancias de Amazon Linux, RHEL o Ubuntu Server o `C:\ProgramData\Amazon\CodeDeploy\conf.yml` para instancias de Windows Server). Para obtener más información, consulte [CodeDeploy referencia de configuración del agente](#).

#### Note

Si utiliza un archivo de configuración en instancias de Amazon Linux, RHEL o Ubuntu Server, debe:

- Usar las variables `:root_dir:` y `:log_dir:` para especificar ubicaciones distintas de las predeterminadas para la carpeta raíz de la implementación y las carpetas del directorio de archivos log
- Se usa `sudo` para ejecutar comandos de CodeDeploy agente.

## Creación de una implementación local

En la instancia en la que desea crear la implementación local, abra una sesión del terminal (para instancias de Amazon Linux, RHEL o Ubuntu Server) o un símbolo del sistema (para Windows Server) para ejecutar los comandos de la herramienta.

**Note**

El comando `codedeploy-local` se instala en las siguientes ubicaciones:

- Para Amazon Linux, RHEL y Ubuntu Server: `/opt/codedeploy-agent/bin`.
- Para Windows Server: `C:\ProgramData\Amazon\CodeDeploy\bin`.

## Sintaxis básica de los comandos

```
codedeploy-local [options]
```

## Sinopsis

```
codedeploy-local
[--bundle-location <value>]
[--type <value>]
[--file-exists-behavior <value>]
[--deployment-group <value>]
[--events <comma-separated values>]
[--agent-configuration-file <value>]
[--appspec-filename <value>]
```

## Opciones

`-l, --bundle-location`

La ubicación del paquete de revisión de la aplicación. Si no especifica una ubicación, la herramienta utiliza el directorio en el que está trabajando actualmente de forma predeterminada. Si especifica un valor en `--bundle-location`, también debe especificar uno en `--type`.

Ejemplos de formato de ubicación del paquete:

- Instancia local de Amazon Linux, RHEL o Ubuntu Server: `/path/to/local/bundle.tgz`
- Instancia local de Windows Server: `C:/path/to/local/bundle`
- Bucket de Amazon S3: `s3://mybucket/bundle.tar`
- GitHub repositorio: `https://github.com/<account-name>/<repository-name>/`

`-t, --type`

El formato del paquete de revisión de la aplicación. Los tipos admitidos son `tgz`, `tar`, `zip` y `directory`. Si no especifica un tipo, la herramienta usa `directory` de manera predeterminada. Si especifica un valor en `--type`, también debe especificar uno en `--bundle-location`.

`-b, --file-exists-behavior`

Indica cómo se tratan los archivos que ya existen en la ubicación de destino de una implementación pero que no forman parte de una implementación anterior realizada correctamente. Las opciones son `DISALLOW`, `OVERWRITE`, `RETAIN`. Para obtener más información, consulte la [Referencia fileExistsBehavior de la AWS CodeDeploy API](#).

`-g, --deployment-group`

La ruta a la carpeta que es la ubicación de destino del contenido que se va a implementar. Si no especificas una carpeta, la herramienta crea una con un nombre en `default-local-deployment-group` el directorio raíz de la implementación. Para cada implementación local que cree, la herramienta crea un subdirectorio dentro de esta carpeta con nombres como `d-98761234-local`.

`-e, --events`

Un conjunto de enlaces de eventos de anulación del ciclo de vida que desee ejecutar, en orden, en lugar de los eventos que enumeró en el AppSpec archivo. Se pueden especificar varios enlaces, separados por comas. Puede utilizar esta opción si:

- Quieres ejecutar un conjunto diferente de eventos sin tener que actualizar el AppSpec archivo.
- Desea ejecutar un único enlace de eventos como una excepción a lo que hay en el AppSpec archivo, por ejemplo `ApplicationStop`.

Si no especificas `DownloadBundle` ni instalas los eventos en la lista de anulaciones, se ejecutarán antes que todos los enlaces de eventos que especifiques. Si incluyes `DownloadBundle` `Install` en la lista de `--events` opciones, solo deben ir precedidas de eventos que normalmente se ejecutan antes que ellos en las CodeDeploy implementaciones. Para obtener más información, consulte [AppSpec sección de «ganchos»](#).

`-c, --agent-configuration-file`

La ubicación de un archivo de configuración que se utiliza para la implementación, si está almacenado en una ubicación distinta de la predeterminada. Un archivo de configuración especifica alternativas a otros valores y comportamientos predeterminados de una implementación.

De forma predeterminada, los archivos de configuración se almacenan en `/etc/codedeploy-agent/conf/codedeployagent.yml` (instancias de Amazon Linux, RHEL o Ubuntu Server) o en `C:/ProgramData/Amazon/CodeDeploy/conf.yml` (Windows Server). Para obtener más información, consulte [CodeDeploy referencia de configuración del agente](#).

`-A, --appspec-filename`

El nombre del AppSpec archivo. Para las implementaciones locales, los valores aceptados son `appspec.yml` y `appspec.yaml`. De forma predeterminada, se llama al AppSpec archivo `appspec.yml`.

`-h, --help`

Muestra un resumen del contenido de ayuda.

`-v, --version`

Muestra el número de versión de la herramienta.

## Ejemplos

A continuación se proporcionan ejemplos de formatos de comandos válidos.

```
codedeploy-local
```

```
codedeploy-local --bundle-location /path/to/local/bundle/directory
```

```
codedeploy-local --bundle-location C:/path/to/local/bundle.zip --type zip --deployment-group my-deployment-group
```

```
codedeploy-local --bundle-location /path/to/local/directory --type directory --deployment-group my-deployment-group
```

Implementar un paquete desde Amazon S3:

```
codedeploy-local --bundle-location s3://mybucket/bundle.tgz --type tgz
```

```
codedeploy-local --bundle-location s3://mybucket/bundle.zip?versionId=1234&etag=47e8 --type zip --deployment-group my-deployment-group
```



## Implemente un paquete desde un GitHub repositorio público:

```
codedeploy-local --bundle-location https://github.com/awslabs/aws-codedeploy-sample-tomcat --type zip
```

```
codedeploy-local --bundle-location https://api.github.com/repos/awslabs/aws-codedeploy-sample-tomcat/zipball/master --type zip
```

```
codedeploy-local --bundle-location https://api.github.com/repos/awslabs/aws-codedeploy-sample-tomcat/zipball/HEAD --type zip
```

```
codedeploy-local --bundle-location https://api.github.com/repos/awslabs/aws-codedeploy-sample-tomcat/zipball/1a2b3c4d --type zip
```

## Implementar un paquete especificando varios eventos del ciclo de vida:

```
codedeploy-local --bundle-location /path/to/local/bundle.tar --type tar --application-folder my-deployment --events DownloadBundle,Install,ApplicationStart,HealthCheck
```

## Detenga una aplicación implementada anteriormente mediante el evento de ApplicationStop ciclo de vida:

```
codedeploy-local --bundle-location /path/to/local/bundle.tgz --type tgz --deployment-group --events ApplicationStop
```

## Implementar con un ID de grupo de implementaciones específico:

```
codedeploy-local --bundle-location C:/path/to/local/bundle/directory --deployment-group 1234abcd-5dd1-4774-89c6-30b107ac5dca
```

```
codedeploy-local --bundle-location C:/path/to/local/bundle.zip --type zip --deployment-group 1234abcd-5dd1-4774-89c6-30b107ac5dca
```

# Supervisión de los despliegues en CodeDeploy

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de CodeDeploy sus AWS soluciones. Debe recopilar los datos de supervisión de todas las partes de la AWS solución para poder depurar más fácilmente una falla multipunto en caso de que se produzca. Sin embargo CodeDeploy, antes de empezar a monitorizar, debe crear un plan de monitorización que incluya respuestas a las siguientes preguntas:

- ¿Cuáles son los objetivos de la supervisión?
- ¿Qué recursos va a supervisar?
- ¿Con qué frecuencia va a supervisar estos recursos?
- ¿Qué herramientas de monitoreo va a utilizar?
- ¿Quién se encargará de realizar las tareas de monitoreo?
- ¿Quién debería recibir una notificación cuando surjan problemas?

El siguiente paso es establecer una línea base para el CodeDeploy rendimiento normal de su entorno, midiendo el rendimiento en distintos momentos y bajo diferentes condiciones de carga. Mientras supervisa CodeDeploy, almacene los datos de supervisión históricos para poder compararlos con los datos de rendimiento actuales, identificar los patrones de rendimiento normales y las anomalías de rendimiento, y diseñar métodos para abordar los problemas.

Por ejemplo, si lo utiliza CodeDeploy, puede supervisar el estado de las implementaciones y las instancias de destino. Cuando las implementaciones o las instancias fallan, es posible que tenga que volver a configurar un archivo de especificaciones de la aplicación, reinstalar o actualizar el CodeDeploy agente, actualizar la configuración de una aplicación o un grupo de implementaciones, o realizar cambios en la configuración de la instancia o en un archivo. AppSpec

Para establecer un punto de referencia debe, como mínimo, monitorizar los elementos siguientes:

- Eventos y estado de la implementación
- Eventos y estado de la instancia

## Herramientas de monitoreo automatizadas

AWS proporciona varias herramientas que puede utilizar para supervisar. CodeDeploy Puede configurar algunas de estas herramientas para que monitoricen por usted, pero otras herramientas

requieren intervención manual. Le recomendamos que automatice las tareas de supervisión en la medida de lo posible.

Puede utilizar las siguientes herramientas de supervisión automatizadas para observar CodeDeploy e informar cuando algo va mal:

- Amazon CloudWatch Alarms: observe una sola métrica durante un período de tiempo que especifique y realice una o más acciones en función del valor de la métrica en relación con un umbral determinado durante varios períodos de tiempo. La acción es una notificación enviada a un tema del Servicio de Notificación Simple (Amazon SNS) o a una política de Auto Scaling de Amazon EC2. CloudWatch las alarmas no invocan acciones simplemente porque se encuentran en un estado determinado; el estado debe haber cambiado y se ha mantenido durante un número específico de períodos. Para obtener más información, consulte [Monitoring Deployments with Amazon CloudWatch Tools](#).

Para obtener información sobre cómo actualizar su función de servicio para que funcione con la supervisión de CloudWatch alarmas, consulte [Otorgue CloudWatch permisos a un rol CodeDeploy de servicio](#). Para obtener información sobre cómo añadir la supervisión de CloudWatch alarmas a sus CodeDeploy operaciones [Cree una aplicación con CodeDeploy](#), consulte [Cree un grupo de despliegue con CodeDeploy](#), o [Cambie la configuración del grupo de implementación con CodeDeploy](#).

- Amazon CloudWatch Logs: supervise, almacene y acceda a sus archivos de registro desde AWS CloudTrail u otras fuentes. Para obtener más información, consulte [Supervisión de archivos de registro](#) en la Guía del CloudWatch usuario de Amazon.

Para obtener información sobre el uso de la CloudWatch consola para ver CodeDeploy los registros, consulte [Ver CodeDeploy los registros en la consola de CloudWatch registros](#).

- Amazon CloudWatch Events: haga coincidir los eventos y diríjalos a una o más funciones o transmisiones de destino para realizar cambios, capturar información de estado y tomar medidas correctivas. Para obtener más información, consulta [Qué es Amazon CloudWatch Events](#) en la Guía del CloudWatch usuario de Amazon.

Para obtener información sobre el uso de CloudWatch Events en sus CodeDeploy operaciones, consulte [Supervisión de despliegues con Amazon Events CloudWatch](#).

- AWS CloudTrail Supervisión de registros: comparta archivos de registro entre cuentas, supervise los archivos de CloudTrail registro en tiempo real enviándolos a CloudWatch Logs, cree aplicaciones de procesamiento de registros en Java y valide que los archivos de registro no hayan

cambiado después de su entrega CloudTrail. Para obtener más información, consulte [Trabajar con archivos de CloudTrail registro](#) en la Guía del AWS CloudTrail usuario.

Para obtener información sobre su uso CloudTrail con CodeDeploy, consulte [Monitoring Deployments](#).

- Amazon Simple Notification Service: configure disparadores basados en eventos para recibir notificaciones de correo electrónico o SMS sobre eventos de implementaciones e instancias, como el éxito o el error. Para obtener más información, consulte [Crear un tema](#) y [¿Qué es Amazon Simple Notification Service?](#).

Para obtener información sobre cómo configurar las notificaciones de Amazon SNS para CodeDeploy, consulte. [Monitoring Deployments with Amazon SNS Event Notifications](#)

## Herramientas de monitoreo manuales

Otra parte importante del monitoreo CodeDeploy consiste en monitorear manualmente los elementos que las CloudWatch alarmas no cubren. El CodeDeploy panel de control y otros paneles de la AWS consola proporcionan una at-a-glance vista del estado de su AWS entorno. CloudWatch Le recomendamos que compruebe también los archivos de registro de las CodeDeploy implementaciones.

- CodeDeploy la consola muestra:
  - El estado de las implementaciones
  - La fecha y la hora de cada uno de los últimos intentos de implementación y la última implementación correcta de una revisión
  - El número de instancias correctas, erróneas, omitidas o en curso de una implementación
  - El estado de las instancias on-premise
  - La fecha y la hora en las que se han registrado o anulado las instancias on-premise
- CloudWatch la página de inicio muestra:
  - Alarmas y estado actual
  - Gráficos de alarmas y recursos
  - Estado de los servicios

Además, se puede utilizar CloudWatch para hacer lo siguiente:

- [Crear paneles personalizados](#) para monitorizar los servicios que le interesan

- Realizar un gráfico con los datos de las métricas para resolver problemas y descubrir tendencias
- Busque y explore todas sus métricas AWS de recursos
- Crear y editar las alarmas de notificación de problemas

## Temas

- [Monitoring Deployments with Amazon CloudWatch Tools](#)
- [Monitoring Deployments](#)
- [Monitoring Deployments with Amazon SNS Event Notifications](#)

# Supervisión de las implementaciones con las herramientas de Amazon CloudWatch

Puede supervisar CodeDeploy las implementaciones mediante las siguientes CloudWatch herramientas: Amazon CloudWatch Events, CloudWatch alarmas y Amazon CloudWatch Logs.

Revisar los registros creados por el CodeDeploy agente y las implementaciones puede ayudarle a solucionar las causas de los errores de implementación. Como alternativa a revisar CodeDeploy los registros de una instancia a la vez, puede usar CloudWatch los registros para monitorear todos los registros en una ubicación central.

Para obtener información sobre el uso de CloudWatch alarmas y CloudWatch eventos para monitorear CodeDeploy las implementaciones, consulta los siguientes temas.

## Temas

- [Supervisión de las implementaciones con CloudWatch alarmas en CodeDeploy](#)
- [Supervisión de despliegues con Amazon Events CloudWatch](#)

# Supervisión de las implementaciones con CloudWatch alarmas en CodeDeploy

Puede crear una CloudWatch alarma para una instancia o grupo de Auto Scaling de Amazon EC2 que utilice en sus CodeDeploy operaciones. Una alarma vigila una única métrica durante el período especificado y realiza una o varias acciones según el valor de la métrica relativo a un determinado

umbral durante varios períodos de tiempo. CloudWatch las alarmas invocan acciones cuando su estado cambia (por ejemplo, de OK aALARM).

Con la funcionalidad de CloudWatch alarma nativa, puede especificar cualquiera de las acciones que se admiten CloudWatch cuando se produce un error en una instancia que está utilizando en una implementación, como enviar una notificación de Amazon SNS o detener, terminar, reiniciar o recuperar una instancia. Para sus CodeDeploy operaciones, puede configurar un grupo de despliegues para que detenga un despliegue cada vez que se active cualquier CloudWatch alarma que asocie al grupo de despliegues.

Puede asociar hasta diez CloudWatch alarmas a un grupo CodeDeploy de despliegue. Si se activa cualquiera de las alarmas especificadas, la implementación se detiene y el estado se actualiza a Stopped. Para utilizar esta opción, debe conceder CloudWatch permisos a su función CodeDeploy de servicio.

Para obtener información sobre la configuración de CloudWatch alarmas en la CloudWatch consola, consulte [Creación de CloudWatch alarmas de Amazon](#) en la Guía del CloudWatch usuario de Amazon.

Para obtener información sobre cómo asociar una CloudWatch alarma a un grupo de despliegues en CodeDeploy, consulte [Cree un grupo de despliegue con CodeDeploy](#) y [Cambie la configuración del grupo de implementación con CodeDeploy](#).

## Temas

- [Otorgue CloudWatch permisos a un rol CodeDeploy de servicio](#)

## Otorgue CloudWatch permisos a un rol CodeDeploy de servicio

Antes de poder utilizar la supervisión de CloudWatch alarmas en sus despliegues, debe concederse permiso a la función de servicio que utilice en sus CodeDeploy operaciones para acceder a los CloudWatch recursos.

Para conceder CloudWatch permisos a un rol de servicio

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la consola de IAM, elija Roles.
3. Elija el nombre de la función de servicio que utiliza en sus AWS CodeDeploy operaciones.

4. En la pestaña Permissions, en el área Inline Policies, elija Create Role Policy.

–o bien–

Si el botón Create Role Policy no está disponible, expanda el área Inline Policies y después elija [click here](#).

5. En la página Set Permissions, elija Custom Policy y después Select.
6. En la página Review Policy, en el campo Policy Name, escriba un nombre que identifique esta política, como CWAAlarms.
7. Pegue la siguiente información en el campo Policy Document:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "cloudwatch:DescribeAlarms",
 "Resource": "*"
 }
]
}
```

8. Seleccione Apply Policy.

## Supervisión de despliegues con Amazon Events CloudWatch

Puede usar Amazon CloudWatch Events para detectar cambios en el estado de una instancia o un despliegue (un «evento») en sus CodeDeploy operaciones y reaccionar ante ellos. Luego, en función de las reglas que cree, CloudWatch Events invocará una o más acciones objetivo cuando una implementación o instancia entre en el estado que especifique en una regla. En función del tipo de cambio de estado, es posible que desee enviar notificaciones, capturar información de estado, tomar medidas correctivas, iniciar eventos o adoptar otras medidas. Puede seleccionar los siguientes tipos de objetivos cuando utilice CloudWatch Events como parte de sus CodeDeploy operaciones:

- AWS Lambda funciones
- Flujos de Kinesis
- Colas de Amazon SQS

- Destinos integrados (EC2 CreateSnapshot API call, EC2 RebootInstances API call, EC2 StopInstances API call y EC2 TerminateInstances API call)
- Temas de Amazon SNS

A continuación se indican algunos casos de uso:

- Utilice una función Lambda para pasar una notificación a un canal de Slack siempre que se produzcan errores en las implementaciones.
- Envíe datos acerca de las implementaciones o instancias a un flujo de Kinesis para permitir una monitorización completa del estado en tiempo real.
- Utilice acciones de CloudWatch alarma para detener, finalizar, reiniciar o recuperar automáticamente las instancias de Amazon EC2 cuando se produzca un evento de implementación o instancia que especifique.

En el resto de este tema se describe el procedimiento básico para crear una regla de CloudWatch eventos para CodeDeploy. Sin embargo, antes de crear reglas de eventos para utilizarlas en sus CodeDeploy operaciones, debe hacer lo siguiente:

- Complete los requisitos previos de los CloudWatch eventos. Para obtener más información, consulte los [requisitos previos de Amazon CloudWatch Events](#).
- Familiarícese con los eventos, las reglas y los objetivos de CloudWatch Events. Para obtener más información, consulta [¿Qué es Amazon CloudWatch Events?](#) y [nuevos CloudWatch eventos: realice un seguimiento de los cambios en sus AWS recursos y responda a ellos](#).
- Crear el destino o los destinos que se utilizarán en las reglas de eventos.

Para crear una regla de CloudWatch eventos para CodeDeploy:

1. Abra la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, elija Events (Eventos).
3. Elija Create rule (Crear regla) y, a continuación, en Event selector (Selector de eventos), elija AWS CodeDeploy.
4. Especifique un tipo de detalle:
  - Para crear una regla que se aplique a todos los cambios de estado de las instancias y las implementaciones, elija Any detail type (Cualquier tipo de detalle) y continúe en el paso 6.



- Para crear una regla que se aplique solo a las instancias, elija un tipo de detalle específico y, a continuación, elija Notificación de cambio de estado de la CodeDeploy instancia.
  - Para crear una regla que se aplique únicamente a las implementaciones, elija un tipo de detalle específico y, a continuación, elija Notificación de cambio de estado de la CodeDeploy implementación.
5. Especifique los cambios de estado a los que se aplica la regla:
- Para crear una regla que se aplique a todos los cambios de estado, elija Any state (Cualquier estado).
  - Para crear una regla que se aplique únicamente a algunos cambios de estado, seleccione Specific state(s) (Estados específicos) y luego elija uno o varios valores de estado de la lista. En la siguiente tabla se muestran los valores de estado que puede elegir:

| Valores de estado de implementación | Valores de estado de instancia |
|-------------------------------------|--------------------------------|
| FAILURE                             | FAILURE                        |
| START                               | START                          |
| STOP                                | READY                          |
| QUEUED                              | SUCCESS                        |
| READY                               |                                |
| SUCCESS                             |                                |

6. Especifique a qué CodeDeploy aplicaciones se aplica la regla:
- Para crear una regla que se aplique a todas las aplicaciones, elija Any application (Cualquier aplicación) y continúe en el paso 8.
  - Para crear una regla que se aplique únicamente a una aplicación, seleccione Specific application (Aplicación específica) y luego elija el nombre de la aplicación en la lista.
7. Especifique los grupos de implementaciones a los que se aplica la regla:
- Para crear una regla que se aplique a todos los grupos de implementaciones asociados a la aplicación seleccionada, elija Any deployment group (Cualquier grupo de implementaciones).

- Para crear una regla que se aplique únicamente a uno de los grupos de implementaciones asociados a la aplicación seleccionada, seleccione Specific deployment group(s) (Grupos de implementaciones específicos) y, a continuación, elija el nombre del grupo de implementaciones en la lista.
8. Revise la configuración de las reglas para asegurarse de que se ajusta a los requisitos de monitorización de eventos.
  9. En el área Targets, seleccione Add target\*.
  10. En la lista Select target type, elija el tipo de destino que haya preparado para usarlo con esta regla y luego configure las opciones adicionales que requiera dicho tipo.
  11. Seleccione Configurar los detalles.
  12. En la página Configure rule details (Configurar detalles de la regla), escriba un nombre y la descripción de la regla, y, a continuación, elija la casilla State (Estado) para habilitar la regla ahora.
  13. Si está satisfecho con la regla, elija Create rule (Crear regla).

## Supervisar los despliegues con AWS CloudTrail

CodeDeploy está integrado con CloudTrail un servicio que captura las llamadas a la API realizadas por o en su nombre CodeDeploy en su AWS cuenta y entrega los archivos de registro a un bucket de Amazon S3 que especifique. CloudTrail captura las llamadas a la API desde la CodeDeploy consola, desde CodeDeploy los comandos a través de ella o directamente desde las CodeDeploy API. AWS CLI Con la información recopilada por CloudTrail, puede determinar a qué solicitud se realizó CodeDeploy, la dirección IP de origen desde la que se realizó la solicitud, quién la hizo, cuándo se realizó, etc. Para obtener más información CloudTrail, incluido cómo configurarla y habilitarla, consulte la [Guía AWS CloudTrail del usuario](#).

## CodeDeploy información en CloudTrail

Cuando se habilita el CloudTrail registro en su AWS cuenta, las llamadas a la API realizadas a CodeDeploy las acciones se registran en los archivos de registro. CodeDeploy los registros se escriben junto con otros registros AWS de servicio en un archivo de registro. CloudTrail determina cuándo crear y escribir en un nuevo archivo en función del período de tiempo y el tamaño del archivo.

Todas las CodeDeploy acciones se registran y documentan en la Referencia de la [línea de AWS CodeDeploy comandos y la Referencia](#) de la [AWS CodeDeploy API](#). Por ejemplo, las llamadas

para crear despliegues, eliminar aplicaciones y registrar las revisiones de las aplicaciones generan entradas en los archivos de CloudTrail registro.

Cada entrada de registro contiene información sobre quién generó la solicitud. La información de identidad del usuario en el registro le ayuda a determinar si la solicitud se realizó con credenciales raíz o de usuario, con credenciales de seguridad temporales para un rol o usuario federado, o por otro AWS servicio. [Para obtener más información, consulta el campo UserIdentity en la referencia del evento. CloudTrail](#)

Puede almacenar los archivos de registro en su bucket durante todo el tiempo que desee, pero también puede definir reglas de ciclo de vida de Amazon S3 para archivar o eliminar archivos de registro automáticamente. De forma predeterminada, se usa el cifrado del servidor (SSE) de Amazon S3 para cifrar los archivos de registro.

Puede hacer que se CloudTrail publiquen notificaciones de Amazon SNS cuando se entreguen nuevos archivos de registro. Para obtener más información, consulte [Configuración de las notificaciones de Amazon SNS](#) para. CloudTrail

También puede agregar archivos de CodeDeploy registro de varias AWS regiones y AWS cuentas en un único bucket de Amazon S3. Para obtener más información, consulte [Recibir archivos de CloudTrail registro de varias regiones](#).

## Descripción de las entradas de los archivos de CodeDeploy registro

CloudTrail Los archivos de registro pueden contener una o más entradas de registro, donde cada entrada se compone de varios eventos con formato JSON. Una entrada de registro representa una única solicitud de cualquier origen e incluye información sobre la acción solicitada, los parámetros, la fecha y la hora de la acción, etcétera. Las entradas de registro no se muestran en ningún orden concreto. Es decir, no representan un rastro de la pila ordenada de las llamadas a API públicas.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que muestra la acción de CodeDeploy crear un grupo de despliegues:

```
{
 "Records": [{
 "eventVersion": "1.02",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAI44QH8DHBEXAMPLE:203.0.113.11",
 "arn": "arn:aws:sts::123456789012:assumed-role/example-role/203.0.113.11",
 "accountId": "123456789012",
```

```
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2014-11-27T03:57:36Z"
 },
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAI44QH8DHBEXAMPLE",
 "arn": "arn:aws:iam::123456789012:role/example-role",
 "accountId": "123456789012",
 "userName": "example-role"
 }
},
"eventTime": "2014-11-27T03:57:36Z",
"eventSource": "codedeploy.amazonaws.com",
"eventName": "CreateDeploymentGroup",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.11",
"userAgent": "example-user-agent-string",
"requestParameters": {
 "applicationName": "ExampleApplication",
 "serviceRoleArn": "arn:aws:iam::123456789012:role/example-instance-group-role",
 "deploymentGroupName": "ExampleDeploymentGroup",
 "ec2TagFilters": [{
 "value": "CodeDeployDemo",
 "type": "KEY_AND_VALUE",
 "key": "Name"
 }],
 "deploymentConfigName": "CodeDeployDefault.HalfAtATime"
},
"responseElements": {
 "deploymentGroupId": "7d64e680-e6f4-4c07-b10a-9e117EXAMPLE"
},
"requestID": "86168559-75e9-11e4-8cf8-75d18EXAMPLE",
"eventID": "832b82d5-d474-44e8-a51d-093ccEXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
},
... additional entries ...
]
```

# Monitorización de implementaciones con notificaciones de eventos de Amazon SNS

Puede añadir activadores a un grupo de CodeDeploy implementaciones para recibir notificaciones sobre eventos relacionados con las implementaciones o instancias de ese grupo de implementaciones. Estas notificaciones se envían a los destinatarios que están suscritos a un tema de Amazon SNS que forma parte de la acción del activador.

Puede recibir notificaciones de CodeDeploy eventos en mensajes SMS o mensajes de correo electrónico. También puede utilizar de otras formas los datos JSON que se crean cuando se produce un evento determinado, por ejemplo, enviando mensajes a colas de Amazon SQS o invocando una función en AWS Lambda. Para conocer la estructura de los datos JSON proporcionados para los disparadores de implementación e instancia, consulte [Formatos de datos JSON para los CodeDeploy activadores](#).

Puede optar por utilizar disparadores para recibir notificaciones si:

- Es desarrollador y necesita saber cuándo se detiene o cuándo se produce un error en una implementación para que pueda solucionar los problemas.
- Es administrador del sistema y necesita saber en cuántas instancias se producen errores para poder monitorizar el estado de la flota de Amazon EC2.
- Usted es un administrador que desea llevar un registro at-a-glance de los eventos de implementación e instancia, lo que puede obtener mediante reglas de filtrado que redirigen los distintos tipos de notificaciones a las carpetas de su cliente de correo electrónico de escritorio.

Puede crear hasta 10 activadores para cada grupo de CodeDeploy implementación, para cualquiera de los siguientes tipos de eventos.

| Eventos de implementación                                                                                                                                                         | Eventos de instancia                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Success</li> <li>• Failure</li> <li>• Started</li> <li>• Stopped (Detenido)</li> <li>• Reversión</li> <li>• Ready<sup>1</sup></li> </ul> | <ul style="list-style-type: none"> <li>• Success</li> <li>• Failure</li> <li>• Started</li> <li>• Ready<sup>1</sup></li> <li>• Todos los eventos de instancia</li> </ul> |

## Eventos de implementación

## Eventos de instancia

- Todos los eventos de implementación

<sup>1</sup>Se aplica únicamente a las implementaciones azul/verde. Indica que la última revisión de la aplicación se ha instalado en las instancias de un entorno de sustitución y que el tráfico del entorno original se puede redirigir ahora tras un balanceador de carga. Para más información, consulte [Trabajar con despliegues en CodeDeploy](#).

## Temas

- [Conceder permisos de Amazon SNS a un rol de servicio CodeDeploy](#)
- [Cree un desencadenante para un evento CodeDeploy](#)
- [Edite un activador en un grupo CodeDeploy de implementación](#)
- [Elimine un activador de un grupo CodeDeploy de implementación](#)
- [Formatos de datos JSON para los CodeDeploy activadores](#)

## Conceder permisos de Amazon SNS a un rol de servicio CodeDeploy

Antes de que sus activadores puedan generar notificaciones, debe concederse permiso al rol de servicio que utilice en sus CodeDeploy operaciones para acceder a los recursos de Amazon SNS.

Para conceder permisos de Amazon SNS a un rol de servicio

1. [Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. En el panel de navegación de la consola de IAM, elija Roles.
3. Elija el nombre del rol de servicio que usará en las operaciones de AWS CodeDeploy .
4. En la pestaña Permissions, en el área Inline Políticas, elija Create Role Policy.

–o bien–

Si el botón Create Role Policy no está disponible, expanda el área Inline Políticas y después elija [click here](#).

5. En la página Set Permissions, elija Custom Policy y después Select.
6. En la página Review Policy (Revisar la política), en el campo Policy Name (Nombre de la política), introduzca un nombre que identifique esta política, como SNSPublish.

7. Pegue la siguiente información en el campo Policy Document:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "sns:Publish",
 "Resource": "*"
 }
]
}
```

8. Seleccione Apply Policy.

## Cree un desencadenante para un evento CodeDeploy

Puede crear un disparador que publique un tema de Amazon Simple Notification Service (Amazon SNS) para un evento de implementación o de instancia de AWS CodeDeploy . A continuación, cuando se produzca ese evento, todos los suscriptores del tema asociado recibirán notificaciones a través del punto de conexión especificado en el tema, como un mensaje SMS o un mensaje de correo electrónico. Amazon SNS ofrece varios métodos para suscribirse a temas.

Antes de crear un disparador, debe configurar el tema de Amazon SNS al que apuntará el disparador. Para obtener información, consulte [Creación de un tema](#). Cuando cree un tema, recomendamos que le asigne un nombre que identifique su finalidad, con un formato como Topic-group-us-west-3-deploy-fail o Topic-group-project-2-instance-stop.

También debes conceder permisos de Amazon SNS a un rol de CodeDeploy servicio antes de poder enviar las notificaciones para tu activador. Para obtener más información, consulte [Conceder permisos de Amazon SNS a un rol de servicio CodeDeploy](#) .

Una vez que haya creado el tema, puede añadir suscriptores. Para obtener información sobre cómo crear, administrar y suscribirse a los temas, consulte [¿Qué es Amazon Simple Notification Service?](#).

## Cree un disparador para enviar notificaciones de CodeDeploy eventos (consola)

Puedes usar la CodeDeploy consola para crear activadores de un CodeDeploy evento. Al finalizar el proceso de configuración, se envía un mensaje de notificación de prueba para garantizar que tanto los permisos como los detalles de los desencadenadores están configurados correctamente.

## Para crear un desencadenante para un CodeDeploy evento

1. En el AWS Management Console, abre la AWS CodeDeploy consola.
2. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

3. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
4. En la página Applications (Aplicaciones), elija el nombre de la aplicación asociada al grupo de implementaciones en el que desee añadir un disparador.
5. En la página Application details (Detalles de la aplicación), elija el grupo de implementaciones en el que desee añadir un disparador.
6. Elija Editar.
7. Expanda Advanced (Opciones avanzadas) (opcional).
8. En el área Triggers (Disparadores), elija Create trigger (Crear disparador).
9. En el panel Create deployment trigger (Crear disparador de implementaciones), realice lo siguiente:
  - a. En Trigger name (Nombre del disparador), introduzca un nombre para el disparador que facilite la identificación de su finalidad. Recomendamos un formato como `Trigger-group-us-west-3-deploy-fail` o `Trigger-group-eu-central-instance-stop`.
  - b. En Eventos, elija los tipos de evento que desee que disparen el envío de notificaciones por parte del tema de Amazon SNS.
  - c. En Temas de Amazon SNS, elija el nombre del tema que ha creado para enviar notificaciones para este disparador.
  - d. Seleccione Crear disparador. CodeDeploy envía una notificación de prueba para confirmar que ha configurado correctamente el acceso entre CodeDeploy y el tema Amazon SNS. En función del tipo de punto de enlace que haya seleccionado para el tema y de si se ha suscrito al tema, recibe un mensaje SMS o de correo electrónico de confirmación.
10. Elija Guardar cambios.



## Crear un disparador para enviar notificaciones de CodeDeploy eventos (CLI)

Puede utilizar la CLI para incluir disparadores al crear un grupo de implementaciones, o bien puede añadir disparadores a un grupo de implementaciones existente.

Para crear un disparador de envío de notificaciones para un nuevo grupo de implementaciones

Cree un archivo JSON para configurar el grupo de implementación y, a continuación, ejecute el [create-deployment-group](#) comando mediante la `--cli-input-json` opción.

La forma más sencilla de crear el archivo JSON es utilizar la opción `--generate-cli-skeleton` para obtener una copia del formato JSON y, a continuación, proporcionar los valores necesarios en un editor de texto sin formato.

1. Ejecute el siguiente comando y, a continuación, copie los resultados en un editor de texto sin formato.

```
aws deploy create-deployment-group --generate-cli-skeleton
```

2. Agrega el nombre de una CodeDeploy aplicación existente a la salida:

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentGroupName": "",
 "deploymentConfigName": "",
 "ec2TagFilters": [
 {
 "Key": "",
 "Value": "",
 "Type": ""
 }
],
 "onPremisesInstanceTagFilters": [
 {
 "Key": "",
 "Value": "",
 "Type": ""
 }
],
 "autoScalingGroups": [
 ""
],
}
```

```
"serviceRoleArn": "",
"triggerConfigurations": [
 {
 "triggerName": "",
 "triggerTargetArn": "",
 "triggerEvents": [
 ""
]
 }
]
```

3. Proporcione los valores de los parámetros que desea configurar.

Al utilizar el [create-deployment-group](#) comando, debe proporcionar, como mínimo, valores para los siguientes parámetros:

- `applicationName`: el nombre de una aplicación ya creada en su cuenta.
- `deploymentGroupName`: un nombre para el grupo de implementaciones que va a crear.
- `serviceRoleArn`: El ARN de un rol de servicio existente configurado CodeDeploy en su cuenta. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).

En la sección `triggerConfigurations`, proporcione valores para los siguientes parámetros:

- `triggerName`: el nombre que desea asignar al disparador para que pueda identificarlo fácilmente. Recomendamos un formato como `Trigger-group-us-west-3-deploy-fail` o `Trigger-group-eu-central-instance-stop`.
- `triggerTargetArn`: el ARN del tema de Amazon SNS que ha creado para asociar al disparador, en este formato: `arn:aws:sns:us-east-2:444455556666:NewTestTopic`.
- `triggerEvents`: el tipo de evento o eventos para los que desea disparar notificaciones. Puede especificar uno o varios tipos de eventos, separando con comas los distintos nombres de tipo de evento (por ejemplo, `"triggerEvents": ["DeploymentSuccess", "DeploymentFailure", "InstanceFailure"]`). Cuando añada más de un tipo de evento, las notificaciones de todos esos tipos se envían al tema especificado, en lugar de enviarse a un tema diferente para cada una de ellas. Puede elegir entre los siguientes tipos de evento:
  - `DeploymentStart`

- DeploymentSuccess
- DeploymentFailure
- DeploymentStop
- DeploymentRollback
- DeploymentReady (Solo se aplica a las instancias de reemplazo en una implementación azul/verde)
- InstanceStart
- InstanceSuccess
- InstanceFailure
- InstanceReady (Solo se aplica a las instancias de reemplazo en una implementación azul/verde)

El siguiente ejemplo de configuración crea un grupo de implementaciones denominado dep-group-ghi-789-2 para una aplicación llamada TestApp-us-east-2 y un disparador que provoca el envío de notificaciones siempre que una implementación comience o se realice correctamente o cuando se produzca un error en ella:

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "deploymentGroupName": "dep-group-ghi-789-2",
 "ec2TagFilters": [
 {
 "Key": "Name",
 "Value": "Project-ABC",
 "Type": "KEY_AND_VALUE"
 }
],
 "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
 "triggerConfigurations": [
 {
 "triggerName": "Trigger-group-us-east-2",
 "triggerTargetArn": "arn:aws:sns:us-east-2:444455556666:us-east-
deployments",
 "triggerEvents": [
 "DeploymentStart",
 "DeploymentSuccess",
 "DeploymentFailure"
]
 }
]
}
```

```
]
 }
]
}
```

4. Guarde las actualizaciones como un archivo JSON y, a continuación, llame a dicho archivo con la opción `--cli-input-json` cuando ejecute el comando `create-deployment-group`:

**⚠ Important**

Asegúrese de incluir `file://` antes del nombre de archivo. Es obligatorio en este comando.

```
aws deploy create-deployment-group --cli-input-json file://filename.json
```

Al finalizar el proceso de creación, recibe un mensaje de notificación de prueba que indica que tanto los permisos como los detalles de los disparadores están configurados correctamente.

Para crear un disparador de envío de notificaciones para un grupo de implementaciones existente

Para usar el AWS CLI para añadir activadores de CodeDeploy eventos a un grupo de despliegues existente, crea un archivo JSON para actualizar el grupo de despliegues y, a continuación, ejecuta el [update-deployment-group](#) comando mediante la `--cli-input-json` opción.

La forma más sencilla de crear el archivo JSON es ejecutar el comando `get-deployment-group` para obtener una copia de la configuración del grupo de implementaciones, en formato JSON y, a continuación, actualizar los valores de los parámetros en un editor de texto sin formato.

1. Ejecute el siguiente comando y, a continuación, copie los resultados en un editor de texto sin formato.

```
aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
```

2. Elimine lo siguiente del resultado:

- Al principio del resultado, elimine `{ "deploymentGroupInfo":.`
- Al final del resultado, elimine `}`.

- Elimine la fila que contiene `deploymentGroupId`.
- Elimine la fila que contiene `deploymentGroupName`.

El contenido de su archivo de texto debe tener ahora un aspecto similar al siguiente:

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "autoScalingGroups": [],
 "ec2TagFilters": [
 {
 "Type": "KEY_AND_VALUE",
 "Value": "Project-ABC",
 "Key": "Name"
 }
],
 "triggerConfigurations": [],
 "serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
 "onPremisesInstanceTagFilters": []
}
```


3. En la sección `triggerConfigurations`, añade datos para los parámetros `triggerEvents`, `triggerTargetArn` y `triggerName`. Para obtener información sobre los parámetros de configuración de los activadores, consulte [TriggerConfig](#).

El contenido de su archivo de texto debe tener ahora un aspecto similar al siguiente. Este código provoca el envío de notificaciones siempre que una implementación comience o se realice correctamente o cuando se produzca un error en ella.

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "autoScalingGroups": [],
 "ec2TagFilters": [
 {
 "Type": "KEY_AND_VALUE",
 "Value": "Project-ABC",
 "Key": "Name"
 }
],
 "triggerConfigurations": [
```

```
{
 "triggerEvents": [
 "DeploymentStart",
 "DeploymentSuccess",
 "DeploymentFailure"
],
 "triggerTargetArn": "arn:aws:sns:us-east-2:444455556666:us-east-
deployments",
 "triggerName": "Trigger-group-us-east-2"
},
"serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
"onPremisesInstanceTagFilters": []
}
```

4. Guarde las actualizaciones como un archivo JSON y, a continuación, ejecute el [update-deployment-group](#) comando mediante la `--cli-input-json` opción. Asegúrese de incluir la opción `--current-deployment-group-name` y especifique el nombre de su archivo JSON en *filename*:

 Important

Asegúrese de incluir `file://` antes del nombre de archivo. Es obligatorio en este comando.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-
group-name --cli-input-json file://filename.json
```

Al finalizar el proceso de creación, recibe un mensaje de notificación de prueba que indica que tanto los permisos como los detalles de los disparadores están configurados correctamente.

## Edite un activador en un grupo CodeDeploy de implementación

Si sus necesidades de notificación cambian, puede modificar el desencadenador en lugar de crear uno nuevo.

## Modificar un CodeDeploy disparador (CLI)

Para usarlo AWS CLI para cambiar los detalles de los activadores de CodeDeploy los eventos al actualizar un grupo de implementaciones, cree un archivo JSON para definir los cambios en las propiedades del grupo de implementación y, a continuación, ejecute el [update-deployment-group](#) comando con la `--cli-input-json` opción.

La forma más sencilla de crear el archivo JSON es ejecutar el comando `get-deployment-group` para obtener los detalles del grupo de implementaciones actual en formato JSON y, a continuación, editar los valores necesarios en un editor de texto sin formato.

1. Ejecute el siguiente comando, sustituyendo *application* y *deployment-group* por los nombres de su aplicación y grupo de implementaciones:

```
aws deploy get-deployment-group --application-name application --deployment-group-name deployment-group
```


2. Copie los resultados del comando en un editor de texto sin formato y, a continuación, elimine lo siguiente:
  - Al principio del resultado, elimine `{ "deploymentGroupInfo":.`
  - Al final del resultado, elimine `}`.
  - Elimine la fila que contiene `deploymentGroupId`.
  - Elimine la fila que contiene `deploymentGroupName`.

El contenido de su archivo de texto debe tener ahora un aspecto similar al siguiente:

```
{
 "applicationName": "TestApp-us-east-2",
 "deploymentConfigName": "CodeDeployDefault.OneAtATime",
 "autoScalingGroups": [],
 "ec2TagFilters": [
 {
 "Type": "KEY_AND_VALUE",
 "Value": "East-1-Instances",
 "Key": "Name"
 }
],
 "triggerConfigurations": [
 {
```

```
 "triggerEvents": [
 "DeploymentStart",
 "DeploymentSuccess",
 "DeploymentFailure",
 "DeploymentStop"
],
 "triggerTargetArn": "arn:aws:sns:us-east-2:111222333444:Trigger-group-
us-east-2",
 "triggerName": "Trigger-group-us-east-2"
 }
],
"serviceRoleArn": "arn:aws:iam::444455556666:role/AnyCompany-service-role",
"onPremisesInstanceTagFilters": []
}
```

3. Cambie los parámetros según sea necesario. Para obtener información sobre los parámetros de configuración del disparador, consulte [TriggerConfig](#).
4. Guarde las actualizaciones como un archivo JSON y, a continuación, ejecute el [update-deployment-group](#) comando mediante la `--cli-input-json` opción. Asegúrese de incluir la opción `--current-deployment-group-name` y especifique el nombre de su archivo JSON en *filename*:

 Important

Asegúrese de incluir `file://` antes del nombre de archivo. Es obligatorio en este comando.

```
aws deploy update-deployment-group --current-deployment-group-name deployment-
group-name --cli-input-json file://filename.json
```

Al finalizar el proceso de creación, recibe un mensaje de notificación de prueba que indica que tanto los permisos como los detalles de los disparadores están configurados correctamente.

## Elimine un activador de un grupo CodeDeploy de implementación

Dada la limitación de 10 desencadenadores por grupo de implementaciones, es posible que desee eliminar desencadenadores si ya no se utilizan. No puede deshacer la eliminación de un disparador, pero puede volver a crearlos.



## Eliminación de un disparador de un grupo de implementación (consola)

1. Inicie sesión en la CodeDeploy consola AWS Management Console y ábrala en <https://console.aws.amazon.com/codedeploy>.

### Note

Inicie sesión con el mismo usuario que configuró en [Empezar con CodeDeploy](#).

2. En el panel de navegación, expanda Implementar y, después, elija Aplicaciones.
3. En la página Applications (Aplicaciones), elija el nombre de la aplicación asociada al grupo de implementaciones en el que desee eliminar un disparador.
4. En la página Application details (Detalles de la aplicación), elija el grupo de implementaciones en el que desee eliminar un disparador.
5. Elija Editar.
6. Expanda Advanced (Opciones avanzadas) (opcional).
7. En el área Triggers (Disparadores), elija el disparador que desee eliminar y, a continuación, elija Delete trigger (Eliminar disparador).
8. Elija Guardar cambios.

## Eliminación de un disparador de un grupo de implementación (CLI)

Para usar la CLI para eliminar un disparador, llame al [update-deployment-group](#) comando, con los parámetros de configuración del disparador vacíos, especificando:

- El nombre de la aplicación asociada al grupo de implementaciones. Para ver una lista de nombres de aplicaciones, llame al comando [list-applications](#).
- El nombre del grupo de implementación asociado a la aplicación. Para ver una lista de los nombres de los grupos de implementación, [list-deployment-group](#) ejecute el comando.

Por ejemplo:

```
aws deploy update-deployment-group --application-name application-name --current-deployment-group-name deployment-group-name --trigger-configurations
```

## Formatos de datos JSON para los CodeDeploy activadores

Puede utilizar la salida JSON que se crea cuando se activa un disparador para una implementación o una instancia en un flujo de trabajo de notificación personalizado; por ejemplo, enviar mensajes a colas de Amazon SQS o invocar una función de AWS Lambda.

### Note

Esta guía no trata sobre el modo de configurar las notificaciones en JSON. Para obtener información sobre cómo utilizar Amazon SNS para enviar mensajes a las colas de Amazon SQS, consulte [Envío de mensajes de Amazon SNS a colas de Amazon SQS](#). Para obtener información sobre cómo usar Amazon SNS para invocar una función de Lambda, consulte [Invocación de funciones de Lambda mediante notificaciones de Amazon SNS](#).

Los siguientes ejemplos muestran la estructura de la salida JSON disponible con los CodeDeploy activadores.

### Salida JSON de ejemplo para disparadores basados en instancias

```
{
 "region": "us-east-2",
 "accountId": "111222333444",
 "eventTriggerName": "trigger-group-us-east-instance-succeeded",
 "deploymentId": "d-75I7MBT7C",
 "instanceId": "arn:aws:ec2:us-east-2:444455556666:instance/i-496589f7",
 "lastUpdatedAt": "1446744207.564",
 "instanceStatus": "Succeeded",
 "lifecycleEvents": [
 {
 "LifecycleEvent": "ApplicationStop",
 "LifecycleEventStatus": "Succeeded",
 "StartTime": "1446744188.595",
 "EndTime": "1446744188.711"
 },
 {
 "LifecycleEvent": "BeforeInstall",
 "LifecycleEventStatus": "Succeeded",
 "StartTime": "1446744189.827",
 "EndTime": "1446744190.402"
 }
]
}
```

```
//More lifecycle events might be listed here
]
}
```

## Salida JSON de ejemplo para disparadores basados en implementaciones

```
{
 "region": "us-west-1",
 "accountId": "111222333444",
 "eventTriggerName": "Trigger-group-us-west-3-deploy-failed",
 "applicationName": "ProductionApp-us-west-3",
 "deploymentId": "d-75I7MBT7C",
 "deploymentGroupName": "dep-group-def-456",
 "createTime": "1446744188.595",
 "completeTime": "1446744190.402",
 "deploymentOverview": {
 "Failed": "10",
 "InProgress": "0",
 "Pending": "0",
 "Skipped": "0",
 "Succeeded": "0"
 },
 "status": "Failed",
 "errorInformation": {
 "ErrorCode": "IAM_ROLE_MISSING",
 "ErrorMessage": "IAM Role is missing for deployment group: dep-group-def-456"
 }
}
```

# Seguridad en AWS CodeDeploy

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores independientes prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los [programas de conformidad de AWS](#). Para obtener más información sobre los programas de conformidad aplicables AWS CodeDeploy, consulte los [AWS servicios incluidos en el ámbito de aplicación por programa de conformidad](#).
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza CodeDeploy. Los siguientes temas muestran cómo configurarlo CodeDeploy para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus CodeDeploy recursos.

## Temas

- [Protección de datos en AWS CodeDeploy](#)
- [Administración de identidades y accesos para AWS CodeDeploy](#)
- [Inicio de sesión y supervisión CodeDeploy](#)
- [Validación de conformidad para AWS CodeDeploy](#)
- [Resiliencia en AWS CodeDeploy](#)
- [Seguridad de la infraestructura en AWS CodeDeploy](#)

# Protección de datos en AWS CodeDeploy

El modelo de [responsabilidad AWS compartida modelo](#) se aplica a la protección de datos en AWS CodeDeploy. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad 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 .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados 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 por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final 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 introducir nunca 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 trabaja CodeDeploy o Servicios de AWS utiliza la consola, la API o los SDK. AWS CLI AWS Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear 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.

## Privacidad del tráfico entre redes

CodeDeploy es un servicio de implementación totalmente gestionado que admite instancias EC2, funciones Lambda, Amazon ECS y servidores locales. En el caso de las instancias EC2 y los servidores locales, un agente basado en el host se comunica mediante TLS. CodeDeploy

Actualmente, la comunicación entre el agente y el servicio requiere una conexión a Internet saliente para que el agente pueda comunicarse con los puntos de enlace públicos CodeDeploy y del servicio Amazon S3. En una nube virtual privada, esto se puede lograr con una gateway de Internet, una conexión de VPN de sitio a sitio a su red corporativa o una conexión directa.

El CodeDeploy agente admite proxies HTTP.

Los puntos de enlace de Amazon VPC, con tecnología de AWS PrivateLink, están disponibles CodeDeploy en determinadas regiones. Para obtener más detalles, consulte [Úselo CodeDeploy con Amazon Virtual Private Cloud](#).

### Note

El CodeDeploy agente solo es necesario si lo implementa en una plataforma informática Amazon EC2/on-premise. El agente no es necesario para las implementaciones que utilizan Amazon ECS o la plataforma de AWS Lambda cómputo.

## Cifrado en reposo

El código de cliente no se almacena en CodeDeploy. Como servicio de despliegue, envía comandos al CodeDeploy agente que CodeDeploy se ejecuta en instancias EC2 o servidores locales. A continuación, el CodeDeploy agente ejecuta los comandos mediante TLS. Los datos del modelo de servicio para las implementaciones, la configuración de las implementaciones, los grupos de implementación, las aplicaciones y las revisiones de las aplicaciones se almacenan en Amazon DynamoDB y se cifran en reposo Clave propiedad de AWS mediante una, propiedad de y gestionada por. CodeDeploy Para obtener más información, consulte [Clave propiedad de AWS](#).

## Cifrado en tránsito

El CodeDeploy agente inicia todas las comunicaciones a través del puerto 443. CodeDeploy El agente sondea CodeDeploy y escucha un comando. El CodeDeploy agente es de código abierto. Todas service-to-service las client-to-service comunicaciones se cifran en tránsito mediante TLS. Esto protege los datos de los clientes en tránsito entre CodeDeploy y otros servicios, como Amazon S3.

## Administración de claves de cifrado

No hay claves de cifrado para que las administre. Los datos del modelo de CodeDeploy servicio se cifran mediante una Clave propiedad de AWS, propiedad de y gestionados por CodeDeploy. Para obtener más información, consulte [Clave propiedad de AWS](#).

## Administración de identidades y accesos para AWS CodeDeploy

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. CodeDeploy La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

### Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración de acceso mediante políticas](#)
- [¿Cómo AWS CodeDeploy funciona con IAM](#)
- [AWS políticas gestionadas \(predefinidas\) para CodeDeploy](#)
- [CodeDeploy actualizaciones de las políticas AWS gestionadas](#)
- [AWS CodeDeploy ejemplos de políticas basadas en identidad de](#)
- [Solución de problemas de identidades de AWS CodeDeploy y accesos](#)
- [Referencia de permisos de CodeDeploy](#)
- [Prevención de la sustitución confusa entre servicios](#)

## Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice. CodeDeploy

Usuario del servicio: si utiliza el CodeDeploy servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más CodeDeploy funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una función en CodeDeploy, consulte [Solución de problemas de identidades de AWS CodeDeploy y accesos](#).

Administrador de servicios: si está a cargo de CodeDeploy los recursos de su empresa, probablemente tenga acceso total a ellos CodeDeploy. Su trabajo consiste en determinar a qué CodeDeploy funciones y recursos deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar la IAM CodeDeploy, consulte [¿Cómo AWS CodeDeploy funciona con IAM](#).

Administrador de IAM: si es administrador de IAM, puede que le interese obtener más información sobre cómo redactar políticas para administrar el acceso. CodeDeploy Para ver ejemplos de políticas CodeDeploy basadas en la identidad que puede usar en IAM, consulte [AWS CodeDeploy ejemplos de políticas basadas en identidad de](#)

## Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.



Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte [Cómo iniciar sesión Cuenta de AWS en su](#) Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte [Firmar las solicitudes de la AWS API](#) en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte [Autenticación multifactor](#) en la Guía del usuario de AWS IAM Identity Center y [Uso de la autenticación multifactor \(MFA\) en AWS](#) en la Guía del usuario de IAM.

## Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

## Usuarios y grupos

Un [usuario de IAM](#) es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte [Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la

vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte [Cuándo crear un usuario de IAM \(en lugar de un rol\)](#) en la Guía del usuario de IAM.

## Roles de IAM

Un [rol de IAM](#) es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console [cambiando](#) de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte [Uso de roles de IAM](#) en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- **Acceso de usuario federado:** para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte [Creación de un rol para un proveedor de identidades de terceros](#) en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center .
- **Permisos de usuario de IAM temporales:** un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- **Acceso entre cuentas:** puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para

el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
- Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Reenviar sesiones de acceso](#).
- Rol de servicio: un rol de servicio es un [rol de IAM](#) que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.
- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar una AWS función a una instancia EC2 y ponerla a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte [Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2](#) en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte [Cuándo crear un rol de IAM \(en lugar de un usuario\)](#) en la Guía del usuario de IAM.

## Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte [Información general de políticas JSON](#) en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

## Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una

política administrada o una política insertada, consulte [Elegir entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

## Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- **Límites de permisos:** un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- **Políticas de control de servicios (SCP):** las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte [Funcionamiento de las SCP](#) en la Guía del usuario de AWS Organizations .
- **Políticas de sesión:** las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

## Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando

se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

## ¿Cómo AWS CodeDeploy funciona con IAM

Antes de utilizar IAM para gestionar el acceso CodeDeploy, debe saber qué funciones de IAM están disponibles para su uso. CodeDeploy Para obtener más información, consulte [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

### Temas

- [Políticas de CodeDeploy basadas en identidades](#)
- [CodeDeploy políticas basadas en recursos](#)
- [Autorización basada en etiquetas de CodeDeploy](#)
- [CodeDeploy Funciones de IAM](#)

### Políticas de CodeDeploy basadas en identidades

Con las políticas basadas en identidades de IAM, puede especificar las acciones permitidas o denegadas, así como los recursos y las condiciones en las que se permiten o deniegan las acciones. CodeDeploy admite acciones, recursos y claves de condición. Para obtener información sobre los elementos que utiliza en una política JSON, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la Guía del usuario de IAM.

### Acciones

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Las acciones políticas utilizadas CodeDeploy utilizan el `codedeploy:` prefijo antes de la acción. Por ejemplo, el permiso `codedeploy:GetApplication` concede al usuario permiso para realizar la operación `GetApplication`. Las declaraciones de política deben incluir un `NotAction` elemento

Action o. CodeDeploy define su propio conjunto de acciones que describen las tareas que puede realizar con este servicio.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": [
 "codedeploy:action1",
 "codedeploy:action2"
```

Puede utilizar caracteres comodín para especificar varias acciones (\*). Por ejemplo, incluya la acción siguiente para especificar todas las acciones que comiencen por la palabra Describe:

```
"Action": "ec2:Describe*"
```

Para obtener una lista de CodeDeploy acciones, consulte [las acciones definidas por AWS CodeDeploy](#) en la Guía del usuario de IAM.

Para ver una tabla en la que se enumeran todas las acciones de la CodeDeploy API y los recursos a los que se aplican, consulte [Referencia de permisos de CodeDeploy](#).

## Recursos

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento Resource de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento Resource o NotResource. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (\*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Por ejemplo, puede indicar un grupo de despliegue (*myDeploymentGroup*) en su declaración mediante su ARN de la siguiente manera:

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:myApplication/myDeploymentGroup"
```

También puede especificar todos los grupos de implementación que pertenezcan a una cuenta mediante el carácter comodín (\*) del modo siguiente:

```
"Resource": "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*"
```

Para especificar todos los recursos, o si una acción de API no admite ARN, utilice el carácter comodín (\*) en el elemento Resource de la siguiente manera:

```
"Resource": "*"
```

Algunas acciones de la CodeDeploy API aceptan varios recursos (por ejemplo, `BatchGetDeploymentGroups`). Para especificar varios recursos en una única instrucción, separe sus ARN con comas, tal y como se indica a continuación:

```
"Resource": ["arn1", "arn2"]
```

CodeDeploy proporciona un conjunto de operaciones para trabajar con los CodeDeploy recursos. Para ver la lista de las operaciones disponibles, consulte [Referencia de permisos de CodeDeploy](#).


Para obtener una lista de los tipos de CodeDeploy recursos y sus ARN, consulte [los recursos definidos por AWS CodeDeploy](#) en la Guía del usuario de IAM. Para obtener información sobre las acciones en las que puede especificar el ARN de cada recurso, consulte [Acciones definidas por AWS CodeDeploy](#).

## CodeDeploy recursos y operaciones

En CodeDeploy, el recurso principal es un grupo de despliegues. En las políticas se emplean nombres de recurso de Amazon (ARN) para identificar los recursos a los que se aplican las políticas. CodeDeploy admite otros recursos que se pueden usar con los grupos de implementación, incluidas las aplicaciones, las configuraciones de implementación y las instancias. Estos elementos se denominan subrecursos. Estos recursos y subrecursos tienen ARN únicos asociados a ellos. Para obtener más información, consulte [Nombres de recurso de Amazon \(ARN\)](#) en la Referencia general de Amazon Web Services.



| Tipo de recurso                                                                                   | Formato de ARN                                                                                                |
|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| Grupo de implementación                                                                           | arn:aws:codedeploy: <i>region:account-id</i> :deploymentgroup: <i>application-name /deployment-group-name</i> |
| Aplicación                                                                                        | arn:aws:codedeploy: <i>region:account-id</i> :application: <i>application-name</i>                            |
| Configuración de implementación                                                                   | arn:aws:codedeploy: <i>region:account-id</i> :deploymentconfig: <i>deployment-configuration-name</i>          |
| instancia                                                                                         | arn:aws:codedeploy: <i>region:account-id</i> :instance / <i>instance-ID</i>                                   |
| Todos los CodeDeploy recursos                                                                     | arn:aws:codedeploy:*                                                                                          |
| Todos CodeDeploy y los recursos que pertenecen a la cuenta especificada en la región especificada | arn:aws:codedeploy: <i>region:account-id</i> :*                                                               |

 Note

La mayoría de los AWS servicios utilizan dos puntos (:)) o una barra diagonal (/) como el mismo carácter en los ARN. Sin embargo, CodeDeploy utiliza una coincidencia exacta en las reglas y patrones de recursos. Asegúrese de utilizar los caracteres de ARN correctos cuando cree patrones de eventos para que coincidan con la sintaxis de ARN en el recurso.

## Claves de condición

CodeDeploy no proporciona ninguna clave de condición específica del servicio, pero admite el uso de algunas claves de condición globales. Para obtener más información, consulte [Claves de contexto de condición globales de AWS](#) en la Guía del usuario de IAM.

## Ejemplos

Para ver ejemplos de políticas CodeDeploy basadas en la identidad, consulte [AWS CodeDeploy ejemplos de políticas basadas en identidad de](#)

## CodeDeploy políticas basadas en recursos

CodeDeploy no admite políticas basadas en recursos. Para ver un ejemplo de una página detallada de políticas basadas en recursos, consulte [Uso](#) de políticas basadas en recursos para AWS Lambda

## Autorización basada en etiquetas de CodeDeploy

CodeDeploy no admite el etiquetado de los recursos ni el control del acceso en función de las etiquetas.

## CodeDeploy Funciones de IAM

Un [rol de IAM](#) es una entidad de tu AWS cuenta que tiene permisos específicos.

### Usar credenciales temporales con CodeDeploy

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen llamando a operaciones de AWS STS API como [AssumeRole](#) o [GetFederationToken](#).

CodeDeploy admite el uso de credenciales temporales.

### Roles vinculados al servicio

CodeDeploy no admite funciones vinculadas a servicios.

### Roles de servicio

Esta característica permite que un servicio asuma un [rol de servicio](#) en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su AWS cuenta y son propiedad de la cuenta. Esto significa que un usuario puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

CodeDeploy admite funciones de servicio.

## Elegir un rol de IAM en CodeDeploy

Al crear un recurso de grupo de despliegues en CodeDeploy, debe elegir un rol que le permita acceder CodeDeploy a Amazon EC2 en su nombre. Si ha creado anteriormente un rol de servicio o un rol vinculado a un servicio, le CodeDeploy proporciona una lista de roles entre los que puede elegir. Es importante seleccionar un rol que permita el acceso para iniciar y detener una instancia EC2.

## AWS políticas gestionadas (predefinidas) para CodeDeploy

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que son creadas y administradas por AWS. Estas políticas AWS gestionadas conceden permisos para casos de uso comunes, de modo que no tengas que investigar qué permisos son necesarios. Para obtener más información, consulte [Políticas administradas de AWS](#) en la Guía del usuario de IAM.

### Temas

- [Lista de políticas AWS administradas para CodeDeploy](#)
- [CodeDeploy políticas y notificaciones gestionadas](#)

## Lista de políticas AWS administradas para CodeDeploy

Las siguientes políticas AWS administradas, que puede adjuntar a los usuarios de su cuenta, son específicas de CodeDeploy:

- `AWSCodeDeployFullAccess`: concede acceso completo a CodeDeploy.

### Note

`AWSCodeDeployFullAccess` no proporciona permisos para las operaciones en otros servicios necesarios para implementar sus aplicaciones, como Amazon EC2 y Amazon S3, solo para las operaciones específicas de CodeDeploy.

- `AWSCodeDeployDeployerAccess`: otorga permiso para registrar e implementar revisiones.
- `AWSCodeDeployReadOnlyAccess`: concede acceso de solo lectura a CodeDeploy.

- **AWSCodeDeployRole:** Permite CodeDeploy :
  - Leer las etiquetas en sus instancias o identificar sus instancias de Amazon EC2 mediante nombres de grupo de Amazon EC2 Auto Scaling.
  - Leer, crear, actualizar y eliminar grupos de Amazon EC2 Auto Scaling, enlaces de ciclo de vida, políticas de escalado y funciones de grupo de calentamiento.
  - Publicar información para temas de Amazon SNS.
  - recuperar información sobre las CloudWatch alarmas de Amazon
  - Leer y actualizar los recursos del servicio Elastic Load Balancing.

La política contiene el siguiente código:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "autoscaling:CompleteLifecycleAction",
 "autoscaling>DeleteLifecycleHook",
 "autoscaling:DescribeAutoScalingGroups",
 "autoscaling:DescribeLifecycleHooks",
 "autoscaling:PutLifecycleHook",
 "autoscaling:RecordLifecycleActionHeartbeat",
 "autoscaling>CreateAutoScalingGroup",
 "autoscaling>CreateOrUpdateTags",
 "autoscaling:UpdateAutoScalingGroup",
 "autoscaling:EnableMetricsCollection",
 "autoscaling:DescribePolicies",
 "autoscaling:DescribeScheduledActions",
 "autoscaling:DescribeNotificationConfigurations",
 "autoscaling:SuspendProcesses",
 "autoscaling:ResumeProcesses",
 "autoscaling:AttachLoadBalancers",
 "autoscaling:AttachLoadBalancerTargetGroups",
 "autoscaling:PutScalingPolicy",
 "autoscaling:PutScheduledUpdateGroupAction",
 "autoscaling:PutNotificationConfiguration",
 "autoscaling:DescribeScalingActivities",
 "autoscaling>DeleteAutoScalingGroup",
 "autoscaling:PutWarmPool",
 "ec2:DescribeInstances",
```

```

 "ec2:DescribeInstanceStatus",
 "ec2:TerminateInstances",
 "tag:GetResources",
 "sns:Publish",
 "cloudwatch:DescribeAlarms",
 "cloudwatch:PutMetricAlarm",
 "elasticloadbalancing:DescribeLoadBalancers",
 "elasticloadbalancing:DescribeLoadBalancerAttributes",
 "elasticloadbalancing:DescribeInstanceHealth",
 "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
 "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
 "elasticloadbalancing:DescribeTargetGroups",
 "elasticloadbalancing:DescribeTargetGroupAttributes",
 "elasticloadbalancing:DescribeTargetHealth",
 "elasticloadbalancing:RegisterTargets",
 "elasticloadbalancing:DeregisterTargets"
],
 "Resource": "*"
}
]
}

```

- **AWSCodeDeployRoleForLambda**: otorga CodeDeploy permiso de acceso AWS Lambda y cualquier otro recurso necesario para una implementación.
- **AWSCodeDeployRoleForECS**: Concede CodeDeploy permiso para acceder a Amazon ECS y a cualquier otro recurso necesario para una implementación.
- **AWSCodeDeployRoleForECSLimited**: Concede CodeDeploy permiso para acceder a Amazon ECS y a cualquier otro recurso necesario para una implementación, con las siguientes excepciones:
  - En la hooks sección del AppSpec archivo, solo se CodeDeployHook\_ pueden usar las funciones Lambda con nombres que comiencen por. Para obtener más información, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#).

- El acceso del bucket de S3 se limita a buckets de S3 con una etiqueta de registro, `UseWithCodeDeploy`, que tiene un valor de `true`. Para obtener más información, consulte [Etiquetado de objetos](#).
- `AmazonEC2RoleforAWSCodeDeployLimited`: otorga CodeDeploy permiso para obtener y enumerar objetos en un bucket de CodeDeploy Amazon S3. La política contiene el siguiente código:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion",
 "s3:ListBucket"
],
 "Resource": "arn:aws:s3::*/CodeDeploy/*"
 },
 {
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
 }
 }
 }
]
}
```

Los permisos para algunos aspectos del proceso de implementación se otorgan a otros dos tipos de roles que actúan en nombre de CodeDeploy:

- Un perfil de instancia de IAM es un rol de IAM que se vincula a las instancias de Amazon EC2. Este perfil incluye los permisos necesarios para acceder a los GitHub depósitos o repositorios de

Amazon S3 en los que se almacenan las aplicaciones. Para obtener más información, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#).

- Un rol de servicio es un rol de IAM que otorga permisos a un AWS servicio para que pueda acceder a los recursos. AWS Las políticas que se adjuntan a la función de servicio determinan a qué AWS recursos puede acceder el servicio y las acciones que puede realizar con esos recursos. Para CodeDeploy, un rol de servicio se usa para lo siguiente:
  - Para leer las etiquetas aplicadas a las instancias o los nombres de grupo de Amazon EC2 Auto Scaling asociados a las instancias. Esto le CodeDeploy permite identificar las instancias en las que puede implementar aplicaciones.
  - Para realizar operaciones en instancias, grupos de Amazon EC2 Auto Scaling y equilibradores de carga de Elastic Load Balancing.
  - Publicar información en los temas de Amazon SNS para que se puedan enviar notificaciones cuando se produzcan eventos de implementación o instancia específicos.
  - Recuperar información sobre CloudWatch las alarmas a fin de configurar la supervisión de las alarmas para las implementaciones.

Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).

También puede crear políticas de IAM personalizadas para conceder permisos para CodeDeploy acciones y recursos. Adjunta estas políticas personalizadas a los roles de IAM y, a continuación, asigna los roles a los usuarios o grupos que requieren los permisos.

## CodeDeploy políticas y notificaciones gestionadas

CodeDeploy admite las notificaciones, que pueden notificar a los usuarios los cambios importantes en las implementaciones. Las políticas gestionadas CodeDeploy incluyen declaraciones de políticas para la funcionalidad de notificación. Para obtener más información, consulte [¿Qué son las notificaciones?](#)

Permisos relacionados con las notificaciones en políticas administradas de acceso total

La política administrada `AWSCodeDeployFullAccess` incluye las siguientes instrucciones para permitir el acceso completo a las notificaciones. Los usuarios con esta política administrada aplicada también pueden crear y administrar temas de Amazon SNS para notificaciones, suscribirse y cancelar la suscripción de los usuarios a los temas, mostrar los temas que se pueden elegir como destinos para las reglas de notificación y mostrar los clientes de AWS Chatbot configurados para Slack.

```

{
 "Sid": "CodeStarNotificationsReadWriteAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:CreateNotificationRule",
 "codestar-notifications:DescribeNotificationRule",
 "codestar-notifications:UpdateNotificationRule",
 "codestar-notifications>DeleteNotificationRule",
 "codestar-notifications:Subscribe",
 "codestar-notifications:Unsubscribe"
],
 "Resource": "*",
 "Condition" : {
 "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codedeploy:*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListTargets",
 "codestar-notifications:ListTagsForResource",
 "codestar-notifications:ListEventTypes"
],
 "Resource": "*"
},
{
 "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
 "Effect": "Allow",
 "Action": [
 "sns:CreateTopic",
 "sns:SetTopicAttributes"
],
 "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
 "Sid": "SNSTopicListAccess",
 "Effect": "Allow",
 "Action": [
 "sns:ListTopics"
],

```



```

 "Resource": "*"
 },
 {
 "Sid": "CodeStarNotificationsChatbotAccess",
 "Effect": "Allow",
 "Action": [
 "chatbot:DescribeSlackChannelConfigurations",
 "chatbot:ListMicrosoftTeamsChannelConfigurations"
],
 "Resource": "*"
 }
}

```

## Permisos relacionados con las notificaciones en políticas administradas de solo lectura

La política administrada `AWSCodeDeployReadOnlyAccess` incluye las siguientes instrucciones para permitir el acceso de solo lectura a las notificaciones. Los usuarios que apliquen esta política pueden consultar notificaciones de recursos, pero no pueden crearlas, administrarlas ni suscribirse a ellas.

```

{
 "Sid": "CodeStarNotificationsPowerUserAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:DescribeNotificationRule"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codedeploy:*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListEventTypes",
 "codestar-notifications:ListTargets"
],
 "Resource": "*"
}

```

Para obtener más información acerca de IAM y las notificaciones consulte [Identity and Access Management para AWS CodeStar Notifications](#).

## CodeDeploy actualizaciones de las políticas AWS gestionadas

Consulta los detalles sobre las actualizaciones de las políticas AWS administradas CodeDeploy desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbese a la fuente RSS del CodeDeploy [Historial de documentos](#).

| Cambio                                                                                                 | Descripción                                                                                                                                                                                                                                                                                                                                                 | Fecha                |
|--------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Política administrada de <code>AWSCodeDeployRole</code> : actualización de la política existente       | <p>Se agregaron las acciones <code>elasticloadbalancing:DescribeLoadBalancerAttributes</code> y <code>elasticloadbalancing:DescribeTargetGroupAttributes</code> a la declaración de política para respaldar los cambios de Elastic Load Balancing.</p> <p>Para obtener más información sobre esta política, consulte <a href="#">AWSCodeDeployRole</a>.</p> | 16 de agosto de 2023 |
| Política administrada de <code>AWSCodeDeployFullAccess</code> : actualización de la política existente | <p>Se agregó la acción <code>chatbot:ListMicrosoftTeamsChannelConfigurations</code> a la declaración de política para respaldar los cambios en las notificaciones.</p> <p>Para obtener más información sobre esta política, consulte <a href="#">AWSCodeDeployRole</a>.</p>                                                                                 | 11 de mayo de 2023   |

| Cambio                                                                                                                     | Descripción                                                                                                                                                                                                                                                                                | Fecha                          |
|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <p>Política administrada de <code>AWSCodeDeployRole</code> : actualización de la política existente</p>                    | <p>Se agregó la acción <code>autoscaling:CreateOrUpdateTags</code> a la declaración de política para admitir los cambios en la autorización de Amazon EC2 Auto Scaling.</p> <p>Para obtener más información sobre esta política, consulte <a href="#">AWSCodeDeployRole</a>.</p>           | <p>3 de febrero de 2023</p>    |
| <p>Política administrada de <code>AmazonEC2RoleforAWSCodeDeployLimited</code> : actualización de la política existente</p> | <p>Se eliminó la acción <code>s3:ListBucket</code> de la declaración de política que incluye la condición <code>s3:ExistingObjectTag/UseWithCodeDeploy</code>.</p> <p>Para obtener más información sobre esta política, consulte <a href="#">AmazonEC2RoleforAWSCodeDeployLimited</a>.</p> | <p>22 de noviembre de 2021</p> |
| <p>Política administrada de <code>AWSCodeDeployRole</code> : actualización de la política existente</p>                    | <p>Se agregó la acción <code>autoscaling:PutWarmupPool</code> para admitir la <a href="#">adición de grupos de calentamiento a los grupos de Amazon EC2 Auto Scaling</a> para las implementaciones azul/verde.</p> <p>Se eliminaron las acciones duplicadas innecesarias.</p>              | <p>18 de mayo de 2021</p>      |

| Cambio                                    | Descripción                                                                                   | Fecha              |
|-------------------------------------------|-----------------------------------------------------------------------------------------------|--------------------|
| CodeDeploy comenzó a rastrear los cambios | CodeDeploy comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas. | 18 de mayo de 2021 |

## AWS CodeDeploy ejemplos de políticas basadas en identidad de

De forma predeterminada, los usuarios no tienen permiso para crear o modificar CodeDeploy recursos. Tampoco pueden realizar tareas con la AWS API AWS Management Console AWS CLI, o. Un administrador de IAM debe crear políticas de IAM que concedan permisos a los roles de IAM para realizar operaciones en la API en los recursos específicos que necesiten. A continuación, debe asociar esos roles de IAM a los usuarios o grupos que necesiten esos permisos.

Para obtener más información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas de JSON de ejemplo, consulte [Creación de políticas en la pestaña JSON](#) en la Guía del usuario de IAM.

En CodeDeploy, las políticas basadas en la identidad se utilizan para administrar los permisos a los distintos recursos relacionados con el proceso de implementación. Puede controlar el acceso a los siguientes tipos de recurso:

- Aplicaciones y revisiones de aplicación.
- Implementaciones
- Configuraciones de implementación.
- Instancias e instancias en las instalaciones

Las capacidades controladas por políticas de recursos varían en función del tipo de recurso, tal y como se describe en la siguiente tabla:

| Tipos de recurso | Capacidades                         |
|------------------|-------------------------------------|
| Todos            | Ver y enumerar detalles de recursos |
| Aplicaciones     | Crear recursos de                   |

| Tipos de recurso                  | Capacidades                                                                                                                     |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Configuraciones de implementación | Eliminación de recursos                                                                                                         |
| Grupos de implementaciones        |                                                                                                                                 |
| Implementaciones                  | Crear implementaciones<br>Detener implementaciones                                                                              |
| Revisiones de aplicaciones        | Registrar revisiones de aplicación                                                                                              |
| Aplicaciones                      | Recursos actualizados                                                                                                           |
| Grupos de implementaciones        |                                                                                                                                 |
| Instancias en las instalaciones   | Añadir etiquetas a instancias<br>Eliminar etiquetas de instancias<br>Registrar instancias<br>Cancelar el registro de instancias |

El siguiente ejemplo muestra una política de permisos que permite a un usuario eliminar el grupo de implementaciones denominado **WordPress\_DepGroup** asociado a la aplicación denominada **WordPress\_App** en la región **us-west-2**.

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:DeleteDeploymentGroup"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:WordPress_App/WordPress_DepGroup"
]
 }
]
}
```

```
}
```

## Temas

- [Ejemplos de políticas administradas por los clientes](#)
- [Prácticas recomendadas sobre las políticas](#)
- [Mediante la consola de CodeDeploy](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)

## Ejemplos de políticas administradas por los clientes

En esta sección, encontrará ejemplos de políticas que otorgan permisos para diversas CodeDeploy acciones. Estas políticas funcionan cuando utilizas la CodeDeploy API, AWS los SDK o el AWS CLI. Debe conceder permisos adicionales para las acciones que realice en la consola. Para obtener más información sobre cómo conceder permisos de consola, consulte [Mediante la consola de CodeDeploy](#).

### Note

Todos los ejemplos utilizan la región Oeste de EE. UU. (Oregón) (`us-west-2`) y contienen identificadores de cuenta ficticios.

## Ejemplos

- [Ejemplo 1: Permitir el permiso para realizar CodeDeploy operaciones en una sola región](#)
- [Ejemplo 2: permitir a un usuario registrar revisiones para una única aplicación](#)
- [Ejemplo 3: permitir a un usuario crear implementaciones para un único grupo de implementación](#)

### Ejemplo 1: Permitir el permiso para realizar CodeDeploy operaciones en una sola región

El siguiente ejemplo concede permisos para realizar CodeDeploy operaciones únicamente en la **us-west-2** región:

```
{
 "Version": "2012-10-17",
 "Statement" : [
 {
```

```

 "Effect" : "Allow",
 "Action" : [
 "codedeploy:*"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:444455556666:*"
]
 }
]
}

```

### Ejemplo 2: permitir a un usuario registrar revisiones para una única aplicación

En el siguiente ejemplo se conceden permisos para registrar revisiones de aplicación para todas las aplicaciones que comienzan por **Test** en la región **us-west-2**:

```

{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:RegisterApplicationRevision"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:444455556666:application:Test*"
]
 }
]
}

```

### Ejemplo 3: permitir a un usuario crear implementaciones para un único grupo de implementación

En el siguiente ejemplo se concede permiso para crear implementaciones para el grupo de implementación denominado **WordPress\_DepGroup** asociado a la aplicación denominada **WordPress\_App**, la configuración de implementación personalizada denominada **ThreeQuartersHealthy** y cualquier revisión de aplicación asociada a la aplicación denominada **WordPress\_App**. Todos estos recursos se encuentran en la región **us-west-2**.

```

{
 "Version": "2012-10-17",
 "Statement" : [

```

```

 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:CreateDeployment"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:WordPress_App/WordPress_DepGroup"
]
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:GetDeploymentConfig"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:444455556666:deploymentconfig:ThreeQuartersHealthy"
]
 },
 {
 "Effect" : "Allow",
 "Action" : [
 "codedeploy:GetApplicationRevision"
],
 "Resource" : [
 "arn:aws:codedeploy:us-west-2:444455556666:application:WordPress_App"
]
 }
]
}

```

## Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear CodeDeploy recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están



disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de trabajo](#) en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte [Política de validación de Analizador de acceso de IAM](#) en la Guía de usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte [Configuración del acceso a una API protegido por MFA](#) en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

## Mediante la consola de CodeDeploy

Si utilizas la CodeDeploy consola, debes tener un conjunto mínimo de permisos que te permita describir otros AWS recursos de tu cuenta. Para CodeDeploy utilizarlos en la CodeDeploy consola, debe tener permisos de los siguientes servicios:

- Amazon EC2 Auto Scaling
- AWS CodeDeploy
- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- AWS Identity and Access Management
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon CloudWatch

Si crea una política de IAM que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para los usuarios con ese rol para esa política de IAM. Para garantizar que esos usuarios puedan seguir utilizando la CodeDeploy consola, adjunte también la política `AWSCodeDeployReadOnlyAccess` gestionada a la función asignada al usuario, tal y como se describe en [AWS políticas gestionadas \(predefinidas\) para CodeDeploy](#).

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la CodeDeploy API.

## Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ViewOwnUserInfo",
```

```
 "Effect": "Allow",
 "Action": [
 "iam:GetUserPolicy",
 "iam:ListGroupsWithUser",
 "iam:ListAttachedUserPolicies",
 "iam:ListUserPolicies",
 "iam:GetUser"
],
 "Resource": ["arn:aws:iam::*:user/${aws:username}"]
 },
 {
 "Sid": "NavigateInConsole",
 "Effect": "Allow",
 "Action": [
 "iam:GetGroupPolicy",
 "iam:GetPolicyVersion",
 "iam:GetPolicy",
 "iam:ListAttachedGroupPolicies",
 "iam:ListGroupPolicies",
 "iam:ListPolicyVersions",
 "iam:ListPolicies",
 "iam:ListUsers"
],
 "Resource": "*"
 }
]
```

## Solución de problemas de identidades de AWS CodeDeploy y accesos

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con CodeDeploy un IAM.

### Temas

- [No estoy autorizado a realizar el iam: PassRole](#)
- [Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis CodeDeploy recursos](#)

### No estoy autorizado a realizar el iam: PassRole

Si recibes un mensaje de error que indica que no estás autorizado a realizar la `iam:PassRole` acción, debes actualizar tus políticas para que puedas transferirle CodeDeploy una función.

Algunos Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en ella. CodeDeploy Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

## Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis CodeDeploy recursos

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si CodeDeploy es compatible con estas funciones, consulte [¿Cómo AWS CodeDeploy funciona con IAM.](#)
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte [Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en](#) la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta [Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros](#) en la Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

## Referencia de permisos de CodeDeploy

Utilice la siguiente tabla a la hora de configurar el acceso y escribir políticas de permisos que pueda asociar a una identidad de IAM (políticas basadas en identidad). En la tabla se muestra cada operación de la CodeDeploy API, las acciones para las que puede conceder permisos para realizar la acción y el formato del ARN del recurso que se debe utilizar para conceder los permisos. Las acciones se especifican en el campo `Action` de la política. Debe especificar un ARN, con o sin un carácter comodín (\*), como el valor del recurso en el campo `Resource` de la política.

Puede utilizar claves AWS de condición generales en sus CodeDeploy políticas para expresar las condiciones. Para obtener una lista completa de las claves AWS anchas, consulta las [claves disponibles](#) en la Guía del usuario de IAM.

Para especificar una acción, use el prefijo `codedeploy:` seguido del nombre de operación de API (por ejemplo, `codedeploy:GetApplication` y `codedeploy:CreateApplication`). Para especificar varias acciones en una única instrucción, sepárelas con comas (por ejemplo, `"Action": ["codedeploy:action1", "codedeploy:action2"]`).

### Uso de caracteres comodín

Utilice el carácter de comodín (\*) en el ARN para especificar varias acciones o recursos. Por ejemplo, `codedeploy:*` especifica todas CodeDeploy las acciones y `codedeploy:Get*` especifica todas CodeDeploy las acciones que comienzan por la palabra. `Get` En el siguiente ejemplo se concede acceso a todos los grupos de implementaciones con nombres que comienzan por `West` y que se asocian a aplicaciones con nombres comienzan con `Test`.


```
arn:aws:codedeploy:us-west-2:444455556666:deploymentgroup:Test*/West*
```

Utilice comodines con los recursos que se muestran en la tabla a continuación:

- *application-name*

- *deployment-group-name*
- *deployment-configuration-name*
- *instance-ID*

No se puede utilizar comodines con *region* ni con *account-id*. Para obtener más información acerca de los comodines, consulte [identificadores de IAM](#) en la Guía del usuario de IAM.

 Note

En el ARN de cada acción, dos puntos (:) siguen al recurso. También puede seguir al recurso una barra diagonal (/). Para obtener más información, consulte los [CodeDeployARN de ejemplo](#).

CodeDeploy Operaciones de la API y permisos necesarios para las acciones

### [AddTagsToOnPremisesInstances](#)

Acción: `codedeploy:AddTagsToOnPremisesInstances`

Obligatorio para añadir etiquetas a una o más instancias on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [BatchGetApplicationRevisions](#)

Acción: `codedeploy:BatchGetApplicationRevisions`

Obligatorio para obtener información acerca de varias revisiones de aplicación asociadas al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

### [BatchGetApplications](#)

Acción: `codedeploy:BatchGetApplications`

Obligatorio para obtener información acerca de varias aplicaciones asociadas al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:*`

## [BatchGetDeploymentGroups](#)

Acción: `codedeploy:BatchGetDeploymentGroups`

Obligatorio para obtener información acerca de varios grupos de implementaciones asociados al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

## [BatchGetDeploymentInstances](#)

Acción: `codedeploy:BatchGetDeploymentInstances`

Obligatorio para obtener información acerca de una o varias instancias parte de un grupo de implementaciones.

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

## [BatchGetDeployments](#)

Acción: `codedeploy:BatchGetDeployments`

Obligatorio para obtener información acerca de varias implementaciones asociadas al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

## [BatchGetOnPremisesInstances](#)

Acción: `codedeploy:BatchGetOnPremisesInstances`

Obligatorio para obtener información acerca de una o más instancias on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:*`

## [ContinueDeployment](#)

Acción: `codedeploy:ContinueDeployment`

Obligatorio durante una implementación azul/verde para comenzar a registrar instancias en un entorno de sustitución con equilibradores de carga de Elastic Load Balancing.

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

## [CreateApplication](#)

Acción: `codedeploy:CreateApplication`

Obligatorio para crear una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

## [CreateDeployment](#)<sup>1</sup>

Acción: `codedeploy:CreateDeployment`

Obligatorio para crear una implementación para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

## [CreateDeploymentConfig](#)

Acción: `codedeploy:CreateDeploymentConfig`

Obligatorio para crear una configuración de implementación personalizada asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

## [CreateDeploymentGroup](#)

Acción: `codedeploy:CreateDeploymentGroup`

Obligatorio para crear un grupo de implementaciones de una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

## [DeleteApplication](#)

Acción: `codedeploy>DeleteApplication`

Obligatorio para eliminar una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

## [DeleteDeploymentConfig](#)

Acción: `codedeploy>DeleteDeploymentConfig`



Obligatorio para eliminar una configuración de implementación personalizada asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

### DeleteDeploymentGroup

Acción: `codedeploy>DeleteDeploymentGroup`

Obligatorio para eliminar un grupo de implementaciones de una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### DeregisterOnPremisesInstance

Acción: `codedeploy:DeregisterOnPremisesInstance`

Obligatorio para cancelar el registro de una instancia on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### GetApplication

Acción: `codedeploy:GetApplication`

Obligatorio para obtener información acerca de una única aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

### GetApplicationRevision

Acción: `codedeploy:GetApplicationRevision`

Obligatorio para obtener información acerca de una única revisión de aplicación para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

### GetDeployment

Acción: `codedeploy:GetDeployment`

Obligatorio para obtener información acerca de una única implementación de un grupo de implementaciones para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [GetDeploymentConfig](#)

Acción: `codedeploy:GetDeploymentConfig`

Obligatorio para obtener información acerca de una única configuración de implementación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentconfig/deployment-configuration-name`

### [GetDeploymentGroup](#)

Acción: `codedeploy:GetDeploymentGroup`

Obligatorio para obtener información acerca de un único grupo de implementaciones para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [GetDeploymentInstance](#)

Acción: `codedeploy:GetDeploymentInstance`

Obligatorio para obtener información acerca de una única instancia de una implementación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [GetOnPremisesInstance](#)

Acción: `codedeploy:GetOnPremisesInstance`

Obligatorio para obtener información acerca de una única instancia on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [ListApplicationRevisions](#)

Acción: `codedeploy>ListApplicationRevisions`

Obligatorio para obtener información acerca de todas las revisiones de aplicación para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:*`

### [ListApplications](#)

Acción: `codedeploy:ListApplications`

Obligatorio para obtener información acerca de todas las aplicaciones asociadas al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:*`

### [ListDeploymentConfigs](#)

Acción: `codedeploy:ListDeploymentConfigs`

Obligatorio para obtener información acerca de todas las configuraciones de implementaciones asociadas al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentconfig/*`

### [ListDeploymentGroups](#)

Acción: `codedeploy:ListDeploymentGroups`

Obligatorio para obtener información acerca de todos los grupos de implementaciones para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/*`

### [ListDeploymentInstances](#)

Acción: `codedeploy:ListDeploymentInstances`

Obligatorio para obtener información acerca de todas las instancias de una implementación asociadas al usuario.

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [ListDeployments](#)

Acción: `codedeploy:ListDeployments`

Obligatorio para obtener información acerca de todas las implementaciones de un grupo de implementación asociado al usuario, o para obtener todas las implementaciones asociadas al usuario.

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### [ListGitHubAccountTokenNames](#)

Acción: `codedeploy:ListGitHubAccountTokenNames`

Necesario para obtener una lista de los nombres de las conexiones almacenadas a GitHub las cuentas.

Recurso: `arn:aws:codedeploy:region:account-id:*`

### [ListOnPremisesInstances](#)

Acción: `codedeploy:ListOnPremisesInstances`

Obligatorio para obtener una lista de uno o varios nombres de instancias on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:*`

### [RegisterApplicationRevision](#)

Acción: `codedeploy:RegisterApplicationRevision`

Obligatorio para obtener información acerca de una revisión de aplicación para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

### [RegisterOnPremisesInstance](#)

Acción: `codedeploy:RegisterOnPremisesInstance`

Necesario para registrar una instancia local en. CodeDeploy

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [RemoveTagsFromOnPremisesInstances](#)

Acción: `codedeploy:RemoveTagsFromOnPremisesInstances`

Obligatorio para eliminar etiquetas desde una o más instancias on-premises.

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### [SkipWaitTimeForInstanceTermination](#)

Acción: `codedeploy:SkipWaitTimeForInstanceTermination`

Obligatorio para sobrescribir un tiempo de espera especificado y comenzar a terminar instancias en el entorno original inmediatamente después de dirigir correctamente el tráfico en una implementación "blue/green".

Recurso: `arn:aws:codedeploy:region:account-id:instance/instance-ID`

### StopDeployment

Acción: `codedeploy:StopDeployment`

Necesario para detener una implementación en curso en un grupo de implementaciones para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

### UpdateApplication<sup>3</sup>

Acción: `codedeploy:UpdateApplication`

Obligatorio para cambiar información de una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:application:application-name`

### UpdateDeploymentGroup<sup>3</sup>

Acción: `codedeploy:UpdateDeploymentGroup`

Obligatorio para cambiar información de un único grupo de implementaciones para una aplicación asociada al usuario de .

Recurso: `arn:aws:codedeploy:region:account-id:deploymentgroup:application-name/deployment-group-name`

<sup>1</sup> Cuando especifique permisos `CreateDeployment`, también debe especificar permisos `GetDeploymentConfig` para la configuración de implementación y permisos `GetApplicationRevision` o `RegisterApplicationRevision` para la revisión de la aplicación.

<sup>2</sup> Válido para `ListDeployments` cuando se proporciona un grupo de implementación, pero no cuando se muestran todas las implementaciones asociadas con el usuario.

<sup>3</sup> Para `UpdateApplication`, debe tener permisos `UpdateApplication` para el nombre de aplicación antiguo y nuevo. Para las acciones `UpdateDeploymentGroup` que implican cambiar el

nombre de un grupo de implementación, debe tener permisos `UpdateDeploymentGroup` tanto para el nombre de grupo de implementación antiguo como para el nuevo.

## Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger los datos de todos los servicios cuyos directores de servicio tengan acceso a los recursos de su cuenta.

Recomendamos utilizar las claves de contexto [aws: SourceArn](#) y [aws: SourceAccount](#) global condition en las políticas de recursos para limitar los permisos que se CodeDeploy otorgan a otro servicio al recurso. Si utiliza claves de contexto de condición global y el valor de `aws: SourceArn` contiene el ID de cuenta, el valor de `aws: SourceAccount` y la cuenta en el valor de `aws: SourceArn` deben utilizar el mismo ID de cuenta cuando se utiliza en la misma instrucción de política. Utilice `aws: SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utilice `aws: SourceAccount` si quiere que cualquier recurso de esa cuenta se asocie al uso entre servicios.

Para las implementaciones de EC2/on-premise AWS , Lambda y Amazon ECS normales, el valor de debe `aws: SourceArn` incluir el ARN del CodeDeploy grupo de implementaciones con el que CodeDeploy se permite asumir la función de IAM.

En el caso de las [implementaciones azul/verde de Amazon ECS creadas mediante AWS CloudFormation](#), el valor de `aws: SourceArn` debe incluir el CloudFormation ARN de pila con el que CodeDeploy se le permite asumir la función de IAM.

La forma más eficaz de protegerse contra el problema del suplente confuso es utilizar la clave de `aws: SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo o si está especificando varios recursos, utilice los caracteres comodín (\*) para las partes desconocidas.

Por ejemplo, puede usar la siguiente política de confianza con una implementación de EC2/on-premise AWS , Lambda o Amazon ECS normal:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "codedeploy.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": "111122223333"
 },
 "StringLike": {
 "aws:SourceArn": "arn:aws:codedeploy:us-
east-1:111122223333:deploymentgroup:myApplication/*"
 }
 }
 }
]
}
```

Para una [implementación azul/verde de Amazon ECS creada mediante AWS CloudFormation](#), puede usar:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "codedeploy.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": "111122223333"
 },
 "StringLike": {
 "aws:SourceArn": "arn:aws:cloudformation:us-
east-1:111122223333:stack/MyCloudFormationStackName/*"
 }
 }
 }
]
}
```

```
}
 }
}]
}
```

## Inicio de sesión y supervisión CodeDeploy

En esta sección se proporciona una descripción general de la supervisión, el registro y la respuesta a los incidentes en CodeDeploy.

## Auditoría de todas las interacciones con CodeDeploy

CodeDeploy está integrado con AWS CloudTrail un servicio que captura las llamadas a la API realizadas por su AWS cuenta o CodeDeploy en su nombre y envía los archivos de registro a un depósito de S3 que especifique. CloudTrail captura las llamadas a la API desde la CodeDeploy consola, desde CodeDeploy los comandos a través de ella o directamente desde las CodeDeploy API. AWS CLI Con la información recopilada por CloudTrail, puede determinar a qué solicitud se realizó CodeDeploy, la dirección IP de origen desde la que se realizó la solicitud, quién la hizo, cuándo se realizó, etc. Para obtener más información CloudTrail, consulte [Trabajar con archivos de CloudTrail registro](#) en la Guía del AWS CloudTrail usuario.

Puedes ver los datos de registro creados por una CodeDeploy implementación configurando el CloudWatch agente de Amazon para que vea los datos agregados en la CloudWatch consola o iniciando sesión en una instancia para revisar el archivo de registro. Para obtener más información, consulte [Enviar los registros CodeDeploy de los agentes a CloudWatch](#).

## Administración de incidentes y alertas

Puede usar Amazon CloudWatch Events para detectar cambios en el estado de una instancia o un despliegue (un evento) en sus CodeDeploy operaciones y reaccionar ante ellos. Luego, en función de las reglas que cree, CloudWatch Events invoca una o más acciones objetivo cuando una implementación o instancia entra en el estado que especifique en una regla. En función del tipo de cambio de estado, es posible que desee enviar notificaciones, capturar información de estado, tomar medidas correctivas, iniciar eventos o adoptar otras medidas. Puede seleccionar los siguientes tipos de objetivos cuando utilice CloudWatch Events como parte de sus CodeDeploy operaciones:

- AWS Lambda funciones



- Flujos de Kinesis
- Colas SQS de Amazon SQS
- Objetivos integrados (acciones CloudWatch de alarma)
- Temas de Amazon SNS

A continuación se indican algunos casos de uso:

- Utilice una función Lambda para pasar una notificación a un canal de Slack siempre que se produzcan errores en las implementaciones.
- Envíe datos acerca de las implementaciones o instancias a un flujo de Kinesis para permitir una monitorización completa del estado en tiempo real.
- Utilice acciones de CloudWatch alarma para detener, finalizar, reiniciar o recuperar automáticamente las instancias de EC2 cuando se produzca un evento de implementación o instancia que especifique.

Para obtener más información, consulta [Qué es Amazon CloudWatch Events](#) en la Guía del CloudWatch usuario de Amazon.

## Validación de conformidad para AWS CodeDeploy


Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte [Servicios de AWS Alcance por programa de cumplimiento](#) [Servicios de AWS](#) de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de [AWS cumplimiento > Programas AWS](#) .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#) .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- [Guías de inicio rápido sobre seguridad y cumplimiento](#): estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.

- Diseño de [arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services](#): en este documento técnico se describe cómo pueden utilizar AWS las empresas para crear aplicaciones aptas para la HIPAA.

 Note

No Servicios de AWS todas cumplen los requisitos de la HIPAA. Para más información, consulte la [Referencia de servicios compatibles con HIPAA](#).

- [AWS Recursos de](#) cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- [AWS Guías de cumplimiento para clientes](#): comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- [Evaluación de los recursos con reglas](#) en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- [AWS Security Hub](#)— Este Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la [Referencia de controles de Security Hub](#).
- [Amazon GuardDuty](#): Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- [AWS Audit Manager](#)— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

## Resiliencia en AWS CodeDeploy

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están

conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

[Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte la infraestructura global.AWS](#)

## Seguridad de la infraestructura en AWS CodeDeploy

Como servicio gestionado, AWS CodeDeploy está protegido por los procedimientos de seguridad de la red AWS global que se describen en el documento técnico [Amazon Web Services: descripción general de los procesos de seguridad](#).

Utiliza las llamadas a la API AWS publicadas para acceder a CodeDeploy través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Recomendamos TLS 1.3 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 los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

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.

# Referencia

Referencia.

Temas

- [CodeDeploy AppSpec referencia de archivo](#)
- [CodeDeploy referencia de configuración del agente](#)
- [AWS CloudFormation plantillas de CodeDeploy referencia](#)
- [Úselo CodeDeploy con Amazon Virtual Private Cloud](#)
- [CodeDeploy referencia del kit de recursos](#)
- [CodeDeploy cuotas](#)

## CodeDeploy AppSpec referencia de archivo

Esta sección se incluye solo como referencia. Para obtener una visión general conceptual del AppSpec archivo, consulte [Application Specification Files](#).

El archivo de especificaciones de la aplicación (AppSpec archivo) es un archivo con formato [YAML](#) o con formato JSON que se utiliza para administrar una implementación. CodeDeploy

### Note

El AppSpec archivo de una implementación de EC2/local debe tener un nombre `appspec.yaml`, a menos que vaya a realizar una implementación local. Para obtener más información, consulte [Creación de una implementación local](#).

Temas

- [AppSpec archivos en una plataforma informática Amazon ECS](#)
- [AppSpec archivos en una plataforma AWS Lambda informática](#)
- [AppSpec archivos en una plataforma informática local EC2/](#)
- [AppSpec Estructura de archivos](#)
- [AppSpec Ejemplo de archivo](#)

- [AppSpec Espaciado de archivos](#)
- [Valide su AppSpec archivo y su ubicación](#)

## AppSpec archivos en una plataforma informática Amazon ECS

En el caso de las aplicaciones de la plataforma informática Amazon ECS, el AppSpec archivo lo utiliza CodeDeploy para determinar:

- El archivo de definición de la tarea de Amazon ECS. Esto se especifica con su ARN en la `TaskDefinition` instrucción del archivo. AppSpec
- El contenedor y el puerto en el conjunto de tareas de sustitución donde el Equilibrador de carga de aplicación o el Equilibrador de carga de red redirige tráfico durante una implementación. Esto se especifica con la `LoadBalancerInfo` instrucción del AppSpec archivo.
- Información opcional acerca del servicio del servicio de Amazon ECS como, por ejemplo, la versión de la plataforma en la que se ejecuta, sus subredes y sus grupos de seguridad.
- Funciones de Lambda opcionales que ejecutar durante enlaces que se corresponden con eventos de ciclo de vida durante una implementación de Amazon ECS. Para obtener más información, consulte [AppSpec sección «ganchos» para una implementación de Amazon ECS](#).

## AppSpec archivos en una plataforma AWS Lambda informática

En el AWS caso de las aplicaciones de la plataforma de cómputo Lambda, el AppSpec archivo se utiliza CodeDeploy para determinar:

- Qué versión de función de Lambda se debe implementar.
- Qué funciones de Lambda se van a usar como pruebas de validación.

Un AppSpec archivo puede tener formato YAML o JSON. También puede introducir el contenido de un AppSpec archivo directamente en la CodeDeploy consola al crear una implementación.

## AppSpec archivos en una plataforma informática local EC2/

Si su aplicación utiliza la plataforma informática local EC2/el AppSpec archivo debe tener un nombre de archivo con formato YAML `appspec.yml` y debe colocarse en la raíz de la estructura de directorios del código fuente de la aplicación. De lo contrario, las implementaciones producirán un error. Lo utiliza para determinar: CodeDeploy

- Qué debe instalar en sus instancias desde la revisión de su aplicación en Amazon S3 o GitHub.
- Los enlaces de eventos del ciclo de vida que se deben ejecutar en respuesta a los eventos del ciclo de vida de la implementación.

Una vez completado el AppSpec archivo, lo agrupa, junto con el contenido que se va a implementar, en un archivo (zip, tar o tar comprimido). Para obtener más información, consulte [Trabajar con revisiones de aplicaciones para CodeDeploy](#).

#### Note

Los formatos de archivo tar y tar comprimido (.tar y .tar.gz) no son compatibles con las instancias de Windows Server.

Cuando tengas un archivo empaquetado (conocido CodeDeploy como revisión), lo subes a un bucket de Amazon S3 o a un repositorio de Git. A continuación, se utiliza CodeDeploy para implementar la revisión. Para ver instrucciones, consulte [Cree una implementación con CodeDeploy](#).

El archivo appspec.yml para la implementación en una plataforma de informática de EC2/en las instalaciones se guarda en el directorio raíz de la revisión. Para obtener más información, consulte [Agregue un AppSpec archivo para una implementación local de EC2/](#) y [Planifique una revisión para CodeDeploy](#).

## AppSpec Estructura de archivos

La siguiente es la estructura de alto nivel de un AppSpec archivo que se utiliza para las implementaciones en plataformas informáticas AWS Lambda y EC2/on-premise.

Los valores de un AppSpec archivo con formato YAML que sean cadenas no deben estar entre comillas («») a menos que se especifique lo contrario.

## AppSpec estructura de archivos para las implementaciones de Amazon ECS

#### Note

Este AppSpec archivo está escrito en YAML, pero puede usar la misma estructura para escribir uno en JSON. Las cadenas de un AppSpec archivo con formato JSON siempre aparecen entre comillas («»).

```
version: 0.0
resources:
 ecs-service-specifications
hooks:
 deployment-lifecycle-event-mappings
```

En esta estructura:

#### versión

En esta sección se especifica la versión del archivo. AppSpec No cambie este valor. Es obligatorio. El único valor permitido actualmente es **0.0**. Está reservado CodeDeploy para uso futuro.

Especifique version con una cadena.

#### resources

En esta sección se especifica información sobre la aplicación de Amazon ECS que se va a implementar.

Para obtener más información, consulte [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#).

#### enlaces

Esta sección especifica las funciones de Lambda que se van a ejecutar en enlaces de eventos específicos del ciclo de vida de la implementación para validar la implementación.

Para obtener más información, consulte [Lista de enlaces de eventos de ciclo de vida para una implementación de Amazon ECS](#).

## AppSpec estructura de archivos para despliegues de AWS Lambda

### Note

Este AppSpec archivo está escrito en YAML, pero puede usar la misma estructura para escribir un AppSpec archivo para una implementación de Lambda en JSON. Las cadenas de un AppSpec archivo con formato JSON siempre aparecen entre comillas («»).

```
version: 0.0
resources:
 lambda-function-specifications
hooks:
 deployment-lifecycle-event-mappings
```

En esta estructura:

#### versión

En esta sección se especifica la versión del archivo. AppSpec No cambie este valor. Es obligatorio. El único valor permitido actualmente es **0.0**. Está reservado CodeDeploy para uso futuro.

Especifique version con una cadena.

#### resources

En esta sección se especifica información sobre la función de Lambda que se va a implementar.

Para obtener más información, consulte [AppSpec sección de «recursos» \(solo Amazon ECS y AWS Lambda despliegues\)](#).

#### enlaces

Esta sección especifica las funciones de Lambda que se van a ejecutar en eventos específicos del ciclo de vida de la implementación para validar la implementación.

Para obtener más información, consulte [AppSpec sección de «ganchos»](#).

## AppSpec estructura de archivos para despliegues de EC2/locales

```
version: 0.0
os: operating-system-name
files:
 source-destination-files-mappings
permissions:
 permissions-specifications
hooks:
 deployment-lifecycle-event-mappings
```

En esta estructura:



## versión

En esta sección se especifica la versión del archivo. AppSpec No cambie este valor. Es obligatorio. El único valor permitido actualmente es **0.0**. Está reservado CodeDeploy para uso futuro.

Especifique version con una cadena.

## os

Esta sección especifica el valor del sistema operativo de la instancia en la que se va a realizar la implementación. Es obligatorio. Se pueden especificar los siguientes valores:

- linux: la instancia es una instancia de Amazon Linux, Ubuntu Server o RHEL.
- windows: la instancia es una instancia de Windows Server.

Especifique os con una cadena.

## files

Esta sección especifica los nombres de los archivos que deben copiarse en la instancia durante el evento Install de la implementación.

Para obtener más información, consulte [AppSpec Sección de «archivos» \(solo para despliegues de EC2/on-premise\)](#).

## permissions

Esta sección especifica cómo los permisos especiales, si hay alguno, deben aplicarse a los archivos de la sección `files` cuando se copien en la instancia. Esta sección se aplica únicamente a instancias de Amazon Linux, Ubuntu Server y Red Hat Enterprise Linux (RHEL).

Para obtener más información, consulte [AppSpec sección de «permisos» \(solo para implementaciones de EC2/locales\)](#).

## enlaces

Esta sección especifica los scripts que se van a ejecutar en eventos específicos del ciclo de vida de la implementación durante la implementación.

Para obtener más información, consulte [AppSpec sección de «ganchos»](#).

## Temas

- [AppSpec Sección de «archivos» \(solo para despliegues de EC2/on-premise\)](#)
- [AppSpec sección de «recursos» \(solo Amazon ECS y AWS Lambda despliegues\)](#)
- [AppSpec sección de «permisos» \(solo para implementaciones de EC2/locales\)](#)
- [AppSpec sección de «ganchos»](#)

### AppSpec Sección de «archivos» (solo para despliegues de EC2/on-premise)

Proporciona información CodeDeploy sobre los archivos de la revisión de la aplicación que se deben instalar en la instancia durante el evento de instalación de la implementación. Esta sección solo es necesaria si va a copiar archivos de la revisión en ubicaciones de la instancia durante la implementación.

Esta sección tiene la siguiente estructura:

```
files:
 - source: source-file-location-1
 destination: destination-file-location-1
file_exists_behavior: DISALLOW|OVERWRITE|RETAIN
```

Se pueden establecer varios pares `source` y `destination`.

La instrucción `source` identifica el archivo o directorio de la revisión que se va a copiar en la instancia:

- Si `source` hace referencia a un archivo, solo se copian los archivos especificados en la instancia.
- Si `source` hace referencia a un directorio, se copian todos los archivos del directorio en la instancia.
- Si `source` es una sola barra diagonal ("/" para instancias de Amazon Linux, RHEL y Ubuntu Server o "\" para instancias de Windows Server), se copian todos los archivos de la revisión en la instancia.

Las rutas utilizadas en `source` son relativas al archivo `appspec.yml`, que debería estar en la raíz de la revisión. Para obtener más información sobre la estructura de archivos de una revisión, consulte [Planifique una revisión para CodeDeploy](#).

La instrucción `destination` identifica el lugar de la instancia en el que deben copiarse los archivos. Debe ser una ruta totalmente cualificada, como `/root/destination/directory` (en Linux, RHEL y Ubuntu) o `c:\destination\folder` (en Windows).

`source` y `destination` se especifican con una cadena.

La `file_exists_behavior` instrucción es opcional y especifica cómo se gestionan los CodeDeploy archivos que ya existen en la ubicación de destino de la implementación, pero que no formaban parte de la anterior implementación exitosa. Puede adoptar cualquiera de los siguientes valores:

- **NO PERMITIR:** se produce un error en la implementación. Esta es la opción predeterminada si no se especifica ninguna opción.
- **SOBRESCRIBIR:** la versión del archivo de la revisión de la aplicación sustituye a la versión ya incluida en la instancia.
- **RETENER:** la versión del archivo que ya está en la instancia se conserva y se usa como parte de la nueva implementación.

Si usa la configuración `file_exists_behavior`, debe comprender que esta configuración:

- Solo se puede especificar una vez y se aplica a todos los archivos y directorios enumerados en `files:`.
- tiene prioridad sobre la `--file-exists-behavior` AWS CLI opción y la opción de `fileExistsBehavior` API (ambas opciones también son opcionales).

Esta es una sección `files` de ejemplo para una instancia de Amazon Linux, Ubuntu Server o RHEL.

```
files:
 - source: Config/config.txt
 destination: /webapps/Config
 - source: source
 destination: /webapps/myApp
```

En este ejemplo, se realizan las dos operaciones siguientes durante el evento `Install`:

1. Copiar el archivo `Config/config.txt` en la revisión en la ruta de `/webapps/Config/config.txt` de la instancia.

2. Copiar recursivamente todos los archivos del directorio `source` de la revisión en el directorio `/webapps/myApp` de la instancia.

### Ejemplos de la sección "files"

Los siguientes ejemplos muestran cómo especificar la sección `files`. Aunque estos ejemplos describen estructuras de archivos y directorios (carpetas) de Windows Server, se pueden adaptar fácilmente para instancias de Amazon Linux, Ubuntu Server y RHEL.

#### Note

Solo las implementaciones de EC2/en las instalaciones utilizan la sección `files`. No se aplica a las implementaciones de AWS Lambda.

En los siguientes ejemplos, presuponemos que estos archivos están en el paquete en la raíz de `source`:

- `appspec.yml`
- `my-file.txt`
- `my-file-2.txt`
- `my-file-3.txt`

```
1) Copy only my-file.txt to the destination folder c:\temp.
#
files:
 - source: .\my-file.txt
 destination: c:\temp
#
Result:
c:\temp\my-file.txt
#

#
2) Copy only my-file-2.txt and my-file-3.txt to the destination folder c:\temp.
#
files:
 - source: my-file-2.txt
 destination: c:\temp
```

```
- source: my-file-3.txt
 destination: c:\temp
#
Result:
c:\temp\my-file-2.txt
c:\temp\my-file-3.txt
#

#
3) Copy my-file.txt, my-file-2.txt, and my-file-3.txt (along with the appspec.yml
file) to the destination folder c:\temp.
#
files:
 - source: \
 destination: c:\temp
#
Result:
c:\temp\appspec.yml
c:\temp\my-file.txt
c:\temp\my-file-2.txt
c:\temp\my-file-3.txt
```

En los siguientes ejemplos, presuponemos que `appspec.yml` está en el paquete en la raíz de `source` junto con una carpeta denominada `my-folder` que contiene tres archivos:

- `appspec.yml`
- `my-folder\my-file.txt`
- `my-folder\my-file-2.txt`
- `my-folder\my-file-3.txt`

```
4) Copy the 3 files in my-folder (but do not copy my-folder itself) to the
destination folder c:\temp.
#
files:
 - source: .\my-folder
 destination: c:\temp
#
Result:
c:\temp\my-file.txt
c:\temp\my-file-2.txt
c:\temp\my-file-3.txt
```

```
#

#
5) Copy my-folder and its 3 files to my-folder within the destination folder c:\temp.
#
files:
 - source: .\my-folder
 destination: c:\temp\my-folder
#
Result:
c:\temp\my-folder\my-file.txt
c:\temp\my-folder\my-file-2.txt
c:\temp\my-folder\my-file-3.txt
#

#
6) Copy the 3 files in my-folder to other-folder within the destination folder c:
\temp.
#
files:
 - source: .\my-folder
 destination: c:\temp\other-folder
#
Result:
c:\temp\other-folder\my-file.txt
c:\temp\other-folder\my-file-2.txt
c:\temp\other-folder\my-file-3.txt
#

#
7) Copy only my-file-2.txt and my-file-3.txt to my-folder within the destination
folder c:\temp.
#
files:
 - source: .\my-folder\my-file-2.txt
 destination: c:\temp\my-folder
 - source: .\my-folder\my-file-3.txt
 destination: c:\temp\my-folder
#
Result:
c:\temp\my-folder\my-file-2.txt
c:\temp\my-folder\my-file-3.txt
#

```

```
#
8) Copy only my-file-2.txt and my-file-3.txt to other-folder within the destination
folder c:\temp.
#
files:
 - source: .\my-folder\my-file-2.txt
 destination: c:\temp\other-folder
 - source: .\my-folder\my-file-3.txt
 destination: c:\temp\other-folder
#
Result:
c:\temp\other-folder\my-file-2.txt
c:\temp\other-folder\my-file-3.txt
#

#
9) Copy my-folder and its 3 files (along with the appspec.yml file) to the
destination folder c:\temp. If any of the files already exist on the instance,
overwrite them.
#
files:
 - source: \
 destination: c:\temp
file_exists_behavior: OVERWRITE
#
Result:
c:\temp\appspec.yml
c:\temp\my-folder\my-file.txt
c:\temp\my-folder\my-file-2.txt
c:\temp\my-folder\my-file-3.txt
```

## AppSpec sección de «recursos» (solo Amazon ECS y AWS Lambda despliegues)

El contenido de la 'resources' sección del AppSpec archivo varía en función de la plataforma informática de la implementación. La sección 'resources' de una implementación de Amazon ECS contiene la definición de tareas de Amazon ECS, contenedor y puerto para dirigir el tráfico a su conjunto de tareas de Amazon ECS actualizado y otra información opcional. La 'resources' sección de una AWS Lambda implementación contiene el nombre, el alias, la versión actual y la versión de destino de una función Lambda.

### Temas

- [AppSpec sección de «recursos» para despliegues de AWS Lambda](#)

- [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#)

## AppSpec sección de «recursos» para despliegues de AWS Lambda

La sección 'resources' especifica la función de Lambda que se va a implementar y tiene la siguiente estructura:

YAML:

```
resources:
 - name-of-function-to-deploy:
 type: "AWS::Lambda::Function"
 properties:
 name: name-of-lambda-function-to-deploy
 alias: alias-of-lambda-function-to-deploy
 currentversion: version-of-the-lambda-function-traffic-currently-points-to
 targetversion: version-of-the-lambda-function-to-shift-traffic-to
```

JSON:

```
"resources": [
 {
 "name-of-function-to-deploy" {
 "type": "AWS::Lambda::Function",
 "properties": {
 "name": "name-of-lambda-function-to-deploy",
 "alias": "alias-of-lambda-function-to-deploy",
 "currentversion": "version-of-the-lambda-function-traffic-currently-points-to",
 "targetversion": "version-of-the-lambda-function-to-shift-traffic-to"
 }
 }
 }
]
```

Cada propiedad se especifica con una cadena.

- **name**: obligatorio. Es el nombre de la función de Lambda que se va a implementar.
- **alias**: obligatorio. Es el nombre del alias de la función de Lambda.
- **currentversion**: obligatorio. Es la versión de la función de Lambda a la que apunta actualmente el tráfico. Este valor debe ser un entero positivo válido.



- `targetversion`: obligatorio. Es la versión de la función de Lambda a la que se va a desviar el tráfico. Este valor debe ser un entero positivo válido.

## AppSpec sección de «recursos» para las implementaciones de Amazon ECS

La sección 'resources' especifica el servicio de Amazon ECS que se va a implementar y tiene la siguiente estructura:

YAML:

```
Resources:
 - TargetService:
 Type: AWS::ECS::Service
 Properties:
 TaskDefinition: "task-definition-arn"
 LoadBalancerInfo:
 ContainerName: "ecs-container-name"
 ContainerPort: "ecs-application-port"
Optional properties
 PlatformVersion: "ecs-service-platform-version"
 NetworkConfiguration:
 AwsVpcConfiguration:
 Subnets: ["ecs-subnet-1", "ecs-subnet-n"]
 SecurityGroups: ["ecs-security-group-1", "ecs-security-group-n"]
 AssignPublicIp: "ENABLED | DISABLED"
 CapacityProviderStrategy:
 - Base: integer
 CapacityProvider: "capacityProviderA"
 Weight: integer
 - Base: integer
 CapacityProvider: "capacityProviderB"
 Weight: integer
```

JSON:

```
"Resources": [
 {
 "TargetService": {
 "Type": "AWS::ECS::Service",
 "Properties": {
 "TaskDefinition": "task-definition-arn",
 "LoadBalancerInfo": {
```

```

 "ContainerName": "ecs-container-name",
 "ContainerPort": "ecs-application-port"
 },
 "PlatformVersion": "ecs-service-platform-version",
 "NetworkConfiguration": {
 "AwsVpcConfiguration": {
 "Subnets": [
 "ecs-subnet-1",
 "ecs-subnet-n"
],
 "SecurityGroups": [
 "ecs-security-group-1",
 "ecs-security-group-n"
],
 "AssignPublicIp": "ENABLED | DISABLED"
 }
 },
 "CapacityProviderStrategy": [
 {
 "Base": integer,
 "CapacityProvider": "capacityProviderA",
 "Weight": integer
 },
 {
 "Base": integer,
 "CapacityProvider": "capacityProviderB",
 "Weight": integer
 }
]
}
}
}
]

```

Cada propiedad se especifica con una cadena excepto ContainerPort, que es un número.

- **TaskDefinition**: obligatorio. Esta es la definición de tarea para el servicio de Amazon ECS que se va a implementar. Se especifica con el ARN de la definición de tarea. El formato del ARN es `arn:aws:ecs:aws-region:account-id:task-definition/task-definition-family:task-definition-revision`. Para obtener más información, consulte [Nombres de recursos de Amazon \(ARN\) y espacios de nombres AWS de servicios](#).

**Note**

La parte :*task-definition-revision* del ARN es opcional. Si se omite, Amazon ECS utiliza la última revisión ACTIVA de la definición de la tarea.

- **ContainerName:** obligatorio. Este es el nombre del contenedor de Amazon ECS que contiene la aplicación de Amazon ECS. Debe ser un contenedor especificado en la definición de tarea de Amazon ECS.
- **ContainerPort:** obligatorio. Este es el puerto del contenedor al que se enrutará el tráfico.
- **PlatformVersion:** Opcional. La versión de la plataforma de las tareas Fargate en el servicio de Amazon ECS implementado. Para obtener más información, consulte [Versiones de la plataforma de AWS Fargate](#). Si no se especifica, se utilizará LATEST de forma predeterminada.
- **NetworkConfiguration:** Opcional. En `AwsVpcConfiguration`, puede especificar lo siguiente. Para obtener más información, consulte la referencia [AwsVpcConfiguration](#) de la API de Amazon ECS Container Service.
  - **Subnets:** Opcional. Una lista separada por comas de una o varias subredes en su servicio de Amazon ECS.
  - **SecurityGroups:** Opcional. Una lista separada por comas de uno o varios grupos de seguridad en su Amazon Elastic Container Service.
  - **AssignPublicIp:** Opcional. Una cadena que especifica si su interfaz de red elástica del servicio de Amazon ECS recibe una dirección IP pública. Los valores válidos son ENABLED y DISABLED.

**Note**

Deben especificarse todos o ninguno de los ajustes de `NetworkConfiguration`. Por ejemplo, si desea especificar `Subnets`, también debe especificar `SecurityGroups` y `AssignPublicIp`. Si no se especifica ninguno, CodeDeploy utiliza la configuración actual de Amazon ECS de la red.

- **CapacityProviderStrategy:** Opcional. Una lista de los proveedores de capacidad de Amazon ECS que desea utilizar para su implementación. Para obtener más información, consulte [Proveedores de capacidad de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service. Puede especificar la siguiente configuración para cada

proveedor de capacidad. Para obtener más información sobre estos ajustes, consulte [AWS::ECS::ServiceCapacityProviderStrategyItem](#) la Guía AWS CloudFormation del usuario

- **Base:** Opcional. El valor de base designa cuántas tareas, como mínimo, se ejecutarán en el proveedor de capacidad especificado. Solo un proveedor de capacidad en una estrategia de proveedor de capacidad puede tener una base definida. Si no se especifica ningún valor, se utiliza el valor predeterminado 0.
- **CapacityProvider:** Opcional. Nombre abreviado del proveedor de capacidad. Ejemplo: `capacityProviderA`
- **Weight:** Opcional.

El valor peso designa el porcentaje relativo del número total de tareas lanzadas que debe utilizar el proveedor de capacidad especificado. El valor `weight` se tomará en cuenta luego de que el valor base se cumpla, en el caso de haber sido definido.

Si no se especifica ningún valor, se utiliza el valor predeterminado 0. Cuando se especifican varios proveedores de capacidad dentro de una estrategia de provisión de capacidad, al menos uno de los proveedores de capacidad deberá tener un valor de peso superior a cero y los proveedores de capacidad con un peso de 0 no se utilizarán para asignar tareas. Si especifican varios proveedores de capacidad en una estrategia en la que todos tienen un peso de 0, se producirá un error en cualquiera de las acciones `RunTask` o `CreateService` que utilicen la estrategia de provisión de capacidad.

Por ejemplo, si define una estrategia que contiene dos proveedores de capacidad y ambos tienen un peso de 1, cuando base se cumpla las tareas se dividirán uniformemente entre los dos proveedores de capacidad. Siguiendo esta misma lógica, si especifica una ponderación de 1 para `capacityProviderA` y una ponderación de 4 para `capacityProviderB`, entonces por cada tarea que se ejecute usando `capacityProviderA`, habrá cuatro tareas que usen `capacityProviderB`.

## AppSpec sección de «permisos» (solo para implementaciones de EC2/locales)

La sección `'permissions'` especifica cómo los permisos especiales, si hay alguno, deben aplicarse a los archivos y directorios/carpetas de la sección `'files'` una vez copiados en la instancia. Puede especificar varias instrucciones `object`. Esta sección es opcional. Se aplica únicamente a las instancias de Amazon Linux, Ubuntu Server y RHEL.

**Note**

La sección 'permissions' solo se utiliza para las implementaciones de EC2/en las instalaciones. No se utiliza para las implementaciones de AWS Lambda o Amazon ECS.

Esta sección tiene la siguiente estructura:

```
permissions:
 - object: object-specification
 pattern: pattern-specification
 except: exception-specification
 owner: owner-account-name
 group: group-name
 mode: mode-specification
 acls:
 - acls-specification
 context:
 user: user-specification
 type: type-specification
 range: range-specification
 type:
 - object-type
```

Las instrucciones son las siguientes:

- **object**: obligatorio. Este es un conjunto de objetos del sistema de archivos (archivos o directorios/carpetas) a los que se aplican los permisos especificados una vez que se copian en la instancia.

Especifique **object** con una cadena.

- **pattern**: opcional. Especifica un patrón para aplicar permisos. Si no se especifica o se especifica con los caracteres especiales "\*\*\*", los permisos se aplican a todos los archivos o directorios coincidentes, en función del valor de **type**.

Especifique **pattern** con una cadena con comillas ("").

- **except**: opcional. Especifica los archivos o directorios que son excepciones a **pattern**.

Especifique **except** con una lista separada por comas de cadenas entre corchetes.

- **owner**: opcional. El nombre del propietario de `object`. Si no se especifica, todos los propietarios existentes aplicados a la estructura original de archivos o directorios/carpetas permanecen sin cambios después de la operación de copia.

Especifique `owner` con una cadena.

- **group**: opcional. El nombre del grupo de `object`. Si no se especifica, todos los grupos existentes aplicados a la estructura original de archivos o directorios/carpetas permanecen sin cambios después de la operación de copia.

Especifique `group` con una cadena.

- **mode**: opcional. Un valor numérico que especifica los permisos que se van a aplicar a `object`. La configuración del modo sigue la sintaxis del comando `chmod` de Linux.

#### Important

Si el valor incluye un cero a la izquierda, debe ponerlo entre comillas dobles o eliminar el cero inicial para que solo queden tres dígitos.

#### Note

El ajuste `u+x` no admite una notación simbólica como `mode`.

Ejemplos:

- `mode: "0644"` otorga permisos de lectura y escritura al propietario del objeto (6), permisos de solo lectura al grupo (4) y permisos de solo lectura a todos los demás usuarios (4).
- `mode: 644` concede los mismos permisos que `mode: "0644"`.
- `mode: 4755` establece el atributo `setuid` (4), otorga permisos de control total al propietario (7), otorga permisos de lectura y ejecución al grupo (5) y otorga permisos de lectura y ejecución a todos los demás usuarios (5).

Para ver más ejemplos, consulte la documentación del comando `chmod` de Linux.

Si no se especifica `mode`, todos los modos existentes aplicados a la estructura original de archivos o carpetas permanecen sin cambios después de la operación de copia.

- `ac1s`: opcional. Una lista de cadenas de caracteres que representan una o varias entradas de la lista de control de acceso (ACL) aplicadas a `object`. Por ejemplo, `u: bob:rw` representa permisos de lectura y escritura para el usuario **bob**. (Para ver más ejemplos, consulte los ejemplos de formato de entradas de ACL en la documentación del comando `setfacl` de Linux). Puede especificar varias entradas de ACL. Si no se especifica `ac1s`, todas las listas de control de acceso aplicadas a la estructura original de archivos o directorios/carpetas permanecen sin cambios después de la operación de copia. Reemplazan a todas las listas de control de acceso existentes.

Especifique una `ac1s` con un guion (-), seguido de un espacio y, a continuación, una cadena (por ejemplo, `- u: jane:rw`). Si tiene varias ACL, cada una de ellas se especifica en una línea independiente.

#### Note

Si se configuran usuarios sin nombre, grupos sin nombre u otras entradas de ACL similares, se produce un error en el AppSpec archivo. Utilice `mode` para especificar estos tipos de permisos en su lugar.

- `context`: opcional. Para las instancias habilitadas para Security-Enhanced Linux (SELinux), una lista de etiquetas de contexto relevantes para la seguridad que se aplica a los objetos copiados. Las etiquetas se especifican como claves que contienen `user`, `type` y `range`. (Para obtener más información, consulte la documentación de SELinux). Cada clave se escribe con una cadena. Si no se especifica, todas las etiquetas existentes aplicadas a la estructura original de archivos o directorios/carpetas permanecen sin cambios después de la operación de copia.
  - `user`: opcional. El usuario de SELinux.
  - `type`: opcional. El nombre de tipo de SELinux.
  - `range`: opcional. El rango especificado de SELinux. No tiene ningún efecto a no ser que Multi-Level Security (MLS) y Multi-Category Security (MCS) estén habilitados en la máquina. Si no está habilitada, el valor predeterminado de `range` es `s0`.

Especifique `context` con una cadena (por ejemplo, `user: unconfined_u`). Cada `context` se especifica en una línea separada.

- `type`: opcional. Los tipos de objetos a los que se van a aplicar los permisos especificados. `type` es una cadena que se puede establecer en **file** o en **directory**. Si se especifica **file**, los permisos se aplican únicamente a los archivos que estén incluidos en `object` después de la operación de copia (y no al propio `object`). Si se especifica **directory**, los permisos se aplican

recursivamente a todos los directorios o carpetas incluidos en `object` después de la operación de copia (y no al propio `object`).

Especifique `type` con un guion (-), seguido de un espacio y, a continuación, una cadena (por ejemplo, `- file`).

### Ejemplo de la sección "permissions"

En el siguiente ejemplo se muestra cómo especificar la sección 'permissions' con las instrucciones `object`, `pattern`, `except`, `owner`, `mode` y `type`. Este ejemplo se aplica únicamente a las instancias de Amazon Linux, Ubuntu Server y RHEL. En este ejemplo, se presupone que se copian los siguientes archivos y carpetas en la instancia en esta jerarquía:

```
/tmp
 |-- my-app
 |-- my-file-1.txt
 |-- my-file-2.txt
 |-- my-file-3.txt
 |-- my-folder-1
 |-- my-file-4.txt
 |-- my-file-5.txt
 |-- my-file-6.txt
 |-- my-folder-2
 |-- my-file-7.txt
 |-- my-file-8.txt
 |-- my-file-9.txt
 |-- my-folder-3
```

El siguiente AppSpec archivo muestra cómo establecer los permisos en estos archivos y carpetas después de copiarlos:

```
version: 0.0
os: linux
Copy over all of the folders and files with the permissions they
were originally assigned.
files:
- source: ./my-file-1.txt
 destination: /tmp/my-app
- source: ./my-file-2.txt
 destination: /tmp/my-app
- source: ./my-file-3.txt
```



```
destination: /tmp/my-app
- source: ./my-folder-1
 destination: /tmp/my-app/my-folder-1
- source: ./my-folder-2
 destination: /tmp/my-app/my-folder-2
1) For all of the files in the /tmp/my-app folder ending in -3.txt
(for example, just my-file-3.txt), owner = adm, group = wheel, and
mode = 464 (-r--rw-r--).
permissions:
- object: /tmp/my-app
 pattern: "*-3.txt"
 owner: adm
 group: wheel
 mode: 464
 type:
 - file
2) For all of the files ending in .txt in the /tmp/my-app
folder, but not for the file my-file-3.txt (for example,
just my-file-1.txt and my-file-2.txt),
owner = ec2-user and mode = 444 (-r--r--r--).
- object: /tmp/my-app
 pattern: "*.txt"
 except: [my-file-3.txt]
 owner: ec2-user
 mode: 444
 type:
 - file
3) For all the files in the /tmp/my-app/my-folder-1 folder except
for my-file-4.txt and my-file-5.txt, (for example,
just my-file-6.txt), owner = operator and mode = 646 (-rw-r--rw-).
- object: /tmp/my-app/my-folder-1
 pattern: "*"
 except: [my-file-4.txt, my-file-5.txt]
 owner: operator
 mode: 646
 type:
 - file
4) For all of the files that are immediately under
the /tmp/my-app/my-folder-2 folder except for my-file-8.txt,
(for example, just my-file-7.txt and
my-file-9.txt), owner = ec2-user and mode = 777 (-rwxrwxrwx).
- object: /tmp/my-app/my-folder-2
 pattern: "*"
 except: [my-file-8.txt]
```

```

 owner: ec2-user
 mode: 777
 type:
 - file
5) For all folders at any level under /tmp/my-app that contain
the name my-folder but not
/tmp/my-app/my-folder-2/my-folder-3 (for example, just
/tmp/my-app/my-folder-1 and /tmp/my-app/my-folder-2),
owner = ec2-user and mode = 555 (dr-xr-xr-x).
- object: /tmp/my-app
 pattern: "*my-folder*"
 except: [tmp/my-app/my-folder-2/my-folder-3]
 owner: ec2-user
 mode: 555
 type:
 - directory
6) For the folder /tmp/my-app/my-folder-2/my-folder-3,
group = wheel and mode = 564 (dr-xrw-r--).
- object: /tmp/my-app/my-folder-2/my-folder-3
 group: wheel
 mode: 564
 type:
 - directory

```

Los permisos resultantes son los siguientes:

```

-r--r--r-- ec2-user root my-file-1.txt
-r--r--r-- ec2-user root my-file-2.txt
-r--rw-r-- adm wheel my-file-3.txt

dr-xr-xr-x ec2-user root my-folder-1
-rw-r--r-- root root my-file-4.txt
-rw-r--r-- root root my-file-5.txt
-rw-r--rw- operator root my-file-6.txt

dr-xr-xr-x ec2-user root my-folder-2
-rwxrwxrwx ec2-user root my-file-7.txt
-rw-r--r-- root root my-file-8.txt
-rwxrwxrwx ec2-user root my-file-9.txt

dr-xrw-r-- root wheel my-folder-3

```

En el siguiente ejemplo se muestra cómo especificar la sección 'permissions' con la adición de las instrucciones `acls` y `context`. Este ejemplo se aplica únicamente a las instancias de Amazon Linux, Ubuntu Server y RHEL.

```
permissions:
 - object: /var/www/html/WordPress
 pattern: "*"
 except: [/var/www/html/WordPress/ReadMe.txt]
 owner: bob
 group: writers
 mode: 644
 acls:
 - u:mary:rw
 - u:sam:rw
 - m::rw
 context:
 user: unconfined_u
 type: httpd_sys_content_t
 range: s0
 type:
 - file
```

## AppSpec sección de «ganchos»

El contenido de la 'hooks' sección del AppSpec archivo varía en función de la plataforma informática de la implementación. La sección 'hooks' de una implementación de EC2/en las instalaciones contiene los mapeos que vinculan los enlaces de eventos del ciclo de vida de la implementación a uno o varios scripts. La sección 'hooks' de una implementación de Lambda o de Amazon ECS especifica las funciones de validación de Lambda que se ejecutan durante un evento del ciclo de vida de la implementación. Si un enlace de evento no está presente, no se ejecuta ninguna operación para ese evento. Esta sección solo es necesaria si ejecuta scripts o funciones de validación de Lambda como parte de la implementación.

### Temas

- [AppSpec sección «ganchos» para una implementación de Amazon ECS](#)
- [AppSpec sección de «ganchos» para una implementación de AWS Lambda](#)
- [AppSpec Sección de «enganches» para una implementación local de EC2/](#)

## AppSpec sección «ganchos» para una implementación de Amazon ECS

### Temas

- [Lista de enlaces de eventos de ciclo de vida para una implementación de Amazon ECS](#)
- [Ejecute el orden de los enlaces en una implementación de Amazon ECS.](#)
- [Estructura de la sección "hooks"](#)
- [Ejemplo de función de "hooks" de Lambda](#)

### Lista de enlaces de eventos de ciclo de vida para una implementación de Amazon ECS

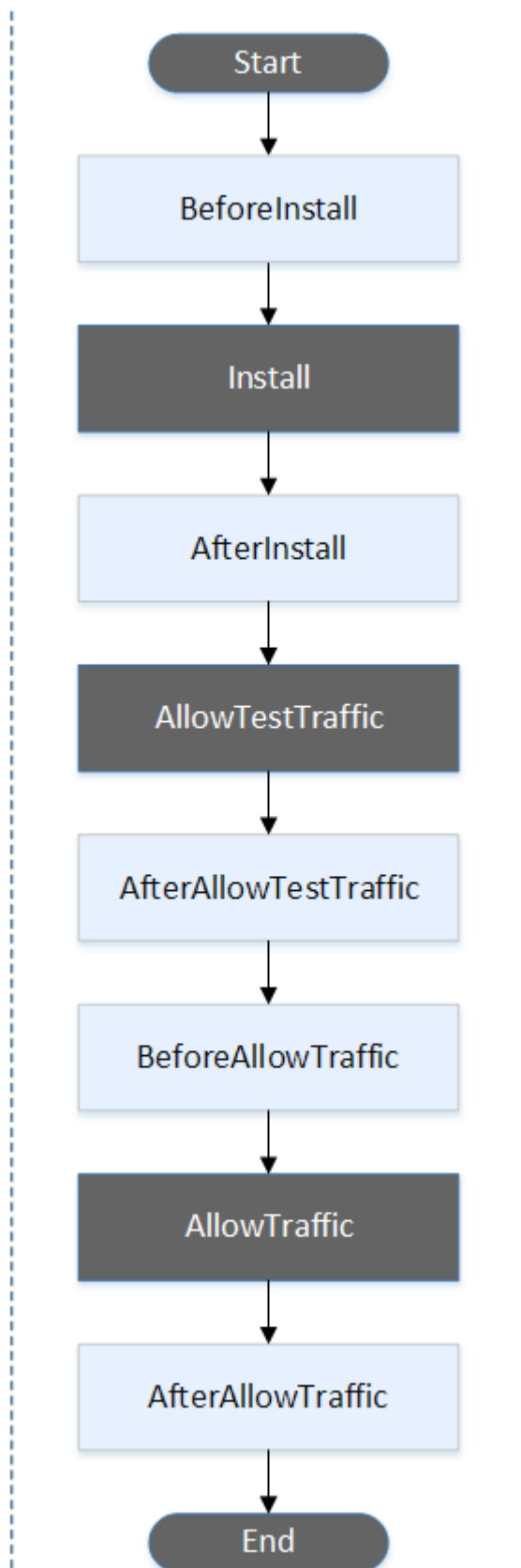
Un enlace de AWS Lambda es una función de Lambda especificada con una cadena en una línea nueva después del nombre del evento del ciclo de vida. Cada enlace se ejecuta una vez por implementación. A continuación, se muestran las descripciones de los eventos del ciclo de vida donde puede ejecutar un enlace durante una implementación de Amazon ECS.

- **BeforeInstall:** se utiliza para ejecutar tareas antes de crear el conjunto de tareas de sustitución. Un grupo de destino se asocia al conjunto de tareas original. Si se especifica un agente de escucha de prueba opcional, se asocia al conjunto de tareas original. En este momento no es posible una restauración.
- **AfterInstall:** se utiliza para ejecutar tareas una vez que el conjunto de tareas se ha creado y uno de los grupos de destino se ha asociado con él. Si se especifica un agente de escucha de prueba opcional, se asocia al conjunto de tareas original. El resultado de la función de enlace en este evento de ciclo de vida puede activar una restauración.
- **AfterAllowTestTraffic:** se utiliza para ejecutar tareas después de que el oyente de prueba envía tráfico al conjunto de tareas de sustitución. El resultado de la función de enlace en este punto puede activar una restauración.
- **BeforeAllowTraffic:** se utiliza para ejecutar tareas una vez que el segundo grupo de destino se ha asociado con el conjunto de tareas de sustitución, pero antes de que se envíe tráfico al conjunto de tareas de sustitución. El resultado de la función de enlace en este evento de ciclo de vida puede activar una restauración.
- **AfterAllowTraffic:** se utiliza para ejecutar tareas después de que el segundo grupo de destino envíe tráfico al conjunto de tareas de sustitución. El resultado de la función de enlace en este evento de ciclo de vida puede activar una restauración.

Para obtener más información, consulte [¿Qué sucede durante una implementación de Amazon ECS?](#) y [Tutorial: Implementación de un servicio de Amazon ECS con una prueba de validación](#).

Ejecute el orden de los enlaces en una implementación de Amazon ECS.

En una implementación de Amazon ECS, los enlaces de eventos se ejecutan en el siguiente orden:



**Note**

Los eventos de inicio TestTrafficAllowTraffic, instalación y finalización de la implementación no se pueden programar, por lo que aparecen en gris en este diagrama.

## Estructura de la sección "hooks"

Los siguientes ejemplos muestran la estructura de la sección 'hooks'.

Mediante YAML:

Hooks:

- BeforeInstall: *"BeforeInstallHookFunctionName"*
- AfterInstall: *"AfterInstallHookFunctionName"*
- AfterAllowTestTraffic: *"AfterAllowTestTrafficHookFunctionName"*
- BeforeAllowTraffic: *"BeforeAllowTrafficHookFunctionName"*
- AfterAllowTraffic: *"AfterAllowTrafficHookFunctionName"*

Mediante JSON:

```
"Hooks": [
 {
 "BeforeInstall": "BeforeInstallHookFunctionName"
 },
 {
 "AfterInstall": "AfterInstallHookFunctionName"
 },
 {
 "AfterAllowTestTraffic": "AfterAllowTestTrafficHookFunctionName"
 },
 {
 "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
 },
 {
 "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
 }
]
```

## Ejemplo de función de “hooks” de Lambda

Utilice la 'hooks' sección para especificar una función de Lambda a la que CodeDeploy se pueda llamar para validar una implementación de Amazon ECS. Puede usar la misma función o una diferente para los eventos del BeforeInstall ciclo de vida de la AfterAllowTraffic implementación y AfterInstall AfterAllowTestTrafficBeforeAllowTraffic, Tras completar las pruebas de validación, la AfterAllowTraffic función Lambda vuelve a llamar CodeDeploy y entrega un resultado de Succeeded o. Failed

### Important

Se considera que el despliegue ha fallado si CodeDeploy la función de validación de Lambda no lo notifica en el plazo de una hora.

Antes de invocar una función de enlace de Lambda, el servidor debe recibir una notificación con el ID de implementación y el ID de ejecución del enlace de evento del ciclo de vida con el comando `putLifecycleEventHookExecutionStatus`.

A continuación, se muestra una función de enlace de Lambda de ejemplo escrita en Node.js.

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
 //Read the DeploymentId from the event payload.
 var deploymentId = event.DeploymentId;

 //Read the LifecycleEventHookExecutionId from the event payload
 var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

 /*
 Enter validation tests here.
 */

 // Prepare the validation test results with the deploymentId and
 // the lifecycleEventHookExecutionId for CodeDeploy.
 var params = {
 deploymentId: deploymentId,
 lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
```



```
 status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
 };

 // Pass CodeDeploy the prepared validation test results.
 codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
 if (err) {
 // Validation failed.
 callback('Validation test failed');
 } else {
 // Validation succeeded.
 callback(null, 'Validation test succeeded');
 }
 });
};
```

AppSpec sección de «ganchos» para una implementación de AWS Lambda

## Temas

- [Lista de enlaces de eventos del ciclo de vida para una implementación de AWS Lambda](#)
- [Orden de ejecución de los enlaces en una implementación de versiones de funciones de Lambda](#)
- [Estructura de la sección "hooks"](#)
- [Ejemplo de función de "hooks" de Lambda](#)

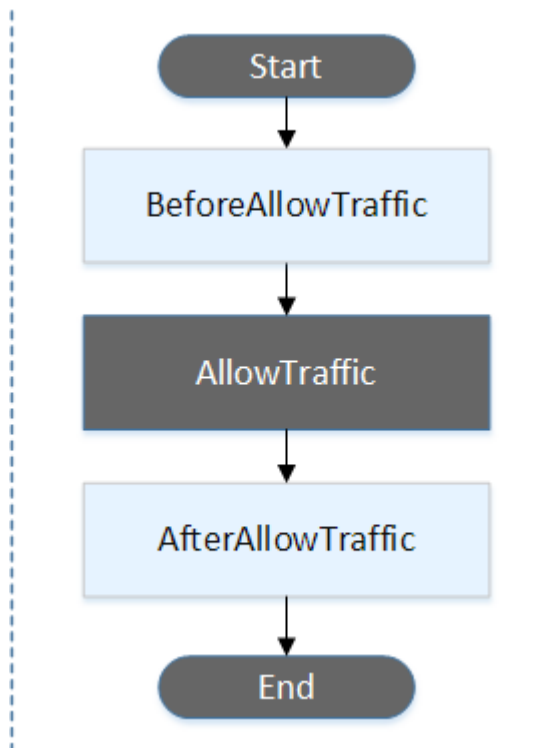
Lista de enlaces de eventos del ciclo de vida para una implementación de AWS Lambda

Un enlace de AWS Lambda es una función de Lambda especificada con una cadena en una línea nueva después del nombre del evento del ciclo de vida. Cada enlace se ejecuta una vez por implementación. Estas son las descripciones de los ganchos disponibles para su AppSpec uso en el archivo.

- **BeforeAllowTraffic**— Se utiliza para ejecutar tareas antes de que el tráfico se traslade a la versión de la función Lambda implementada.
- **AfterAllowTraffic**— Se utiliza para ejecutar tareas después de que todo el tráfico se haya desplazado a la versión de la función Lambda implementada.

Orden de ejecución de los enlaces en una implementación de versiones de funciones de Lambda

En una implementación de versiones de funciones de Lambda sin servidor, los enlaces de eventos se ejecutan en el siguiente orden:

**Note**

Los eventos de inicio y finalización de la implementación no se pueden programar, por lo que aparecen en gris en este diagrama. AllowTraffic

### Estructura de la sección "hooks"

Los siguientes ejemplos muestran la estructura de la sección "hooks".

Mediante YAML:

```
hooks:
 - BeforeAllowTraffic: BeforeAllowTrafficHookFunctionName
 - AfterAllowTraffic: AfterAllowTrafficHookFunctionName
```

Mediante JSON:

```
"hooks": [{
 "BeforeAllowTraffic": "BeforeAllowTrafficHookFunctionName"
},
```

```
{
 "AfterAllowTraffic": "AfterAllowTrafficHookFunctionName"
}]
```

## Ejemplo de función de “hooks” de Lambda

Utilice la sección «hooks» para especificar una función de Lambda a la que CodeDeploy se pueda llamar para validar una implementación de Lambda. Puede usar la misma función o una diferente para los eventos del ciclo de vida de la `AfterAllowTraffic` implementación `BeforeAllowTraffic` y para los eventos del ciclo de vida. Tras completar las pruebas de validación, la función de validación de Lambda vuelve a llamar CodeDeploy y entrega un resultado de `Succeeded` o `Failed`.

### Important

Se considera que el despliegue ha fallado si CodeDeploy la función de validación de Lambda no lo notifica en el plazo de una hora.

Antes de invocar una función de enlace de Lambda, el servidor debe recibir una notificación con el ID de implementación y el ID de ejecución del enlace de evento del ciclo de vida con el comando `putLifecycleEventHookExecutionStatus`.

A continuación, se muestra una función de enlace de Lambda de ejemplo escrita en Node.js.

```
'use strict';

const aws = require('aws-sdk');
const codedeploy = new aws.CodeDeploy({apiVersion: '2014-10-06'});

exports.handler = (event, context, callback) => {
 //Read the DeploymentId from the event payload.
 var deploymentId = event.DeploymentId;

 //Read the LifecycleEventHookExecutionId from the event payload
 var lifecycleEventHookExecutionId = event.LifecycleEventHookExecutionId;

 /*
 Enter validation tests here.
 */
```

```
// Prepare the validation test results with the deploymentId and
// the lifecycleEventHookExecutionId for CodeDeploy.
var params = {
 deploymentId: deploymentId,
 lifecycleEventHookExecutionId: lifecycleEventHookExecutionId,
 status: 'Succeeded' // status can be 'Succeeded' or 'Failed'
};

// Pass CodeDeploy the prepared validation test results.
codedeploy.putLifecycleEventHookExecutionStatus(params, function(err, data) {
 if (err) {
 // Validation failed.
 callback('Validation test failed');
 } else {
 // Validation succeeded.
 callback(null, 'Validation test succeeded');
 }
});
};
```

## AppSpec Sección de «enganches» para una implementación local de EC2/

### Temas

- [Lista de enlaces de eventos del ciclo de vida](#)
- [Disponibilidad de los enlaces de eventos del ciclo de vida](#)
- [Orden de ejecución de los enlaces en una implementación](#)
- [Estructura de la sección "hooks"](#)
- [Hacer referencia a los archivos en los scripts de enlaces](#)
- [Disponibilidad de las variables de entorno para los enlaces](#)
- [Ejemplo de enlaces](#)


### Lista de enlaces de eventos del ciclo de vida

Un enlace de implementación de EC2/en las instalaciones se ejecuta una vez por cada implementación en una instancia. Puede especificar uno o varios scripts para ejecutar en un enlace. Cada enlace de un evento del ciclo de vida se especifica con una cadena en una línea independiente. Estas son las descripciones de los ganchos disponibles para su uso en el archivo.

### AppSpec

Para obtener información sobre los enlaces de eventos del ciclo de vida válidos para cada tipo de implementación y restauración, consulte [Disponibilidad de los enlaces de eventos del ciclo de vida](#).

- **ApplicationStop**: este evento de ciclo de vida de la implementación se produce incluso antes de que se descargue la revisión de la aplicación. Puede especificar scripts para este evento para detener con fluidez la aplicación o para eliminar los paquetes instalados actualmente en la preparación de una implementación. El AppSpec archivo y los scripts utilizados para este evento del ciclo de vida de la implementación provienen de la revisión anterior de la aplicación que se implementó correctamente.

 Note

No existe ningún AppSpec archivo en una instancia antes de realizar la implementación en ella. Por este motivo, el enlace `ApplicationStop` no se ejecuta la primera vez que se realiza la implementación en la instancia. Puede utilizar el enlace `ApplicationStop` la segunda vez que realice la implementación en una instancia.

Para determinar la ubicación de la última revisión de la aplicación implementada correctamente, el CodeDeploy agente busca la ubicación que aparece en el `deployment-group-id_last_successful_install` archivo. Este archivo se encuentra en:

Carpeta `/opt/codedeploy-agent/deployment-root/deployment-instructions` en las instancias de Amazon EC2 en Amazon Linux, Ubuntu Server y RHEL.

Carpeta `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions` en las instancias de Amazon EC2 en Windows Server.

Para solucionar un error de una implementación durante el evento del ciclo de vida de la implementación `ApplicationStop`, consulte [Solución de problemas relacionados con un error o un evento del ciclo de vida de `ApplicationStop` la `BeforeBlockTraffic` implementación `AfterBlockTraffic`](#).

- **DownloadBundle**— Durante este evento del ciclo de vida de la implementación, el CodeDeploy agente copia los archivos de revisión de la aplicación en una ubicación temporal:

Carpeta `/opt/codedeploy-agent/deployment-root/deployment-group-id/deployment-id/deployment-archive` en las instancias de Amazon EC2 en Amazon Linux, Ubuntu Server y RHEL.

Carpeta C:\ProgramData\Amazon\CodeDeploy\*deployment-group-id*\*deployment-id*\deployment-archive en las instancias de Amazon EC2 en Windows Server.

Este evento está reservado para el CodeDeploy agente y no se puede utilizar para ejecutar scripts.

Para solucionar un error de una implementación durante el evento del ciclo de vida de la implementación DownloadBundle, consulte [Solución de problemas de un evento fallido en el ciclo de vida de una DownloadBundle implementación con UnknownError: no abierto para lectura](#).

- **BeforeInstall**: puede usar este evento de ciclo de vida de la implementación para tareas de preinstalación, como descifrar archivos y crear un backup de la versión actual.
- **Install**— Durante este evento del ciclo de vida de la implementación, el CodeDeploy agente copia los archivos de revisión de la ubicación temporal a la carpeta de destino final. Este evento está reservado para el CodeDeploy agente y no se puede utilizar para ejecutar scripts.
- **AfterInstall**: puede usar este evento de ciclo de vida de la implementación para configurar la aplicación o para cambiar los permisos de los archivos.
- **ApplicationStart**: este evento de ciclo de vida de la implementación se utiliza normalmente para reiniciar los servicios que se detuvieron durante **ApplicationStop**.
- **ValidateService**: este es el último evento de ciclo de vida de la implementación. Se utiliza para verificar que la implementación se ha completado correctamente.
- **BeforeBlockTraffic**: puede utilizar este evento de ciclo de vida de la implementación para ejecutar tareas en las instancias antes de que se cancele su registro de un equilibrador de carga.

Para solucionar un error de una implementación durante el evento del ciclo de vida de la implementación **BeforeBlockTraffic**, consulte [Solución de problemas relacionados con un error o un evento del ciclo de vida de \*\*ApplicationStop\*\* la \*\*BeforeBlockTraffic\*\* implementación \*\*AfterBlockTraffic\*\*](#).

- **BlockTraffic**: durante este evento de ciclo de vida de la implementación, se bloquea el tráfico de Internet a las instancias que están actualmente sirviendo tráfico. Este evento está reservado para el CodeDeploy agente y no se puede usar para ejecutar scripts.
- **AfterBlockTraffic**: puede utilizar este evento de ciclo de vida de la implementación para ejecutar tareas en las instancias después de que se cancele su equilibrador de carga.

Para solucionar un error de una implementación durante el evento del ciclo de vida de la implementación **AfterBlockTraffic**, consulte [Solución de problemas relacionados con un](#)

[error o un evento del ciclo de vida de ApplicationStop la BeforeBlockTraffic implementación AfterBlockTraffic](#) .

- **BeforeAllowTraffic:** puede utilizar este evento de ciclo de vida de la implementación para ejecutar tareas en las instancias antes de que se registren con un equilibrador de carga.
- **AllowTraffic:** durante este evento de ciclo de vida de la implementación, se permite el acceso del tráfico de Internet a las instancias después de una implementación. Este evento está reservado para el CodeDeploy agente y no se puede usar para ejecutar scripts.
- **AfterAllowTraffic:** puede utilizar este evento de ciclo de vida de la implementación para ejecutar tareas en las instancias después de que se registren con un equilibrador de carga.

### Disponibilidad de los enlaces de eventos del ciclo de vida

En la siguiente tabla se indican los enlaces de eventos del ciclo de vida disponibles para cada escenario de implementación y restauración.

| Nombre de evento del ciclo de vida | Implementación de lanzamiento de Auto Scaling <sup>1</sup> | Implementación de terminación de Auto Scaling <sup>1</sup> | Implementación local <sup>1</sup> | Implementación blue/green: instancias originales | Implementación blue/green: instancias de sustitución | Restauración de implementación blue/green: instancias originales | Restauración de implementación blue/green: instancias de sustitución |
|------------------------------------|------------------------------------------------------------|------------------------------------------------------------|-----------------------------------|--------------------------------------------------|------------------------------------------------------|------------------------------------------------------------------|----------------------------------------------------------------------|
| ApplicationStop                    | ✓                                                          | ✓                                                          | ✓                                 |                                                  | ✓                                                    |                                                                  |                                                                      |
| DownloadBundle <sup>3</sup>        | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    |                                                                  |                                                                      |
| BeforeInstall                      | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    |                                                                  |                                                                      |
| Install <sup>2</sup>               | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    |                                                                  |                                                                      |

| Nombre de evento del ciclo de vida | Implementación de lanzamiento de Auto Scaling <sup>1</sup> | Implementación de terminación de Auto Scaling <sup>1</sup> | Implementación local <sup>1</sup> | Implementación blue/green: instancias originales | Implementación blue/green: instancias de sustitución | Restauración de implementación blue/green: instancias originales | Restauración de implementación blue/green: instancias de sustitución |
|------------------------------------|------------------------------------------------------------|------------------------------------------------------------|-----------------------------------|--------------------------------------------------|------------------------------------------------------|------------------------------------------------------------------|----------------------------------------------------------------------|
| AfterInstall                       | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    |                                                                  |                                                                      |
| ApplicationStart                   | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    |                                                                  |                                                                      |
| ValidateService                    | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    |                                                                  |                                                                      |
| BeforeBlockTraffic                 |                                                            | ✓                                                          | ✓                                 | ✓                                                |                                                      |                                                                  | ✓                                                                    |
| BlockTraffic <sup>3</sup>          |                                                            | ✓                                                          | ✓                                 | ✓                                                |                                                      |                                                                  | ✓                                                                    |
| AfterBlockTraffic                  |                                                            | ✓                                                          | ✓                                 | ✓                                                |                                                      |                                                                  | ✓                                                                    |
| BeforeAllowTraffic                 | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    | ✓                                                                |                                                                      |
| AllowTraffic <sup>3</sup>          | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    | ✓                                                                |                                                                      |
| AfterAllowTraffic                  | ✓                                                          |                                                            | ✓                                 |                                                  | ✓                                                    | ✓                                                                |                                                                      |



| Nombre de evento del ciclo de vida | Implementación de lanzamiento de Auto Scaling <sup>1</sup> | Implementación de terminación de Auto Scaling <sup>1</sup> | Implementación local <sup>1</sup> | Implementación blue/green: instancias originales | Implementación blue/green: instancias de sustitución | Restauración de implementación blue/green: instancias originales | Restauración de implementación blue/green: instancias de sustitución |
|------------------------------------|------------------------------------------------------------|------------------------------------------------------------|-----------------------------------|--------------------------------------------------|------------------------------------------------------|------------------------------------------------------------------|----------------------------------------------------------------------|
|------------------------------------|------------------------------------------------------------|------------------------------------------------------------|-----------------------------------|--------------------------------------------------|------------------------------------------------------|------------------------------------------------------------------|----------------------------------------------------------------------|

<sup>1</sup> Para obtener más información sobre implementaciones de Amazon EC2 Auto Scaling, consulte [Cómo funciona Amazon EC2 Auto Scaling con CodeDeploy](#).

<sup>2</sup> Se aplica también a la reversión de una implementación local.

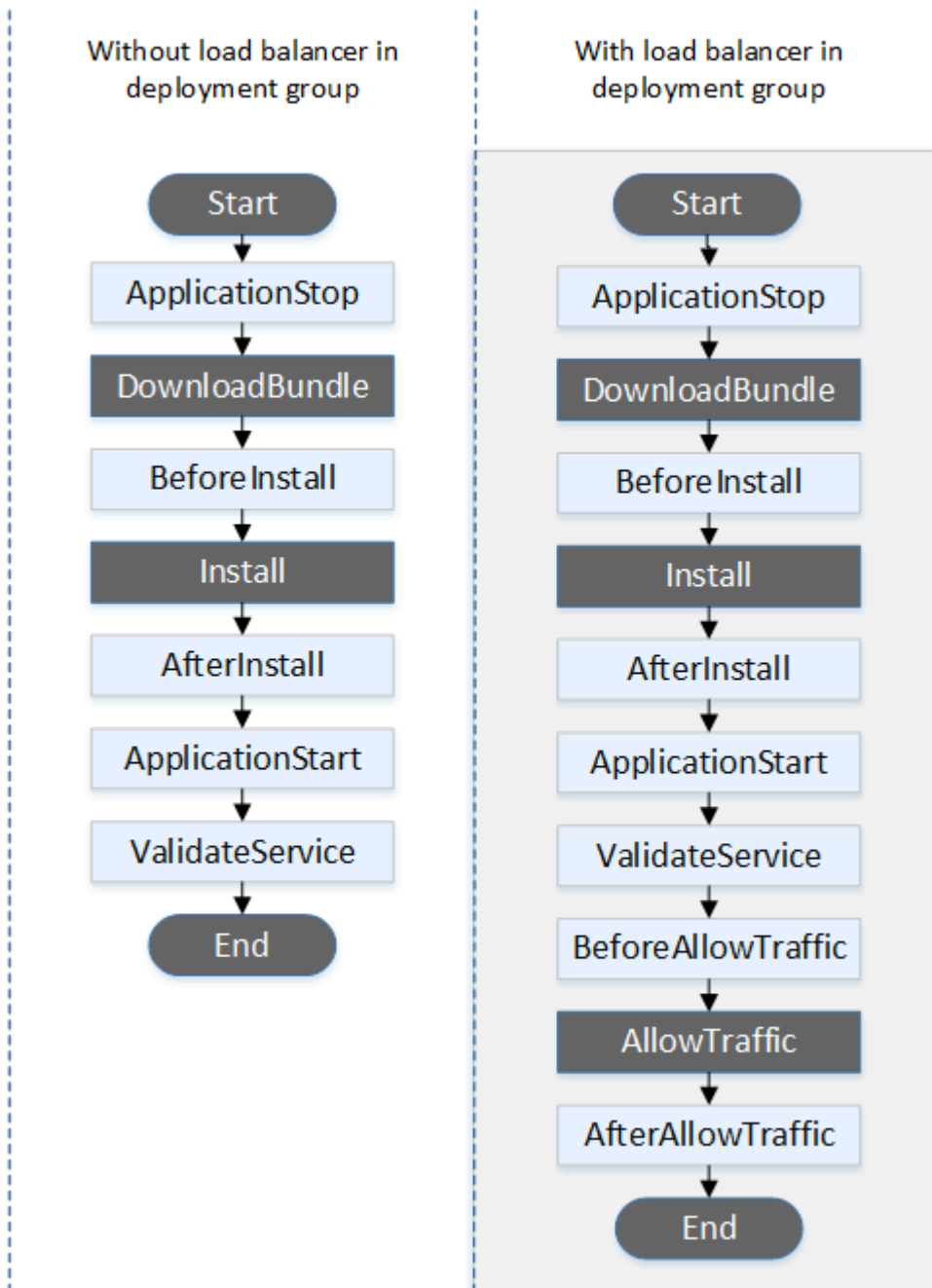
<sup>3</sup> Reservado para CodeDeploy operaciones. No se puede utilizar para ejecutar scripts.

## Orden de ejecución de los enlaces en una implementación

### Implementaciones de lanzamiento de Auto Scaling

Durante una implementación de lanzamiento de Auto Scaling, CodeDeploy ejecuta los enlaces de eventos en el siguiente orden.

Para obtener más información sobre implementaciones de lanzamiento Auto Scaling, consulte [Cómo funciona Amazon EC2 Auto Scaling con CodeDeploy](#).



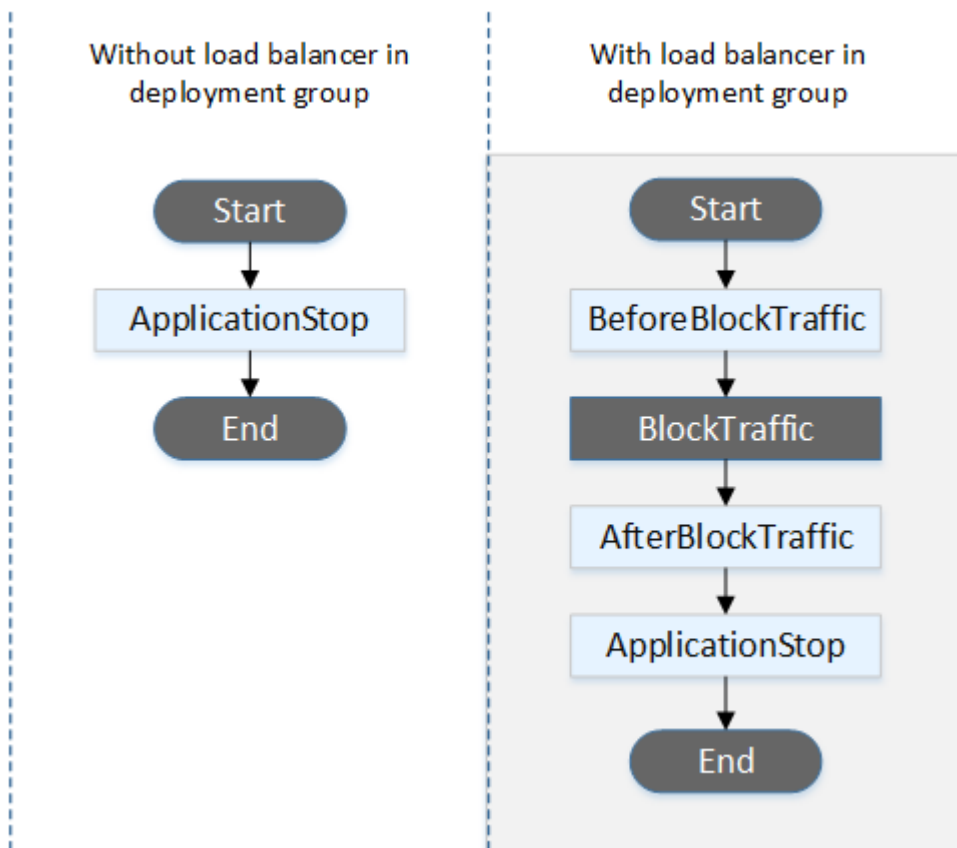
### Note

Los eventos de inicio `DownloadBundleAllowTraffic`, instalación y finalización de la implementación no se pueden programar, por lo que aparecen en gris en este diagrama. Sin embargo, puede editar la 'files' sección del AppSpec archivo para especificar qué se instalará durante el evento de instalación.

## Implementaciones de terminación de Auto Scaling

Durante un despliegue de terminación de Auto Scaling, CodeDeploy ejecuta los enlaces de eventos en el siguiente orden.

Para obtener más información sobre implementaciones de terminación de Auto Scaling, consulte [Habilitación de implementaciones de terminación durante los eventos de reducción horizontal de Auto Scaling](#).




### **i** Note

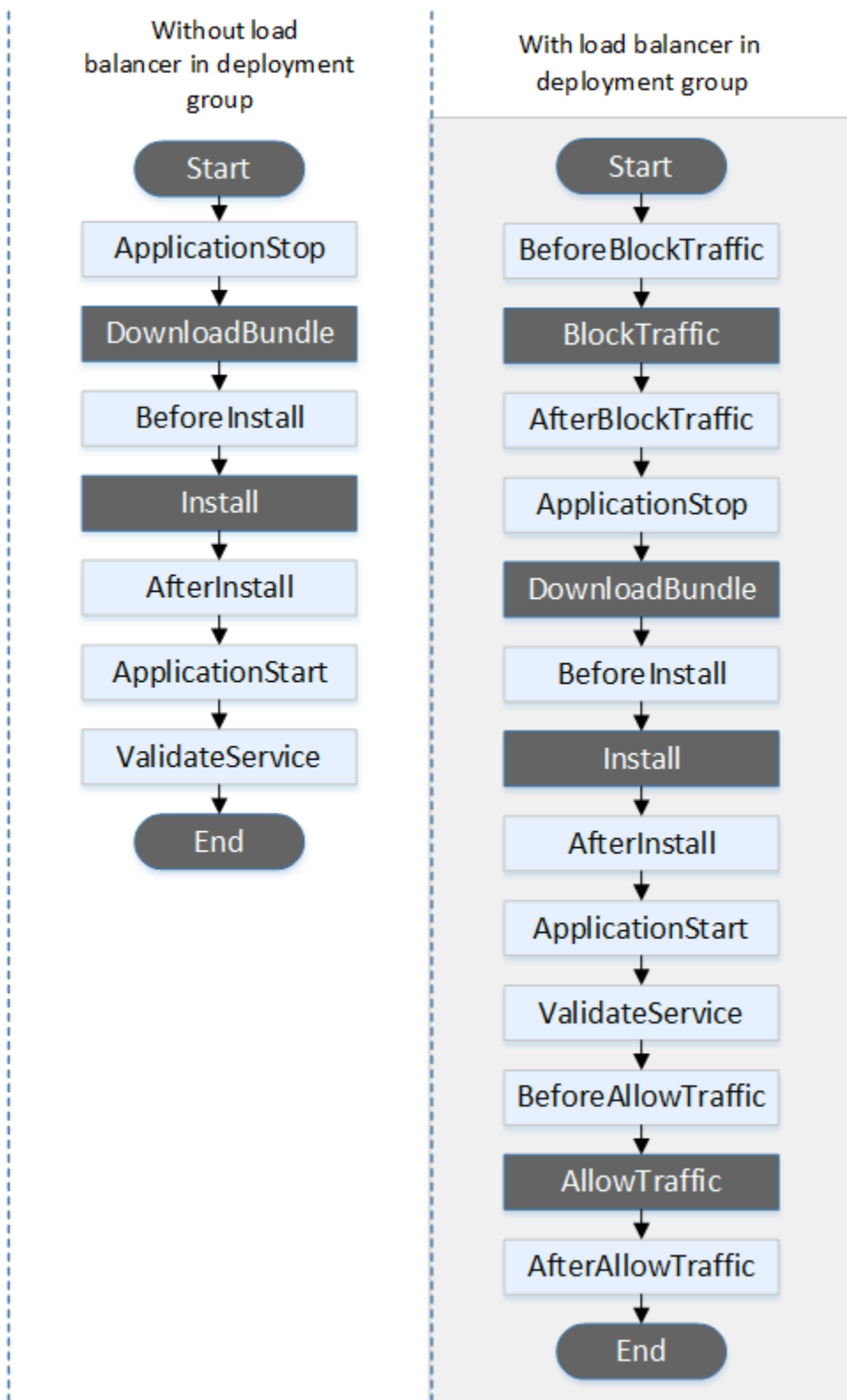
Los eventos de inicio y finalización de la implementación no se pueden programar, por lo que aparecen en gris en este diagrama. BlockTraffic

## Implementaciones in situ

En una implementación in situ, incluida la restauración de una implementación in situ, los enlaces de eventos se ejecutan en el orden siguiente:

 **Note**

En el caso de las implementaciones locales, los seis enlaces relacionados con el bloqueo y el permiso del tráfico solo se aplican si se ha especificado un equilibrador de carga clásico, un equilibrador de carga de aplicación o un equilibrador de carga de red de Elastic Load Balancing en el grupo de implementación.

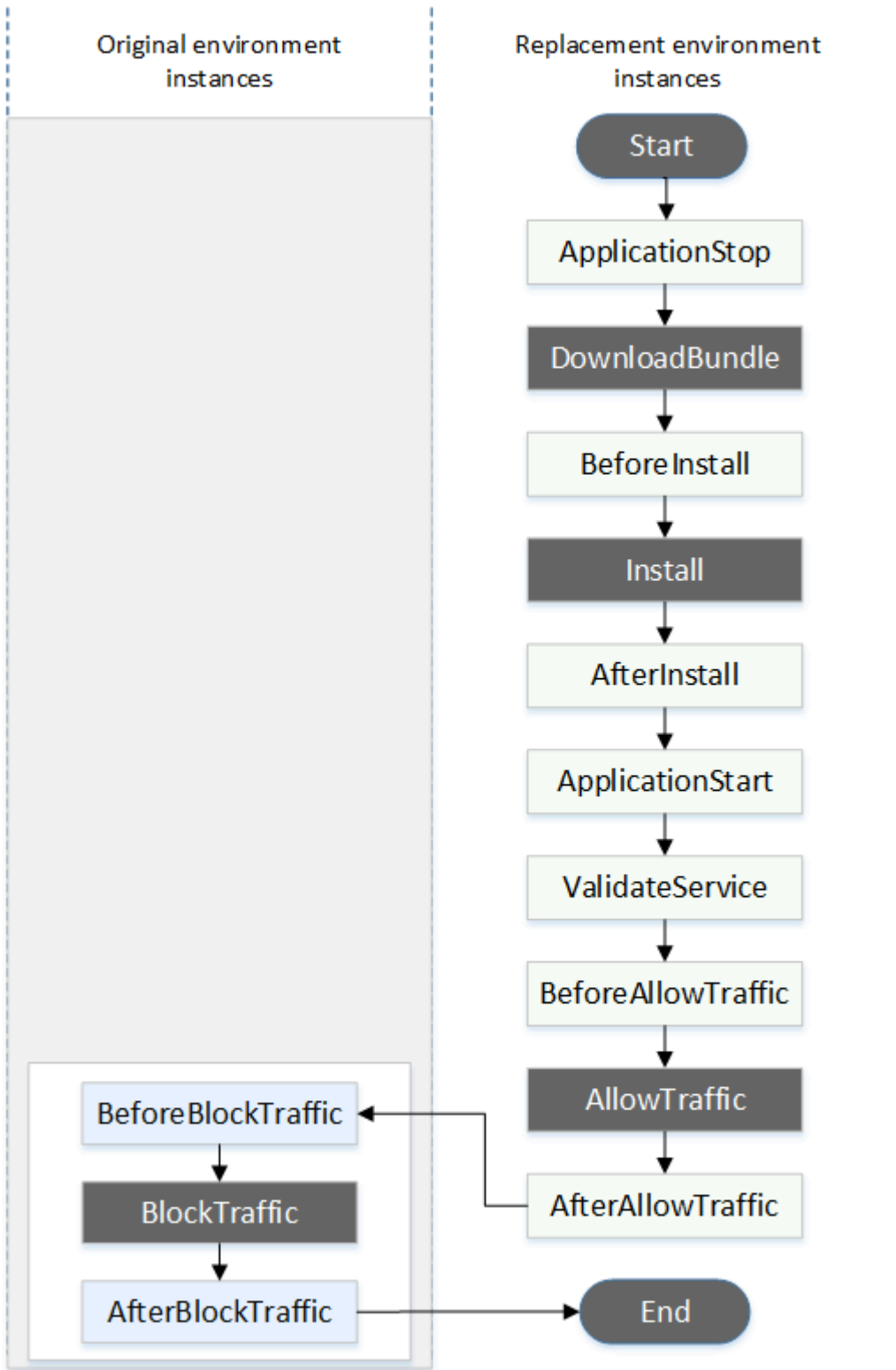


**Note**

Los eventos de inicio DownloadBundle, instalación y finalización de la implementación no se pueden programar, por lo que aparecen en gris en este diagrama. Sin embargo, puede editar la 'files' sección del AppSpec archivo para especificar qué se instalará durante el evento de instalación.

## Implementaciones blue/green

En una implementación blue/green, los enlaces de eventos se ejecutan en el siguiente orden:



**Note**

Los eventos de inicio `DownloadBundleBlockTrafficAllowTraffic`, instalación y finalización de la implementación no se pueden programar, por lo que aparecen en gris en este diagrama. Sin embargo, puede editar la sección de «archivos» del AppSpec archivo para especificar qué se instalará durante el evento de instalación.

## Estructura de la sección "hooks"

La sección 'hooks' tiene la siguiente estructura:

```
hooks:
 deployment-lifecycle-event-name:
 - location: script-location
 timeout: timeout-in-seconds
 runas: user-name
```

Puede incluir los siguientes elementos en una entrada hook detrás del nombre de evento del ciclo de vida de la implementación:

### location

Obligatorio. La ubicación en el paquete del archivo de script de la revisión. La ubicación de los scripts que especifica en la sección hooks es relativa a la raíz del paquete de revisión de la aplicación. Para obtener más información, consulte [Planifique una revisión para CodeDeploy](#).

### timeout

Opcional. El número de segundos que se puede ejecutar el script antes de que se considere que se ha producido un error. El valor predeterminado es 3 600 segundos (1 hora).

**Note**

3 600 segundos (1 hora) es la cantidad máxima de tiempo de ejecución permitido para el script para cada evento del ciclo de vida de la implementación. Si el script supera este límite, la implementación se detiene y la implementación en la instancia da un error. Asegúrese de que el número total de segundos especificados en timeout para todos los scripts en cada evento del ciclo de vida de la implementación no supere este límite.



## runas

Opcional. El usuario que se va a suplantar al ejecutar el script. De forma predeterminada, este es el CodeDeploy agente que se ejecuta en la instancia. CodeDeploy no almacena contraseñas, por lo que no se puede suplantar al usuario si el usuario runas necesita una contraseña. Este ejemplo se aplica únicamente a las instancias de Amazon Linux y Ubuntu Server.

### Hacer referencia a los archivos en los scripts de enlaces

Si va a conectar un script a un evento CodeDeploy del ciclo de vida, tal y como se describe en [AppSpec sección de «ganchos»](#), y desea hacer referencia a un archivo (por ejemplo `helper.sh`) del script, tendrá que especificarlo mediante: `helper.sh`

- (Recomendado) Una ruta absoluta. Consulte [Uso de rutas absolutas](#).
- Una ruta relativa. Consulte [Uso de rutas relativas](#).

### Uso de rutas absolutas

Para hacer referencia a un archivo mediante su ruta absoluta, puede hacer lo siguiente:

- Especifique la ruta absoluta en la `files` sección del AppSpec archivo, en la `destination` propiedad. A continuación, especifique la misma ruta absoluta en el script de enlace. Para obtener más información, consulte [AppSpec Sección de «archivos» \(solo para despliegues de EC2/on-premise\)](#).
- Especifique una ruta absoluta dinámica en el script de enlace. Para obtener más información, consulte [Ubicación del archivo de implementación](#).

### Ubicación del archivo de implementación

Durante el evento [DownloadBundle](#) del ciclo de vida, el CodeDeploy agente extrae la [revisión](#) para el despliegue en un directorio que tiene el siguiente formato:

*root-directory/deployment-group-id/deployment-id*/deployment-archive

La parte del *directorio raíz* de la ruta siempre se establece en el valor predeterminado que se muestra en la siguiente tabla o se controla mediante el ajuste de configuración `:root_dir`. Para obtener más información acerca de los ajustes de configuración, consulte [CodeDeploy referencia de configuración del agente](#).

| Plataforma de agentes                       | Directorio raíz predeterminado        |
|---------------------------------------------|---------------------------------------|
| Linux: todas las distribuciones rpm         | /opt/codedeploy-agent/deployment-root |
| Ubuntu Server: todas las distribuciones deb | /opt/codedeploy-agent/deployment-root |
| Windows Server                              | %ProgramData%\Amazon\CodeDeploy       |

Desde sus scripts de enlace, puede acceder al archivo de implementación actual utilizando la ruta del directorio raíz y las variables de entorno `DEPLOYMENT_ID` y `DEPLOYMENT_GROUP_ID`. Para obtener más información acerca de las que puede usar, consulte [Disponibilidad de las variables de entorno para los enlaces](#).

Por ejemplo, así es como puede acceder a un archivo `data.json` que se encuentra en la raíz de su revisión en Linux:

```
#!/bin/bash

rootDirectory="/opt/codedeploy-agent/deployment-root" # note: this will be different if
you
customize the :root_dir
configuration
dataFile="$rootDirectory/$DEPLOYMENT_GROUP_ID/$DEPLOYMENT_ID/deployment-archive/
data.json"
data=$(cat dataFile)
```

Otro ejemplo: así es como puede acceder a un archivo `data.json` que se encuentra en la raíz de la revisión con Powershell en Windows:

```
$rootDirectory="$env:ProgramData\Amazon\CodeDeploy" # note: this will be different if
you
customize the :root_dir
configuration
$dataFile="$rootDirectory\$env:DEPLOYMENT_GROUP_ID\$env:DEPLOYMENT_ID\deployment-
archive\data.json"
$data=(Get-Content $dataFile)
```

## Uso de rutas relativas

Para hacer referencia a un archivo mediante su ruta relativa, necesitará conocer el directorio de trabajo del CodeDeploy agente. Las rutas de los archivos son relativas a este directorio.

La siguiente tabla muestra el directorio de trabajo de cada plataforma compatible del CodeDeploy agente.

| Plataforma de agente                           | Método de administración de procesos    | Directorio de trabajo para los scripts de eventos de ciclo de vida |
|------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------|
| Linux: todas las distribuciones rpm            | systemd (predeterminado)                | /                                                                  |
|                                                | init.d: <a href="#">más información</a> | /opt/codedeploy-agent                                              |
| Ubuntu Server: todas las distribuciones debian | Todos                                   | /opt/codedeploy-agent                                              |
| Windows Server                                 | no aplicable                            | C:\Windows\System32                                                |

## Disponibilidad de las variables de entorno para los enlaces

Durante cada evento del ciclo de vida de la implementación, los scripts de enlace pueden tener acceso a las siguientes variables de entorno:

### APPLICATION\_NAME

El nombre de la aplicación CodeDeploy que forma parte de la implementación actual (por ejemplo,WordPress\_App).

### DEPLOYMENT\_ID

El ID se CodeDeploy ha asignado a la implementación actual (por ejemplo,d-AB1CDEF23).

### DEPLOYMENT\_GROUP\_NAME

El nombre del grupo de despliegues CodeDeploy que forma parte del despliegue actual (por ejemplo,WordPress\_DepGroup).

## DEPLOYMENT\_GROUP\_ID

El ID del grupo de despliegues CodeDeploy que forma parte de la implementación actual (por ejemplo, b1a2189b-dd90-4ef5-8f40-4c1c5EXAMPLE).

## LIFECYCLE\_EVENT

El nombre del evento del ciclo de vida de la implementación actual (por ejemplo, `AfterInstall`).

Estas variables de entorno son locales con respecto a cada evento del ciclo de vida de la implementación.

Hay variables de entorno adicionales disponibles para conectar scripts de enlace en función del origen del paquete de implementación:

### Paquete de Amazon S3

- BUNDLE\_BUCKET

El nombre del bucket de Amazon S3 desde el que se descargó el paquete de implementación (por ejemplo, `my-s3-bucket`).

- BUNDLE\_KEY

La clave de objeto del paquete descargado dentro del bucket de Amazon S3 (por ejemplo, `WordPress_App.zip`).

- BUNDLE\_VERSION

La versión del objeto del paquete (por ejemplo, `3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo`). Esta variable solo se establece si el bucket de Amazon S3 tiene activado el [control de versiones de objetos](#).

- BUNDLE\_ETAG

La ETag del objeto del paquete (por ejemplo, `b10a8db164e0754105b7a99be72e3fe5-4`).

### Paquete de GitHub

- BUNDLE\_COMMIT

El hash de confirmación SHA256 del paquete generado por Git (por ejemplo, d2a84f4b8b650937ec8f73cd8be2c74add5a911ba64df27458ed8229da804a26).

El script siguiente cambia el puerto de escucha de un servidor HTTP Apache a 9090 en lugar de a 80 si el valor de `DEPLOYMENT_GROUP_NAME` es igual a `Staging`. Este script debe invocarse durante el evento del ciclo de vida de implementación `BeforeInstall`:

```
if ["$DEPLOYMENT_GROUP_NAME" == "Staging"]
then
 sed -i -e 's/Listen 80/Listen 9090/g' /etc/httpd/conf/httpd.conf
fi
```

El script de ejemplo siguiente cambia el nivel de detalle de los mensajes registrados en su log de errores de advertencia a depurar si el valor de la variable de entorno `DEPLOYMENT_GROUP_NAME` es igual a `Staging`. Este script debe invocarse durante el evento del ciclo de vida de implementación `BeforeInstall`:

```
if ["$DEPLOYMENT_GROUP_NAME" == "Staging"]
then
 sed -i -e 's/LogLevel warn/LogLevel debug/g' /etc/httpd/conf/httpd.conf
fi
```

El script de ejemplo siguiente sustituye el texto de la página web especificada por el texto que muestra el valor de estas variables de entorno. Este script debe invocarse durante el evento del ciclo de vida de implementación `AfterInstall`:

```
#!/usr/bin/python

import os

strToSearch("<h2>This application was deployed using CodeDeploy.</h2>")
strToReplace("<h2>This page for "+os.environ['APPLICATION_NAME']+"
application and "+os.environ['DEPLOYMENT_GROUP_NAME']+" deployment group with
"+os.environ['DEPLOYMENT_GROUP_ID']+" deployment group ID was generated by a
"+os.environ['LIFECYCLE_EVENT']+" script during "+os.environ['DEPLOYMENT_ID']+"
deployment.</h2>")

fp=open("/var/www/html/index.html","r")
buffer=fp.read()
```

```
fp.close()

fp=open("/var/www/html/index.html","w")
fp.write(buffer.replace(strToSearch,strToReplace))
fp.close()
```

## Ejemplo de enlaces

A continuación se muestra un ejemplo de una entrada hooks (enlaces) que especifica dos enlaces para el evento de ciclo de vida `AfterInstall`.

```
hooks:
 AfterInstall:
 - location: Scripts/RunResourceTests.sh
 timeout: 180
 - location: Scripts/PostDeploy.sh
 timeout: 180
```

El script `Scripts/RunResourceTests.sh` se ejecuta durante la fase `AfterInstall` del proceso de implementación. La implementación no tendrá éxito si el script tarda más de 180 segundos (3 minutos) en ejecutarse.

La ubicación de los scripts que especifica en la sección "hooks" es relativa a la raíz del paquete de revisión de la aplicación. En el ejemplo anterior, un archivo denominado `RunResourceTests.sh` está en un directorio denominado `Scripts`. El directorio `Scripts` está en la raíz del paquete. Para obtener más información, consulte [Planifique una revisión para CodeDeploy](#).

## AppSpec Ejemplo de archivo

En este tema se proporcionan AppSpec archivos de ejemplo para una implementación AWS Lambda y una implementación local de EC2.

### Temas

- [AppSpec Ejemplo de archivo para una implementación de Amazon ECS](#)
- [AppSpec Ejemplo de archivo para una implementación de AWS Lambda](#)
- [AppSpec Ejemplo de archivo para una implementación local de EC2/](#)

## AppSpec Ejemplo de archivo para una implementación de Amazon ECS

A continuación, se muestra un ejemplo de un AppSpec archivo escrito en YAML para implementar un servicio de Amazon ECS.

```
version: 0.0
Resources:
 - TargetService:
 Type: AWS::ECS::Service
 Properties:
 TaskDefinition: "arn:aws:ecs:us-east-1:111222333444:task-definition/my-task-
definition-family-name:1"
 LoadBalancerInfo:
 ContainerName: "SampleApplicationName"
 ContainerPort: 80
Optional properties
PlatformVersion: "LATEST"
NetworkConfiguration:
 AwsvpcConfiguration:
 Subnets: ["subnet-1234abcd", "subnet-5678abcd"]
 SecurityGroups: ["sg-12345678"]
 AssignPublicIp: "ENABLED"
CapacityProviderStrategy:
 - Base: 1
 CapacityProvider: "FARGATE_SPOT"
 Weight: 2
 - Base: 0
 CapacityProvider: "FARGATE"
 Weight: 1
Hooks:
 - BeforeInstall: "LambdaFunctionToValidateBeforeInstall"
 - AfterInstall: "LambdaFunctionToValidateAfterInstall"
 - AfterAllowTestTraffic: "LambdaFunctionToValidateAfterTestTrafficStarts"
 - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeAllowingProductionTraffic"
 - AfterAllowTraffic: "LambdaFunctionToValidateAfterAllowingProductionTraffic"
```

Esta es una versión del ejemplo anterior escrita en JSON.

```
{
 "version": 0.0,
 "Resources": [
 {
 "TargetService": {
```

```
 "Type": "AWS::ECS::Service",
 "Properties": {
 "TaskDefinition": "arn:aws:ecs:us-east-1:111222333444:task-
definition/my-task-definition-family-name:1",
 "LoadBalancerInfo": {
 "ContainerName": "SampleApplicationName",
 "ContainerPort": 80
 },
 "PlatformVersion": "LATEST",
 "NetworkConfiguration": {
 "AwsvpcConfiguration": {
 "Subnets": [
 "subnet-1234abcd",
 "subnet-5678abcd"
],
 "SecurityGroups": [
 "sg-12345678"
],
 "AssignPublicIp": "ENABLED"
 }
 },
 "CapacityProviderStrategy": [
 {
 "Base" : 1,
 "CapacityProvider" : "FARGATE_SPOT",
 "Weight" : 2
 },
 {
 "Base" : 0,
 "CapacityProvider" : "FARGATE",
 "Weight" : 1
 }
]
 }
 },
},
],
"Hooks": [
 {
 "BeforeInstall": "LambdaFunctionToValidateBeforeInstall"
 },
 {
 "AfterInstall": "LambdaFunctionToValidateAfterInstall"
 },
],
```



```
{
 "AfterAllowTestTraffic": "LambdaFunctionToValidateAfterTestTrafficStarts"
},
{
 "BeforeAllowTraffic":
"LambdaFunctionToValidateBeforeAllowingProductionTraffic"
},
{
 "AfterAllowTraffic":
"LambdaFunctionToValidateAfterAllowingProductionTraffic"
}
]
}
```

A continuación se muestra la secuencia de eventos durante la implementación:

1. Antes de instalar la aplicación de Amazon ECS actualizada en el conjunto de tareas de sustitución, se ejecuta la función de Lambda denominada `LambdaFunctionToValidateBeforeInstall`.
2. Una vez instalada la aplicación de Amazon ECS actualizada en el conjunto de tareas de sustitución, pero antes de recibir tráfico, se ejecuta la función de Lambda denominada `LambdaFunctionToValidateAfterInstall`.
3. Después de que la aplicación de Amazon ECS en el conjunto de tareas de sustitución empieza a recibir tráfico del oyente de prueba, se ejecuta la función de Lambda denominada `LambdaFunctionToValidateAfterTestTrafficStarts`. Esta función probablemente ejecuta pruebas de validación para determinar si la implementación continúa. Si no especifica un agente de escucha de prueba en el grupo de implementaciones, esta se omite.
4. Después de completar las pruebas de validación en el enlace `AfterAllowTestTraffic` y antes de enviar tráfico de producción a la aplicación de Amazon ECS actualizada, se ejecuta la función de Lambda denominada `LambdaFunctionToValidateBeforeAllowingProductionTraffic`.
5. Después de enviar tráfico de producción a la aplicación de Amazon ECS actualizada en el conjunto de tareas de sustitución, se ejecuta la función de Lambda denominada `LambdaFunctionToValidateAfterAllowingProductionTraffic`.

Las funciones de Lambda que se ejecutan durante cualquier enlace pueden llevar a cabo pruebas de validación o recopilar métricas de tráfico.

## AppSpec Ejemplo de archivo para una implementación de AWS Lambda

Este es un ejemplo de un AppSpec archivo escrito en YAML para implementar una versión de la función Lambda.

```
version: 0.0
Resources:
 - myLambdaFunction:
 Type: AWS::Lambda::Function
 Properties:
 Name: "myLambdaFunction"
 Alias: "myLambdaFunctionAlias"
 CurrentVersion: "1"
 TargetVersion: "2"
Hooks:
 - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeTrafficShift"
 - AfterAllowTraffic: "LambdaFunctionToValidateAfterTrafficShift"
```

Esta es una versión del ejemplo anterior escrita en JSON.

```
{
 "version": 0.0,
 "Resources": [{
 "myLambdaFunction": {
 "Type": "AWS::Lambda::Function",
 "Properties": {
 "Name": "myLambdaFunction",
 "Alias": "myLambdaFunctionAlias",
 "CurrentVersion": "1",
 "TargetVersion": "2"
 }
 }
]},
 "Hooks": [{
 "BeforeAllowTraffic": "LambdaFunctionToValidateBeforeTrafficShift"
 },
 {
 "AfterAllowTraffic": "LambdaFunctionToValidateAfterTrafficShift"
 }
]
```

A continuación se muestra la secuencia de eventos durante la implementación:

1. Antes de desviar el tráfico de la versión 1 de la función de Lambda llamada `myLambdaFunction` a la versión 2, ejecute una función de Lambda llamada `LambdaFunctionToValidateBeforeTrafficShift` que valide que la implementación está lista para empezar a desviar el tráfico.
2. Si `LambdaFunctionToValidateBeforeTrafficShift` devuelve un código de salida de 0 (éxito), se empieza a desviar el tráfico a la versión 2 de `myLambdaFunction`. La configuración de esta implementación determina la velocidad a la que se desvía el tráfico.
3. Una vez completado el desvío de tráfico de la versión 1 de la función de Lambda llamada `myLambdaFunction` a la versión 2, ejecute una función de Lambda llamada `LambdaFunctionToValidateAfterTrafficShift` que valide que la implementación se ha realizado correctamente.

## AppSpec Ejemplo de archivo para una implementación local de EC2/

Este es un ejemplo de un AppSpec archivo para una implementación local en una instancia de Amazon Linux, Ubuntu Server o RHEL.

### Note

Las implementaciones en instancias de Windows Server no admiten el elemento `runas`. Si va a realizar una implementación en instancias de Windows Server, no lo incluya en el archivo AppSpec .

```
version: 0.0
os: linux
files:
 - source: Config/config.txt
 destination: /webapps/Config
 - source: source
 destination: /webapps/myApp
hooks:
 BeforeInstall:
 - location: Scripts/UnzipResourceBundle.sh
 - location: Scripts/UnzipDataBundle.sh
 AfterInstall:
```

```
- location: Scripts/RunResourceTests.sh
 timeout: 180
ApplicationStart:
- location: Scripts/RunFunctionalTests.sh
 timeout: 3600
ValidateService:
- location: Scripts/MonitorService.sh
 timeout: 3600
 runas: codedeployuser
```

Para una instancia de Windows Server, cambie `os: linux` por `os: windows`. Además, las rutas de `destination` deben ser completas (por ejemplo, `c:\temp\webapps\Config` y `c:\temp\webapps\myApp`). No incluya el elemento `runas`.

A continuación se muestra la secuencia de eventos durante la implementación:

1. Ejecutar el script que se encuentra en `Scripts/UnzipResourceBundle.sh`.
2. Si el script anterior devuelve un código de salida de 0 (éxito), ejecutar el script ubicado en `Scripts/UnzipDataBundle.sh`.
3. Copiar el archivo desde la ruta de `Config/config.txt` a la ruta `/webapps/Config/config.txt`.
4. Copiar recursivamente todos los archivos del directorio `source` en el directorio `/webapps/myApp`.
5. Ejecutar el script ubicado en `Scripts/RunResourceTests.sh` con un tiempo de espera de 180 segundos (3 minutos).
6. Ejecutar el script ubicado en `Scripts/RunFunctionalTests.sh` con un tiempo de espera de 3 600 segundos (1 hora).
7. Ejecutar el script ubicado en `Scripts/MonitorService.sh` como el usuario `codedeploy` con un tiempo de espera de 3 600 segundos (1 hora).

## AppSpec Espaciado de archivos

El siguiente es el formato correcto para el espaciado de AppSpec archivos. Los números entre corchetes indican el número de espacios que debe aparecer entre los elementos. Por ejemplo, `[4]` significa insertar cuatro espacios entre los elementos. CodeDeploy genera un error que puede ser difícil de depurar si las ubicaciones y el número de espacios de un AppSpec archivo no son correctos.

```

version:[1]version-number
os:[1]operating-system-name
files:
[2]-[1]source:[1]source-files-location
[4]destination:[1]destination-files-location
permissions:
[2]-[1]object:[1]object-specification
[4]pattern:[1]pattern-specification
[4]except:[1]exception-specification
[4]owner:[1]owner-account-name
[4]group:[1]group-name
[4]mode:[1]mode-specification
[4]acls:
[6]-[1]acls-specification
[4]context:
[6]user:[1]user-specification
[6]type:[1]type-specification
[6]range:[1]range-specification
[4]type:
[6]-[1]object-type
hooks:
[2]deployment-lifecycle-event-name:
[4]-[1]location:[1]script-location
[6]timeout:[1]timeout-in-seconds
[6]runas:[1]user-name

```

A continuación, se muestra un ejemplo de un archivo correctamente espaciado AppSpec :

```

version: 0.0
os: linux
files:
 - source: /
 destination: /var/www/html/WordPress
hooks:
 BeforeInstall:
 - location: scripts/install_dependencies.sh
 timeout: 300
 runas: root
 AfterInstall:
 - location: scripts/change_permissions.sh
 timeout: 300
 runas: root
 ApplicationStart:

```

```
- location: scripts/start_server.sh
- location: scripts/create_test_db.sh
 timeout: 300
 runas: root
ApplicationStop:
- location: scripts/stop_server.sh
 timeout: 300
 runas: root
```

Para obtener más información acerca del espaciado, consulte la especificación de [YAML](#).

## Valide su AppSpec archivo y su ubicación

### Sintaxis de archivos

Puede utilizar el script AppSpec Assistant AWS proporcionado para validar el contenido de un AppSpec archivo. Puede encontrar el script junto con las plantillas de AppSpec archivo en [GitHub](#).

También puede utilizar en su lugar una herramienta basada en navegador, como [YAML lint](#) u [Online YAML parser](#) para ayudarle a comprobar la sintaxis YAML.

### Ubicación del archivo

Para comprobar que ha colocado el AppSpec archivo en el directorio raíz de la estructura de directorios del contenido fuente de la aplicación, ejecute uno de los siguientes comandos:

En instancias locales de Linux, macOS o Unix:

```
ls path/to/root/directory/appspec.yml
```

Si el AppSpec archivo no se encuentra allí, aparecerá el error «No existe ese archivo o directorio».

En instancias de Windows locales:

```
dir path\to\root\directory\appspec.yml
```

Si el AppSpec archivo no se encuentra allí, aparecerá el error «No se ha encontrado el archivo».

## CodeDeploy referencia de configuración del agente

Cuando se instala el CodeDeploy agente, se coloca un archivo de configuración en la instancia. Este archivo de configuración especifica las rutas de los directorios y otros ajustes CodeDeploy

que se utilizarán cuando interactúe con la instancia. Puede cambiar algunas de las opciones de configuración en el archivo.


En el caso de las instancias de Amazon Linux, Ubuntu Server y Red Hat Enterprise Linux (RHEL), el archivo de configuración se denomina `codedeployagent.yml`. Se coloca en el directorio `/etc/codedeploy-agent/conf`.

En las instancias de Windows Server, el archivo de configuración se denomina `conf.yml`. Se coloca en el directorio `C:\ProgramData\Amazon\CodeDeploy`.

Los valores de configuración incluyen:

`:log_aws_wire:`

`true` Configúrelo para que el CodeDeploy agente capture los registros de cables de Amazon S3 y los escriba en un archivo con el nombre de la ubicación indicada **`codedeploy-agent.wire.log`** en la configuración: `log_dir:`

 **Warning**

Debe establecer `:log_aws_wire:` en `true` solo durante el tiempo necesario para capturar los registros en red. El archivo `codedeploy-agent.wire.log` puede aumentar de tamaño rápidamente. La salida del registro en red de este archivo puede contener información confidencial, incluido el contenido de texto sin formato de los archivos que se transfieren a o desde Amazon S3 mientras que esta configuración estaba establecida en `true`. Los registros de transferencias contienen información sobre toda la actividad de Amazon S3 asociada a la AWS cuenta mientras se estableció esta configuración `true`, no solo sobre

la actividad relacionada con CodeDeploy y las implementaciones.

El ajuste predeterminado es `false`.

Esta configuración se aplica a todos los tipos de instancias. Debe añadir esta opción de configuración a las instancias de Windows Server para que puedan utilizarla.

`:log_dir:`

La carpeta de la instancia en la que se almacenan los archivos de registro relacionados con las operaciones de los CodeDeploy agentes.

La configuración predeterminada es `'/var/log/aws/codedeploy-agent'` para las instancias de Amazon Linux, Ubuntu Server y RHEL y `C:\ProgramData\Amazon\CodeDeploy\log` para las instancias de Windows Server.

`:pid_dir:`

La carpeta en la que se almacena `codedeploy-agent.pid`.

Este archivo contiene el ID de proceso (PID) del CodeDeploy agente. El ajuste predeterminado es `'/opt/codedeploy-agent/state/.pid'`.

Esta configuración se aplica únicamente a las instancias de Amazon Linux, Ubuntu Server y RHEL.



|                                  |                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>:program_name:</code>      | <p>El nombre del programa del CodeDeploy agente.</p> <p>El ajuste predeterminado es <code>codedeploy-agent</code>.</p> <p>Esta configuración se aplica únicamente a las instancias de Amazon Linux, Ubuntu Server y RHEL.</p>                                                                                                                                                                   |
| <code>:root_dir:</code>          | <p>La carpeta donde se almacenan las revisiones relacionadas, el historial de implementación y los scripts de implementación en la instancia.</p> <p>La configuración predeterminada es <code>/opt/codedeploy-agent/deployment-root</code> para las instancias de Amazon Linux, Ubuntu Server y RHEL y <code>C:\ProgramData\Amazon\CodeDeploy</code> para las instancias de Windows Server.</p> |
| <code>:verbose:</code>           | <p>Configúrelo <code>true</code> para que el CodeDeploy agente imprima los archivos de registro de los mensajes de depuración en la instancia.</p> <p>El ajuste predeterminado es <code>false</code>.</p>                                                                                                                                                                                       |
| <code>:wait_between_runs:</code> | <p>El intervalo, en segundos, entre el sondeo de los CodeDeploy agentes en CodeDeploy busca de despliegues pendientes.</p> <p>El ajuste predeterminado es <code>1</code>.</p>                                                                                                                                                                                                                   |

`:on_premises_config_file:`

En las instancias en las instalaciones, la ruta a una ubicación alternativa para el archivo de configuración denominado `codedeploy.onpremises.yml` (para Ubuntu Server y RHEL) o `conf.onpremises.yml` (para Windows Server).

De manera predeterminada, estos archivos se almacenan en `/etc/codedeploy-agent/conf/codedeploy.onpremises.yml` para Ubuntu Server y RHEL, y `C:\ProgramData\Amazon\CodeDeploy\conf.onpremises.yml` para Windows Server.

Disponible en la versión 1.0.1.686 y versiones posteriores del agente. CodeDeploy

`:proxy_uri:`

(Opcional) El proxy HTTP a través del cual desea que el CodeDeploy agente se conecte AWS para sus CodeDeploy operaciones. Utilice un formato similar a `https://user:password@my.proxy:443/path?query`.

Disponible en la versión 1.0.1.824 y versiones posteriores del CodeDeploy agente.

|                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><code>:max_revisions:</code></p>                                                                                                                                                                                           | <p>(Opcional) El número de revisiones de la aplicación de un grupo de implementación que desea que archive el CodeDeploy agente. Se eliminarán todas las revisiones que superen el número especificado.</p> <p>Introduzca un número entero positivo. Si no se especifica ningún valor, CodeDeploy conservará las cinco revisiones más recientes además de la revisión actualmente implementada.</p> <p>Se admite en la versión 1.0.1.966 y en las versiones posteriores del CodeDeploy agente.</p> |
| <p><code>:enable_auth_policy:</code></p>                                                                                                                                                                                      | <p>(Opcional) <code>true</code> Configúrelo en si desea utilizar la <a href="#">autorización de IAM</a> para configurar el control de acceso y limitar los permisos del rol o usuario de IAM que utiliza el CodeDeploy agente. Para <a href="#">Úselo CodeDeploy con Amazon Virtual Private Cloud</a>, este valor debe ser <code>true</code>.</p> <p>El ajuste predeterminado es <code>false</code>.</p>                                                                                           |
| <p>(Opcional) Configúrelo en si desea utilizar la autorización de IAM para configurar el control de acceso y limitar el permiso del rol o usuario de IAM que utiliza el agente. <code>----sep----:disable_imds_v1:</code></p> | <p>Esta configuración está disponible con el agente 1.7.0 y versiones posteriores. CodeDeploy</p> <p><code>true</code> Configúrelo para deshabilitar la alternativa a IMDSv1 cuando se produzcan errores en IMDSv2. El valor predeterminado es (habilita la alternativa). <code>false</code></p>                                                                                                                                                                                                   |

## Temas relacionados de

[Trabajando con el CodeDeploy agente](#)

[Gestión de las operaciones CodeDeploy de los agentes](#)

# AWS CloudFormation plantillas de CodeDeploy referencia

En esta sección, se presentan los AWS CloudFormation recursos, la transformación y el enlace diseñados para funcionar con CodeDeploy las implementaciones. Para ver un tutorial sobre cómo crear una actualización de pila gestionada por AWS CloudFormation Hook for CodeDeploy, consulta [Cree una implementación azul/verde de Amazon ECS mediante AWS CloudFormation](#)

## Note

AWS CloudFormation Los ganchos forman parte de AWS CloudFormation los componentes de los ganchos de eventos del CodeDeploy ciclo de vida AWS y son diferentes de ellos.

Además de los demás métodos disponibles CodeDeploy, puede utilizar AWS CloudFormation plantillas para realizar las siguientes tareas:

- Crear aplicaciones.
- Crear grupos de implementaciones y especificar una revisión de destino.
- Crear configuraciones de implementación.
- Crear instancias de Amazon EC2.

AWS CloudFormation es un servicio que le ayuda a modelar y configurar sus AWS recursos mediante plantillas. Una AWS CloudFormation plantilla es un archivo de texto cuyo formato cumple con el estándar JSON. Usted crea una plantilla que describe todos los AWS recursos que desea y AWS CloudFormation se encarga de aprovisionar y configurar esos recursos por usted.

Para obtener más información, consulte [¿Qué es AWS CloudFormation?](#) y [Trabajo con plantillas de AWS CloudFormation](#) en la Guía del usuario de AWS CloudFormation .

Si piensa utilizar AWS CloudFormation plantillas compatibles con CodeDeploy las de su organización, como administrador debe conceder el acceso a los AWS servicios AWS CloudFormation y acciones de los que AWS CloudFormation depende. Para conceder permisos para crear aplicaciones, grupos de despliegue y configuraciones de despliegue, añada la siguiente política al conjunto de permisos de los usuarios con los que trabajarán AWS CloudFormation:

```
{
 "Version": "2012-10-17",
```

```

"Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cloudformation:*"
],
 "Resource": "*"
 }
]
}

```

Para obtener más información acerca de las políticas, consulte los siguientes temas:

- Para ver la política que debe añadirse al conjunto de permisos de los usuarios que crearán instancias de Amazon EC2, consulte [Crear una instancia de Amazon EC2 para CodeDeploy \(plantilla\)AWS CloudFormation](#).
- Para obtener información sobre cómo añadir políticas a los conjuntos de permisos, consulte [Crear un conjunto de permisos](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo restringir a los usuarios a un conjunto limitado de CodeDeploy acciones y recursos, consulte [AWS políticas gestionadas \(predefinidas\) para CodeDeploy](#).

La siguiente tabla muestra las acciones que una AWS CloudFormation plantilla puede realizar en su nombre e incluye enlaces a más información sobre los tipos de AWS recursos y sus tipos de propiedades que puede añadir a una AWS CloudFormation plantilla.

| Acción                                                                                                                                             | AWS CloudFormation referencia                     | Tipo de referencia         |
|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|----------------------------|
| Crea una CodeDeploy aplicación.                                                                                                                    | <a href="#">AWS::CodeImplementar: :aplicación</a> | AWS CloudFormation recurso |
| Crear y especificar los detalles de un grupo de implementación que se va a utilizar para implementar las revisiones de la aplicación. <sup>1</sup> | <a href="#">AWS::CodeDeploy::DeploymentGroup</a>  | AWS CloudFormation recurso |

| Acción                                                                                                                                                                                                                                                                                             | AWS CloudFormation referencia                                                                     | Tipo de referencia                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Cree un conjunto de reglas de despliegue, condiciones de éxito y condiciones de fallo de despliegue que CodeDeploy y se utilizarán durante un despliegue.</p>                                                                                                                                   | <p><a href="#">AWS::CodeDeploy::DeploymentConfig</a></p>                                          | <p>AWS CloudFormation recurso</p>                                                                                                                                                                                                                                                                                                                                                       |
| <p>Crear una instancia de Amazon EC2. <sup>2</sup></p>                                                                                                                                                                                                                                             | <p><a href="#">AWS::EC2::instance</a></p>                                                         | <p>AWS CloudFormation recurso</p>                                                                                                                                                                                                                                                                                                                                                       |
| <p>Utilice la tecnología AWS CloudFormation <code>AWS::CodeDeployBlueGreen</code> transform and <code>AWS::CodeDeploy::BlueGreen</code> hook para gestionar las actualizaciones de las pilas, crear recursos y redirigir el tráfico hacia despliegues CodeDeploy azules o verdes. <sup>3</sup></p> | <p><a href="#">AWS::CodeDeployBlueGreen</a></p> <p><a href="#">AWS::CodeDeploy::BlueGreen</a></p> | <p>La transformación <code>AWS::CodeDeployBlueGreen</code> es una macro alojada por AWS CloudFormation</p> <p>El <code>AWS::CodeDeploy::BlueGreen</code> gancho está estructurado como un Hook recurso en AWS CloudFormation. El enlace incluye parámetros que sustituyen al CodeDeploy AppSpec archivo y apuntan a los enlaces de eventos CodeDeploy del ciclo de vida designados.</p> |

| Acción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | AWS CloudFormation referencia | Tipo de referencia |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|--------------------|
| <p><sup>1</sup> Si especifica la versión de la revisión de la aplicación que desea que se implemente como parte del grupo de implementación, la revisión de destino se implementará en cuanto se complete el proceso de aprovisionamiento. Para obtener más información sobre la configuración de la plantilla, consulte <a href="#">la <u>revisión de CodeDeploy DeploymentGroup despliegue S3Location</u></a> y <a href="#">la <u>revisión de CodeDeploy DeploymentGroup despliegue GitHubLocation</u></a> en la Guía del AWS CloudFormation usuario.</p> <p><sup>2</sup> Proporcionamos plantillas que puede utilizar para crear instancias de Amazon EC2 en las regiones en las que CodeDeploy se admite. Para obtener más información sobre estas plantillas, consulte <a href="#">Crear una instancia de Amazon EC2 para CodeDeploy (plantilla)AWS CloudFormation</a>.</p> <p><sup>3</sup> Solo las implementaciones azul/verde de Amazon ECS son compatibles con esta configuración de implementación. Para obtener más información acerca de las configuraciones de implementación para implementaciones azul/verde de Amazon ECS a través de AWS CloudFormation, consulte <a href="#">Configuraciones de implementación para implementaciones azul/verde de AWS CloudFormation (Amazon ECS)</a>. Para obtener más información sobre las implementaciones azul/verde de Amazon ECS AWS CloudFormation y sobre cómo ver su implementación en CodeDeploy ellas, consulte. <a href="#">Cree una implementación azul/verde de Amazon ECS mediante AWS CloudFormation</a></p> |                               |                    |

## Úselo CodeDeploy con Amazon Virtual Private Cloud

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus AWS recursos, puede establecer una conexión privada entre su VPC y CodeDeploy. Puede utilizar esta conexión para

poder CodeDeploy comunicarse con los recursos de su VPC sin tener que pasar por la Internet pública.

Amazon VPC es un AWS servicio que puede utilizar para lanzar AWS recursos en una red virtual que usted defina. Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de ruteo y las gateways de red. Con los puntos de enlace de la VPC, la AWS red gestiona el enrutamiento entre la VPC y AWS los servicios, y puede utilizar las políticas de IAM para controlar el acceso a los recursos del servicio.

Para conectar su VPC CodeDeploy, debe definir un punto final de VPC de interfaz para. CodeDeploy Un punto final de interfaz es una interfaz de red elástica con una dirección IP privada que sirve como punto de entrada para el tráfico destinado a un AWS servicio compatible. El punto final proporciona una conectividad fiable y escalable CodeDeploy sin necesidad de una puerta de enlace a Internet, una instancia de traducción de direcciones de red (NAT) o una conexión VPN. Para obtener más información, consulte [¿Qué es Amazon VPC?](#) en la Guía del usuario de Amazon VPC.

Los puntos finales de VPC de interfaz cuentan con una AWS tecnología que permite la comunicación privada entre AWS servicios mediante una interfaz de red elástica con direcciones IP privadas. AWS PrivateLink Para obtener más información, consulte [AWS PrivateLink](#).

Los siguientes pasos son para usuarios de Amazon VPC. Para obtener más información, consulte [Introducción](#) en la Guía del usuario de Amazon VPC.

## Disponibilidad

CodeDeploy tiene dos puntos finales de VPC: uno para las operaciones de los CodeDeploy agentes y otro para CodeDeploy las operaciones de la API. La siguiente tabla muestra las AWS regiones compatibles para cada punto final.

| Nombre de la región                    | Código de región | Punto de conexión del agente | Punto de conexión de la API |
|----------------------------------------|------------------|------------------------------|-----------------------------|
| Este de EE. UU.<br>(Norte de Virginia) | us-east-1        | Sí                           | Sí                          |
| Este de EE. UU.<br>(Ohio)              | us-east-2        | Sí                           | Sí                          |



| Nombre de la región                       | Código de región | Punto de conexión del agente | Punto de conexión de la API |
|-------------------------------------------|------------------|------------------------------|-----------------------------|
| Oeste de EE. UU.<br>(Norte de California) | us-west-1        | Sí                           | Sí                          |
| Oeste de EE. UU.<br>(Oregón)              | us-west-2        | Sí                           | Sí                          |
| África (Ciudad del Cabo)                  | af-south-1       | Sí                           | No                          |
| Asia-Pacífico (Hong Kong)                 | ap-east-1        | Sí                           | Sí                          |
| Asia-Pacífico<br>(Hyderabad)              | ap-south-2       | Sí                           | No                          |
| Asia-Pacífico<br>(Yakarta)                | ap-southeast-3   | Sí                           | No                          |
| Asia-Pacífico<br>(Melbourne)              | ap-southeast-4   | Sí                           | No                          |
| Asia-Pacífico<br>(Bombay)                 | ap-south-1       | Sí                           | Sí                          |
| Asia Pacífico (Osaka)                     | ap-northeast-3   | Sí                           | No                          |
| Asia Pacífico (Seúl)                      | ap-northeast-2   | Sí                           | Sí                          |
| Asia Pacífico<br>(Singapur)               | ap-southeast-1   | Sí                           | Sí                          |
| Asia Pacífico (Sídney)                    | ap-southeast-2   | Sí                           | Sí                          |
| Asia Pacífico (Tokio)                     | ap-northeast-1   | Sí                           | Sí                          |
| Canadá (Central)                          | ca-central-1     | Sí                           | Sí                          |

| Nombre de la región                 | Código de región | Punto de conexión del agente | Punto de conexión de la API |
|-------------------------------------|------------------|------------------------------|-----------------------------|
| China (Pekín)                       | cn-north-1       | Sí                           | No                          |
| China (Ningxia)                     | cn-northwest-1   | No                           | No                          |
| Europa (Fráncfort)                  | eu-central-1     | Sí                           | Sí                          |
| Europa (Irlanda)                    | eu-west-1        | Sí                           | Sí                          |
| Europa (Londres)                    | eu-west-2        | Sí                           | Sí                          |
| Europa (Milán)                      | eu-south-1       | Sí                           | No                          |
| Europa (París)                      | eu-west-3        | Sí                           | Sí                          |
| Europa (España)                     | eu-south-2       | Sí                           | No                          |
| Europa (Estocolmo)                  | eu-north-1       | Sí                           | Sí                          |
| Europa (Zúrich)                     | eu-central-2     | Sí                           | No                          |
| Israel (Tel Aviv)                   | il-central-1     | Sí                           | Sí                          |
| Medio Oriente (Baréin)              | me-south-1       | Sí                           | Sí                          |
| Medio Oriente (EAU)                 | me-central-1     | Sí                           | No                          |
| América del Sur (São Paulo)         | sa-east-1        | Sí                           | Sí                          |
| AWS GovCloud (Este de EE. UU.)      | us-gov-east-1    | No                           | No                          |
| AWS GovCloud (Estados Unidos-Oeste) | us-gov-west-1    | No                           | No                          |

## Cree puntos finales de VPC para CodeDeploy

Para empezar a usarlo CodeDeploy con su VPC, cree un punto de enlace de VPC de interfaz para. CodeDeploy CodeDeploy requiere puntos finales separados para las operaciones del agente Git y para las operaciones de la CodeDeploy API. En función de sus necesidades empresariales, es posible que tenga que crear más de un punto de conexión de VPC. Al crear un punto final de VPC para CodeDeploy, elija AWS Servicios y, en Nombre del servicio, elija una de las siguientes opciones:

- `com.amazonaws.región.codedeploy`: elige esta opción si quieres crear un punto final de VPC para las operaciones de la API. CodeDeploy Por ejemplo, elige esta opción si tus usuarios utilizan la AWS CLI CodeDeploy API o los AWS SDK CodeDeploy para interactuar con ellos en operaciones como `CreateApplication`, y. `GetDeployment` `ListDeploymentGroups`
- `com.amazonaws.región.codedeploy-commands-secure`: elija esta opción si quiere crear un punto final de VPC para las operaciones de los CodeDeploy agentes. También tendrá que definir `:enable_auth_policy`: en `true` en el archivo de configuración del agente y adjuntar los permisos necesarios. Para obtener más información, consulte [Configure los permisos del CodeDeploy agente y de IAM](#).

Si utiliza implementaciones de Lambda o ECS, solo necesita crear un punto de conexión de VPC para `com.amazonaws.región.codedeploy`. Los clientes que utilicen implementaciones de Amazon EC2 necesitarán puntos de enlace de VPC para ambos `com.amazonaws.región.codedeploy` y `com.amazonaws.región.codedeploy-commands-secure`.

## Configure los permisos del CodeDeploy agente y de IAM

Para usar los puntos de enlace de Amazon VPC con CodeDeploy, debe establecer el valor de `:enable_auth_policy`: a `true` en el archivo de configuración del agente ubicado en sus instancias EC2 o locales. Para obtener más información sobre el archivo de configuración del agente, consulte [CodeDeploy referencia de configuración del agente](#).

También debe añadir los siguientes permisos de IAM a su perfil de instancia de Amazon EC2 (si utiliza instancias de Amazon EC2) o a su usuario o rol de IAM (si utiliza instancias en las instalaciones).

```
{
 "Statement": [

```

```
{
 "Action": [
 "codedeploy-commands-secure:GetDeploymentSpecification",
 "codedeploy-commands-secure:PollHostCommand",
 "codedeploy-commands-secure:PutHostCommandAcknowledgement",
 "codedeploy-commands-secure:PutHostCommandComplete"
],
 "Effect": "Allow",
 "Resource": "*"
}
```

Para obtener más información, consulte [Creación de un punto de conexión de tipo interfaz](#) en la Guía del usuario de Amazon VPC.

## CodeDeploy referencia del kit de recursos

Muchos de los archivos en los que CodeDeploy se basa se almacenan en depósitos de Amazon S3 AWS específicos de cada región y disponibles públicamente. Estos archivos incluyen los archivos de instalación del CodeDeploy agente, las plantillas y los archivos de aplicaciones de muestra. A esta colección de archivos la denominamos kit CodeDeploy de recursos.

### Temas

- [Nombres de bucket del kit de recursos por región](#)
- [Contenido del kit de recursos](#)
- [Mostrar una lista de los archivos del kit de recursos](#)
- [Descargar los archivos del kit de recursos](#)

## Nombres de bucket del kit de recursos por región

En esta tabla se muestra los nombres de las sustituciones de *nombre de bucket* que se necesitan para algunos procedimientos de la guía. Estos son los nombres de los buckets de Amazon S3 que contienen los archivos del kit CodeDeploy de recursos.

**Note**

Para acceder al bucket de Amazon S3 en la región de Asia Pacífico (Hong Kong), debe habilitar la región en su AWS cuenta. Para obtener más información, consulte [Gestión de AWS regiones](#).

| Nombres de las regiones                | Sustitución de <i>bucket-name</i> | Identificador de la región |
|----------------------------------------|-----------------------------------|----------------------------|
| Este de EE. UU. (Norte de Virginia)    | aws-codedeploy-us-east-1          | us-east-1                  |
| Este de EE. UU. (Ohio)                 | aws-codedeploy-us-east-2          | us-east-2                  |
| Oeste de EE. UU. (Norte de California) | aws-codedeploy-us-west-1          | us-west-1                  |
| Oeste de EE. UU. (Oregón)              | aws-codedeploy-us-west-2          | us-west-2                  |
| África (Ciudad del Cabo)               | aws-codedeploy-af-south-1         | af-south-1                 |
| Asia-Pacífico (Hong Kong)              | aws-codedeploy-ap-east-1          | ap-east-1                  |
| Asia-Pacífico (Hyderabad)              | aws-codedeploy-ap-south-2         | ap-south-2                 |
| Asia-Pacífico (Yakarta)                | aws-codedeploy-ap-southeast-3     | ap-southeast-3             |
| Asia-Pacífico (Melbourne)              | aws-codedeploy-ap-southeast-4     | ap-southeast-4             |
| Asia-Pacífico (Bombay)                 | aws-codedeploy-ap-south-1         | ap-south-1                 |
| Asia-Pacífico (Osaka)                  | aws-codedeploy-ap-northeast-3     | ap-northeast-3             |
| Asia-Pacífico (Seúl)                   | aws-codedeploy-ap-northeast-2     | ap-northeast-2             |

| Nombres de las regiones        | Sustitución de <i>bucket-name</i> | Identificador de la región |
|--------------------------------|-----------------------------------|----------------------------|
| Asia-Pacífico (Singapur)       | aws-codedeploy-ap-southeast-1     | ap-southeast-1             |
| Asia-Pacífico (Sídney)         | aws-codedeploy-ap-southeast-2     | ap-southeast-2             |
| Asia-Pacífico (Tokio)          | aws-codedeploy-ap-northeast-1     | ap-northeast-1             |
| Canadá (centro)                | aws-codedeploy-ca-central-1       | ca-central-1               |
| Europa (Fráncfort)             | aws-codedeploy-eu-central-1       | eu-central-1               |
| Europa (Irlanda)               | aws-codedeploy-eu-west-1          | eu-west-1                  |
| Europa (Londres)               | aws-codedeploy-eu-west-2          | eu-west-2                  |
| Europa (Milán)                 | aws-codedeploy-eu-south-1         | eu-south-1                 |
| Europa (París)                 | aws-codedeploy-eu-west-3          | eu-west-3                  |
| Europa (España)                | aws-codedeploy-eu-south-2         | eu-south-2                 |
| Europa (Estocolmo)             | aws-codedeploy-eu-north-1         | eu-north-1                 |
| Europa (Zúrich)                | aws-codedeploy-eu-central-2       | eu-central-2               |
| Israel (Tel Aviv)              | aws-codedeploy-il-central-1       | il-central-1               |
| Medio Oriente (Baréin)         | aws-codedeploy-me-south-1         | me-south-1                 |
| Medio Oriente (EAU)            | aws-codedeploy-me-central-1       | me-central-1               |
| América del Sur (São Paulo)    | aws-codedeploy-sa-east-1          | sa-east-1                  |
| AWS GovCloud (Este de EE. UU.) | aws-codedeploy-us-gov-east-1      | us-gov-east-1              |

| Nombres de las regiones      | Sustitución de <i>bucket-na</i><br><i>me</i> | Identificador de la región |
|------------------------------|----------------------------------------------|----------------------------|
| AWS GovCloud (EE. UU.-Oeste) | aws-codedeploy-us-gov-oeste-1                | us-gov-west-1              |

## Contenido del kit de recursos

En la siguiente tabla se enumeran los archivos del kit de CodeDeploy recursos.

| Archivos                    | Descripción                                                                                                                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LATEST_VERSION              | Archivo que utilizan los mecanismos de actualización, como Amazon EC2 Systems Manager, para determinar la versión más reciente CodeDeploy del agente.                                                                                            |
| VERSION                     | El mecanismo de actualización automática se eliminó en la versión 1.1.0 del CodeDeploy agente y este archivo ya no se utiliza. Archivo que utilizan los CodeDeploy agentes para actualizarse a sí mismos mientras se ejecutan en las instancias. |
| codedeploy-agent.noarch.rpm | El CodeDeploy agente para Amazon Linux y Red Hat Enterprise Linux (RHEL). Puede haber varios archivos con el mismo nombre de archivo de base, pero diferentes versiones (como -1.0-0).                                                           |
| codedeploy-agent_all.deb    | El CodeDeploy agente de Ubuntu Server. Puede haber varios archivos con el mismo nombre de archivo de base, pero diferentes versiones (como _1.0-0).                                                                                              |
| codedeploy-agent.msi        | El CodeDeploy agente de Windows Server. Puede haber varios archivos con el mismo                                                                                                                                                                 |

| Archivos                                 | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                          | nombre de archivo de base, pero diferentes versiones (como -1.0-0).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| install                                  | Un archivo que puede utilizar para instalar el CodeDeploy agente con mayor facilidad.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| CodeDeploy_SampleCF_Template.json        | Una AWS CloudFormation plantilla que puede utilizar para lanzar de una a tres instancias de Amazon EC2 que ejecuten Amazon Linux o Windows Server. Puede haber varios archivos con el mismo nombre de archivo de base, pero diferentes versiones (como -1.0.0).                                                                                                                                                                                                                                                                                                              |
| CodeDeploy_SampleCF_ELB_Integration.json | Una AWS CloudFormation plantilla que puede utilizar para crear un sitio web de ejemplo con equilibrio de carga que se ejecute en un servidor web Apache. La aplicación está configurada para abarcar todas las zonas de disponibilidad de la región en la que la creó. Esta plantilla crea tres instancias de Amazon EC2 y un perfil de instancia de IAM para conceder a las instancias acceso a los recursos de Amazon S3 AWS CloudFormation, Amazon EC2 Auto Scaling y Elastic Load Balancing. También crea un balanceador de cargas y una función de servicio. CodeDeploy |
| SampleApp_ELB_Integration.zip            | Una revisión de la aplicación de ejemplo que se puede implementar en una instancia de Amazon EC2 registrada en un equilibrador de carga de Elastic Load Balancing.                                                                                                                                                                                                                                                                                                                                                                                                           |



| Archivos              | Descripción                                                                                                                                                                                                                                                                  |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SampleApp_Linux.zip   | Revisión de la aplicación de muestra que se puede implementar en una instancia de Amazon EC2 que ejecute Amazon Linux o en una instancia de Ubuntu Server o RHEL. Puede haber varios archivos con el mismo nombre de archivo de base, pero diferentes versiones (como -1.0). |
| SampleApp_Windows.zip | Revisión de aplicación de muestra que se puede implementar en una instancia de Windows Server. Puede haber varios archivos con el mismo nombre de archivo de base, pero diferentes versiones (como -1.0).                                                                    |

## Mostrar una lista de los archivos del kit de recursos

Para ver una lista de archivos, utilice el comando `aws s3 ls` correspondiente a su región.

### Note

Los archivos de los buckets están diseñados para funcionar con los recursos de su región correspondiente.

- `aws s3 ls --recursive s3://aws-codedeploy-us-east-2 --region us-east-2`
- `aws s3 ls --recursive s3://aws-codedeploy-us-east-1 --region us-east-1`
- `aws s3 ls --recursive s3://aws-codedeploy-us-west-1 --region us-west-1`
- `aws s3 ls --recursive s3://aws-codedeploy-us-west-2 --region us-west-2`
- `aws s3 ls --recursive s3://aws-codedeploy-ca-central-1 --region ca-central-1`

- ```
aws s3 ls --recursive s3://aws-codedeploy-eu-west-1 --region eu-west-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-eu-west-2 --region eu-west-2
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-eu-west-3 --region eu-west-3
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-eu-central-1 --region eu-central-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-il-central-1 --region il-central-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-ap-east-1 --region ap-east-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-ap-northeast-1 --region ap-northeast-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-ap-northeast-2 --region ap-northeast-2
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-1 --region ap-southeast-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-2 --region ap-southeast-2
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-ap-southeast-4 --region ap-southeast-4
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-ap-south-1 --region ap-south-1
```
- ```
aws s3 ls --recursive s3://aws-codedeploy-sa-east-1 --region sa-east-1
```

Descargar los archivos del kit de recursos

Para descargar un archivo, utilice el comando `aws s3 cp` correspondiente a su región.

Note

Asegúrese de utilizar el punto (.) cerca del final. Esto descarga el archivo en el directorio actual.

Por ejemplo, los siguientes comandos descargan un único archivo denominado `SampleApp_Linux.zip` de una de las carpetas `/samples/latest/` de los buckets:

- ```
aws s3 cp s3://aws-codedeploy-us-east-2/samples/latest/SampleApp_Linux.zip . --region us-east-2
```
- ```
aws s3 cp s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip . --region us-east-1
```
- ```
aws s3 cp s3://aws-codedeploy-us-west-1/samples/latest/SampleApp_Linux.zip . --region us-west-1
```
- ```
aws s3 cp s3://aws-codedeploy-us-west-2/samples/latest/SampleApp_Linux.zip . --region us-west-2
```
- ```
aws s3 cp s3://aws-codedeploy-ca-central-1/samples/latest/SampleApp_Linux.zip . --region ca-central-1
```
- ```
aws s3 cp s3://aws-codedeploy-eu-west-1/samples/latest/SampleApp_Linux.zip . --region eu-west-1
```
- ```
aws s3 cp s3://aws-codedeploy-eu-west-2/samples/latest/SampleApp_Linux.zip . --region eu-west-2
```
- ```
aws s3 cp s3://aws-codedeploy-eu-west-3/samples/latest/SampleApp_Linux.zip . --region eu-west-3
```
- ```
aws s3 cp s3://aws-codedeploy-eu-central-1/samples/latest/SampleApp_Linux.zip . --region eu-central-1
```
- ```
aws s3 cp s3://aws-codedeploy-il-central-1/samples/latest/SampleApp_Linux.zip . --region il-central-1
```
- ```
aws s3 cp s3://aws-codedeploy-ap-east-1/samples/latest/SampleApp_Linux.zip . --region ap-east-1
```
- ```
aws s3 cp s3://aws-codedeploy-ap-northeast-1/samples/latest/SampleApp_Linux.zip . --region ap-northeast-1
```

- ```
aws s3 cp s3://aws-codedeploy-ap-northeast-2/samples/latest/SampleApp_Linux.zip . --region ap-northeast-2
```
- ```
aws s3 cp s3://aws-codedeploy-ap-southeast-1/samples/latest/SampleApp_Linux.zip . --region ap-southeast-1
```
- ```
aws s3 cp s3://aws-codedeploy-ap-southeast-2/samples/latest/SampleApp_Linux.zip . --region ap-southeast-2
```
- ```
aws s3 cp s3://aws-codedeploy-ap-southeast-4/samples/latest/SampleApp_Linux.zip . --region ap-southeast-4
```
- ```
aws s3 cp s3://aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip . --region ap-south-1
```
- ```
aws s3 cp s3://aws-codedeploy-sa-east-1/samples/latest/SampleApp_Linux.zip . --region sa-east-1
```

Para descargar todos los archivos, utilice uno de los siguientes comandos correspondientes a su región:

- ```
aws s3 cp --recursive s3://aws-codedeploy-us-east-2 . --region us-east-2
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-us-east-1 . --region us-east-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-us-west-1 . --region us-west-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-us-west-2 . --region us-west-2
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ca-central-1 . --region ca-central-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-eu-west-1 . --region eu-west-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-eu-west-2 . --region eu-west-2
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-eu-west-3 . --region eu-west-3
```

- ```
aws s3 cp --recursive s3://aws-codedeploy-eu-central-1 . --region eu-central-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-il-central-1 . --region il-central-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ap-east-1 . --region ap-east-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ap-northeast-1 . --region ap-northeast-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ap-northeast-2 . --region ap-northeast-2
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-1 . --region ap-southeast-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-2 . --region ap-southeast-2
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ap-southeast-4 . --region ap-southeast-4
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-ap-south-1 . --region ap-south-1
```
- ```
aws s3 cp --recursive s3://aws-codedeploy-sa-east-1 . --region sa-east-1
```

CodeDeploy cuotas

En las siguientes tablas se describen las cuotas en CodeDeploy.

Note

El límite de Horas de ejecución de la implementación local de EC2/en las instalaciones varía. Para las configuraciones de implementación personalizadas creadas antes de junio de 2023, el límite es de 8 horas. Para las configuraciones de implementación personalizadas creadas en junio de 2023 o después, el límite es de 12 horas. Para las configuraciones de implementación predefinidas, el límite es de 12 horas.

Nombre	Valor predeterminado	Ajuste	Descripción
AWS La implementación de Lambda se ejecuta en horas	Cada región admitida: 50	No	Número máximo de horas que puede ejecutarse una implementación de AWS Lambda (48 horas para el tiempo máximo entre el primer y el último cambio de tráfico, más una hora para cada uno de los dos posibles enlaces del ciclo de vida)
Aplicaciones asociadas por cuenta y por región	Cada región admitida: 1000	Sí	El número máximo de aplicaciones asociadas a una AWS cuenta en una sola región
Alarmas asociadas por grupo de implementación	Cada región admitida: 20	Sí	Número máximo de alarmas asociadas con el grupo de implementación
Grupos de escalado automático en un grupo de implementación	Cada región admitida: 10	Sí	Número máximo de grupos de Amazon EC2 Auto Scaling en un grupo de implementación
Implementaciones simultáneas por cuenta	Cada región admitida: 1000	Sí	Número máximo de despliegues simultáneos asociados a una AWS cuenta. Cada implementación en una instancia de Amazon EC2 escalada de un grupo de Amazon EC2 Auto Scaling cuenta como una implement

Nombre	Valor predeterminado	Ajuste	Descripción
			acción simultánea única para cada aplicación a la que esté asociada la instancia de EC2.
Implementaciones simultáneas por grupo de implementación	Cada región admitida: 1	No	Número máximo de implementaciones simultáneas en un grupo de implementación. Este límite evita implementaciones simultáneas de la misma aplicación en el mismo grupo de implementación.
Configuraciones de implementación personalizadas por cuenta	Cada región admitida: 50	No	Número máximo de configuraciones de despliegue personalizadas asociadas a una cuenta AWS
Número de grupos de implementación asociados con una sola aplicación	Cada región admitida: 1000	Sí	Cantidad máxima de grupos de implementaciones asociados a una sola aplicación

Nombre	Valor predeterminado	Ajuste	Descripción
Horas de ejecución de la implementación azul/verde de EC2/en las instalaciones	Cada región admitida: 109	No	Número máximo de horas durante las que una implementación azul/verde de EC2/en las instalaciones puede ejecutarse (48 horas para cada uno de los dos límites anteriores más una hora para cada uno de los 13 posibles eventos del ciclo de vida)
Horas de ejecución de la implementación local de EC2/en las instalaciones	Cada región admitida: 12	No	Número máximo de horas durante las que una implementación local de EC2/en las instalaciones puede ejecutarse
Desencadenantes de notificaciones de eventos en un grupo de implementación	Cada región admitida: 10	Sí	Cantidad máxima de disparadores de notificaciones de eventos en un grupo de implementaciones
GitHub tokens de conexión por cuenta	Cada región admitida: 25	No	Número máximo de tokens de GitHub conexión para una sola AWS cuenta

Nombre	Valor predeterminado	Ajuste	Descripción
Número de horas entre la finalización de una implementación y la finalización de las instancias originales durante una implementación azul/verde de EC2/en las instalaciones	Cada región admitida: 48	No	Número máximo de horas entre la finalización de una implementación y la finalización de las instancias originales durante una implementación azul/verde de EC2/en las instalaciones
Horas entre la implementación de una revisión y el momento en que el tráfico se desvía a las instancias de reemplazo durante una implementación azul/verde de EC2/en las instalaciones	Cada región admitida: 48	No	Número máximo de horas entre la implementación de una revisión y el momento en que el tráfico se desvía a las instancias de reemplazo durante una implementación azul/verde de EC2/en las instalaciones
Las instancias se cuentan por implementación	us-east-1: 2000 Cada una de las demás regiones compatibles: 1000	Sí	Cantidad máxima de instancias de una única implementación
Número de minutos que una implementación azul/verde puede esperar después de una implementación satisfactoria antes de finalizar las instancias de la implementación original	Cada región admitida: 2800	No	Número máximo de minutos que una implementación azul/verde pueden esperar después de una implementación satisfactoria antes de finalizar las instancias de la implementación original

Nombre	Valor predeterminado	Ajuste	Descripción
Minutos entre el primer y el último cambio de tráfico durante un despliegue AWS lineal o canario de Lambda	Cada región admitida: 2880	No	Número máximo de minutos entre el primer y el último cambio de tráfico durante un despliegue lineal o canario de AWS Lambda
Minutos hasta que se produce un error en la implementación si un evento del ciclo de vida no comienza	Cada región admitida: 5	No	Número máximo de minutos hasta que una implementación falle si un evento del ciclo de vida no se inicia después de (1) que se active una implementación mediante la consola o el comando create-deployment de la AWS CLI, o (2) que se complete el evento del ciclo de vida anterior.
Número de grupos de implementación que se pueden asociar con un servicio de Amazon ECS	Cada región admitida: 1	No	Número máximo de grupos de implementación que se pueden asociar con un servicio de Amazon ECS
Número de instancias que se pueden transferir a la acción de la API BatchGetOnPremisesInstances	Cada región admitida: 100	No	Número máximo de instancias que se pueden pasar a la acción de la BatchGetOnPremisesInstances API

Nombre	Valor predeterminado	Ajuste	Descripción
Número de instancias que pueden utilizar las implementaciones simultáneas en curso por cuenta	us-east-1: 3000 Cada una de las demás regiones compatibles: 1000	Sí	Número máximo de instancias que pueden utilizar las implementaciones simultáneas en curso y que están asociadas con una cuenta
Número de receptores en una ruta de tráfico durante una implementación de Amazon ECS	Cada región admitida: 1	No	Número máximo de receptores en una ruta de tráfico durante una implementación de Amazon ECS
Segundos hasta que se considere que un evento de ciclo de vida de implementación ha fallado si no se ha completado	Cada región admitida: 3600 segundos	No	Número máximo de segundos hasta que se considere que un evento de ciclo de vida de implementación ha fallado si no se ha completado
Tamaño del nombre del grupo de implementación	Cada región admitida: 100	No	Cantidad máxima de caracteres en un nombre de grupo de implementaciones
Tamaño de la clave de la etiqueta	Cada región admitida: 128	No	Cantidad máxima de caracteres en una clave de etiqueta
Tamaño del valor de la etiqueta	Cada región admitida: 256	No	Cantidad máxima de caracteres en un valor de etiqueta

Nombre	Valor predeterminado	Ajuste	Descripción
Etiquetas de un grupo de implementación	Cada región admitida: 10	No	Cantidad máxima de etiquetas en un grupo de implementaciones
Tráfico que se puede desplazar en un incremento durante una implementación de AWS Lambda	Cada región admitida: 99	No	Porcentaje máximo de tráfico que se puede desplazar en un incremento durante una implementación de AWS Lambda

Solución de problemas CodeDeploy

Utilice los temas de esta sección para ayudar a resolver los problemas y errores que puedan surgir al CodeDeploy utilizarlos.

Note

Puede identificar las causas de muchos errores de implementación revisando los archivos log creados durante el proceso de implementación. Para simplificar, recomendamos utilizar Amazon CloudWatch Logs para supervisar de forma centralizada los archivos de registro en lugar de verlos instancia por instancia. Para obtener más información, consulte [Monitoring Deployments with Amazon CloudWatch Tools](#).

Temas

- [Solución de problemas generales](#)
- [Resolución de problemas de implementación de EC2/en las instalaciones](#)
- [Resolución problemas de implementación de Amazon ECS](#)
- [Solución de problemas de implementación de AWS Lambda](#)
- [Solución de problemas de grupos de implementación](#)
- [Solución de problemas de instancias](#)
- [Solucione problemas con los GitHub tokens](#)
- [Solución de problemas de Amazon EC2 Auto Scaling](#)
- [Códigos de error para AWS CodeDeploy](#)

Solución de problemas generales

Temas

- [Lista de comprobación de solución de problemas generales](#)
- [CodeDeploy los recursos de despliegue solo se admiten en algunas regiones AWS](#)
- [Los procedimientos de esta guía no coinciden con los de la CodeDeploy consola](#)
- [Los roles de IAM necesarios no están disponibles](#)

- [El uso de algunos editores de texto para crear AppSpec archivos y scripts de shell puede provocar errores en las implementaciones](#)
- [El uso de Finder en macOS para empaquetar una revisión de la aplicación puede producir un error en la implementación](#)

Lista de comprobación de solución de problemas generales

Puede utilizar la siguiente lista de comprobación para solucionar un error en la implementación.

1. Consulte [Ver detalles CodeDeploy de la implementación](#) y [View Instance Details](#) para determinar por qué se produjo un error en la implementación. Si no puede determinar la causa, revise los elementos de esta lista de comprobación.
2. Compruebe si ha configurado correctamente las instancias:
 - ¿Se ha lanzado la instancia con un par de claves de EC2 especificadas? Para obtener más información, consulte los [pares de claves de EC2](#) en la Guía del usuario de Amazon EC2.
 - ¿Se ha asociado a la instancia el perfil de instancia de IAM correcto? Para obtener más información, consulte [Configurar una instancia de Amazon EC2 con la que trabajar CodeDeploy](#) y [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#).
 - ¿Se ha etiquetado la instancia? Para obtener más información, consulte [Trabajar con etiquetas en la consola en la](#) Guía del usuario de Amazon EC2.
 - ¿El CodeDeploy agente está instalado, actualizado y en ejecución en la instancia? Para obtener más información, consulte [Gestión de las operaciones CodeDeploy de los agentes](#). Para comprobar qué versión del agente está instalada, consulte [Determine la versión del CodeDeploy agente](#).
3. Compruebe la configuración de la aplicación y del grupo de implementaciones:
 - Para comprobar la configuración de la aplicación, consulte [Vea los detalles de la solicitud con CodeDeploy](#).
 - Para comprobar la configuración del grupo de implementaciones, consulte [Vea los detalles del grupo de implementación con CodeDeploy](#).
4. Confirme que la revisión de la aplicación está configurada correctamente:
 - Compruebe el formato del AppSpec archivo. Para obtener más información, consulte [Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy](#) y [CodeDeploy AppSpec referencia de archivo](#).

- Compruebe su depósito o GitHub repositorio de Amazon S3 para comprobar que la revisión de la aplicación se encuentra en la ubicación esperada.
 - Revise los detalles de la revisión de su CodeDeploy aplicación para asegurarse de que está registrada correctamente. Para obtener más información, consulte [Vea los detalles de la revisión de la aplicación con CodeDeploy](#).
 - Si realiza la implementación desde Amazon S3, compruebe su bucket de Amazon S3 para comprobar que se le han CodeDeploy concedido los permisos para descargar la revisión de la aplicación. Para obtener información sobre las políticas de buckets, consulte [Requisitos previos para una implementación](#).
 - Si realiza la implementación desde GitHub, compruebe su GitHub repositorio para comprobar que se le han CodeDeploy concedido los permisos para descargar la revisión de la aplicación. Para obtener más información, consulte [Cree una implementación con CodeDeploy y GitHub autenticación con aplicaciones en CodeDeploy](#).
5. Compruebe si el rol de servicio está configurado correctamente. Para obtener más información, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).
6. Confirme que ha seguido los pasos de [Empezar con CodeDeploy](#) para:
- Aprovisionar a un usuario con los permisos necesarios.
 - Instalar o actualizar y configurar la AWS CLI.
 - Crear un perfil de instancia y un rol de servicio de IAM.
- Para obtener más información, consulte [Administración de identidades y accesos para AWS CodeDeploy](#).
7. Confirma que estás usando AWS CLI la versión 1.6.1 o posterior. Para comprobar la versión que tiene instalada, llame a `aws --version`.

Si sigue sin poder solucionar el error de la implementación, consulte los otros puntos de este tema.

CodeDeploy los recursos de despliegue solo se admiten en algunas regiones AWS

Si no ve o no puede acceder a las aplicaciones, los grupos de despliegue, las instancias u otros recursos de despliegue desde la consola AWS CLI o la CodeDeploy consola, asegúrese de hacer referencia a una de las AWS regiones que aparecen en [Región y puntos finales](#) en Referencia general de AWS

Las instancias EC2 y los grupos de Auto Scaling de Amazon EC2 que se utilizan CodeDeploy en las implementaciones deben lanzarse y crearse en una de estas regiones. AWS

Si utiliza el AWS CLI, ejecute el `aws configure` comando desde. AWS CLI A continuación, puede ver y configurar su AWS región predeterminada.

Si utilizas la CodeDeploy consola, en la barra de navegación, en el selector de regiones, selecciona una de las AWS regiones compatibles.

Important

Para utilizar servicios en la región de China (Pekín) o China (Ningxia), debe disponer de una cuenta y de credenciales para estas regiones. Las cuentas y credenciales de otras AWS regiones no funcionan en las regiones de Beijing y Ningxia, y viceversa.

La información sobre algunos recursos para las regiones de China, como los nombres de los paquetes del kit de CodeDeploy recursos y los procedimientos de instalación de los CodeDeploy agentes, no se incluye en esta edición de la Guía del CodeDeploy usuario.

Para obtener más información:

- [CodeDeploy en Cómo empezar AWS en la región de China \(Pekín\)](#)
- [CodeDeploy Guía del usuario para las regiones de China \(versión en inglés | versión en chino\)](#)

Los procedimientos de esta guía no coinciden con los de la CodeDeploy consola

Los procedimientos de esta guía se han diseñado para reflejar el nuevo diseño de la consola. Si utiliza la versión anterior de la consola, muchos de los conceptos y procedimientos básicos de esta guía se siguen aplicando. Para obtener acceso a ayuda en la nueva consola, elija el icono de información.

Los roles de IAM necesarios no están disponibles

Si utiliza un perfil de instancia de IAM o un rol de servicio que se creó como parte de una AWS CloudFormation pila, si elimina la pila, también se eliminarán todos los roles de IAM. Esta puede ser la razón por la que la función de IAM ya no se muestra en la consola de IAM y ya CodeDeploy no funciona como se esperaba. Para solucionar este problema, debe volver a crear manualmente el rol de IAM eliminado.

El uso de algunos editores de texto para crear AppSpec archivos y scripts de shell puede provocar errores en las implementaciones

Algunos editores de texto introducen en los archivos caracteres no imprimibles que no se admiten. Si utiliza editores de texto para crear o modificar AppSpec archivos o archivos shell script para ejecutarlos en instancias de Amazon Linux, Ubuntu Server o RHEL, cualquier implementación que dependa de estos archivos podría fallar. Cuando CodeDeploy utiliza estos archivos durante una implementación, la presencia de estos caracteres puede provocar errores en la validación de los hard-to-troubleshoot AppSpec archivos y en la ejecución de los scripts.

En la CodeDeploy consola, en la página de detalles del evento de la implementación, selecciona Ver registros. (O puede utilizar el comando AWS CLI para llamar al [get-deployment-instance](#) comando). Busque errores como `invalid character`, `command not found` o `file not found`.

Para solucionar este problema, le recomendamos lo siguiente:

- No utilice editores de texto que introduzcan caracteres no imprimibles, como devoluciones de `^M` carruajes (caracteres) en AppSpec los archivos y en los archivos shell script.
- Utilice editores de texto que muestren caracteres que no se impriman, como devoluciones de carruajes en sus AppSpec archivos y archivos con formato de libreta, de forma que pueda buscar y eliminar los que puedan estar introducidos. Para ver ejemplos de estos tipos de editores de texto, busque en Internet "el editor de texto muestra los retornos de carro".
- Utilice editores de texto que se ejecuten en instancias de Amazon Linux, Ubuntu Server o RHEL para crear archivos de script de intérpretes de comandos que se ejecuten en instancias de Amazon Linux, Ubuntu Server o RHEL. Para ver ejemplos de estos tipos de editores de texto, busque en Internet "editor de script de intérpretes de comandos de Linux".
- Si debe utilizar un editor de texto en Windows o macOS para crear archivos de script de intérpretes de comandos que se van a ejecutar en instancias de Amazon Linux, Ubuntu Server o RHEL, use un programa o una utilidad que convierta el texto en formato Windows o macOS en formato Unix. Para ver ejemplos de estos programas y utilidades, busque en Internet "de DOS a UNIX" o "de Mac a UNIX". Asegúrese de probar los archivos de intérpretes de comandos convertidos en los sistemas operativos de destino.

El uso de Finder en macOS para empaquetar una revisión de la aplicación puede producir un error en la implementación

Las implementaciones podrían fallar si utilizas la aplicación de interfaz gráfica de usuario (GUI) Finder de un Mac para empaquetar (comprimir) un AppSpec archivo y los archivos y scripts relacionados en un archivo de revisión de aplicaciones (.zip). Esto se debe a que Finder crea una carpeta `__MACOSX` intermedia en el archivo .zip y coloca archivos de componentes en ella. CodeDeploy no encuentra los archivos de los componentes, por lo que se produce un error en la implementación.

[Para solucionar este problema, le recomendamos que utilice el comando AWS CLI to call, que comprime los archivos de los componentes en la estructura esperada.](#) Otra opción es usar Terminal en lugar de la GUI para comprimir los archivos de componentes. Terminal no crea una carpeta `__MACOSX` intermedia.

Resolución de problemas de implementación de EC2/en las instalaciones

Temas

- [CodeDeploy error CommandPoller de falta de credenciales en el complemento](#)
- [La implementación produce un error con el mensaje “Validation of PKCS7 signed message failed”](#)
- [La implementación o la reimplementación de los mismos archivos en la misma instancia producen el error "The deployment failed because a specified file already exists at this location"](#)
- [Las rutas de archivo largas provocan el error “No existe tal archivo o directorio”](#)
- [Los procesos de larga duración pueden producir un error en la implementación](#)
- [Solución de problemas de un evento AllowTraffic de ciclo de vida fallido sin que se haya registrado ningún error en los registros de implementación](#)
- [Solución de problemas relacionados con un error o un evento del ciclo de vida de ApplicationStop la BeforeBlockTraffic implementación AfterBlockTraffic](#)
- [Solución de problemas de un evento fallido en el ciclo de vida de una DownloadBundle implementación con UnknownError: no abierto para lectura](#)
- [Solución de problemas cuando se omiten todos los eventos del ciclo de vida](#)
- [PowerShell Los scripts de Windows no utilizan la versión de 64 bits de Windows de forma predeterminada PowerShell](#)

Note

Las causas de muchos errores de implementación pueden identificarse revisando los archivos log creados durante el proceso de implementación. Para simplificar, recomendamos utilizar Amazon CloudWatch Logs para supervisar de forma centralizada los archivos de registro en lugar de verlos instancia por instancia. Para obtener más información, consulte [Ver CodeDeploy registros en la consola de CloudWatch registros](#).

Tip

Para obtener un manual que automatice muchas tareas de solución de problemas relacionadas con las implementaciones de EC2/on-premise, consulte - en [AWS Supportla referencia del](#) manual de automatización de TroubleshootCodeDeploy Systems Manager AWS .

CodeDeploy error CommandPoller de falta de credenciales en el complemento

Si aparece un error similar a

```
InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller: Missing credentials - please check if this instance was started with an IAM instance profile, podría deberse a una de las siguientes causas:
```

- La instancia en la que va a implementar no tiene un perfil de instancia de IAM asociado.
- El perfil de instancia de IAM no tiene configurados los permisos correctos.

Un perfil de instancia de IAM otorga al CodeDeploy agente permiso para comunicarse con su revisión CodeDeploy y descargarla de Amazon S3. Para las instancias EC2, consulte [Administración de identidades y accesos para AWS CodeDeploy](#). En el caso de las instancias locales, consulte [Working with On-Premises Instances](#).

La implementación produce un error con el mensaje “Validation of PKCS7 signed message failed”

Este mensaje de error indica que la instancia ejecuta una versión del CodeDeploy agente que solo admite el algoritmo hash SHA-1. Support para el algoritmo hash SHA-2 se introdujo en la versión 1.0.1.854 del CodeDeploy agente, publicada en noviembre de 2015. A partir del 17 de octubre de 2016, las implementaciones fallan si se instala una versión del CodeDeploy agente anterior a la 1.0.1.854. Para obtener más información, consulte para [cambiar AWS al algoritmo hash SHA256 para los certificados SSL](#), [NOTA: Retirar los agentes CodeDeploy host anteriores](#) a la versión 1.0.1.85 y [Actualice el CodeDeploy agente](#)

La implementación o la reimplementación de los mismos archivos en la misma instancia producen el error "The deployment failed because a specified file already exists at this location"

Si se CodeDeploy intenta implementar un archivo en una instancia, pero ya existe un archivo con el mismo nombre en la ubicación de destino especificada, es posible que se produzca un error en la implementación en esa instancia. Es posible que aparezca el mensaje de error "The deployment failed because a specified file already exists at this location: *nombre-ubicación*" (Se produjo un error en la implementación porque un archivo especificado ya existe en esta ubicación: nombre-ubicación). Esto se debe a que, durante cada implementación, CodeDeploy primero elimina todos los archivos de la implementación anterior, que aparecen en un archivo de registro de limpieza. Si hay archivos en las carpetas de instalación de destino que no aparecen en este archivo de limpieza, el CodeDeploy agente lo interpreta de forma predeterminada como un error y no se realiza la implementación.

Note

En las instancias de Amazon Linux, RHEL y Ubuntu Server, el archivo de limpieza se encuentra en `/opt/codedeploy-agent/deployment-root/deployment-instructions/`. En las instancias de Windows Server, la ubicación es `C:\ProgramData\Amazon\CodeDeploy\deployment-instructions\`.

La forma más sencilla de evitar este error es especificar una opción distinta del comportamiento de error predeterminado de la implementación. Para cada implementación, puede elegir si la

implementación produce un error, si se sobrescriben los archivos no incluidos en el archivo de limpieza o si se conservan los archivos que ya están en la instancia.

La opción de sobrescritura resulta útil cuando, por ejemplo, coloca manualmente un archivo en una instancia después de la última implementación, pero después añade un archivo con el mismo nombre a la siguiente revisión de la aplicación.

Puede elegir la opción de conservación para los archivos que coloca en la instancia que desea que formen parte de la siguiente implementación sin tener que añadirlos al paquete de revisión de la aplicación. La opción de conservación también resulta útil si los archivos de la aplicación ya se encuentran en el entorno de producción y desea implementarlos CodeDeploy por primera vez. Para obtener más información, consulte [Creación de una implementación en una plataforma de informática de EC2/en las instalaciones \(consola\)](#) y [Comportamiento de la restauración con contenido existente](#).

Solución de errores de implementación **The deployment failed because a specified file already exists at this location**

Si decide no especificar una opción para sobrescribir o conservar el contenido que se CodeDeploy detecte en las ubicaciones de despliegue de destino (o si no especifica ninguna opción de despliegue para gestionar el contenido existente en un comando programático), puede optar por solucionar el error.

Esta información se aplica únicamente si decide no conservar ni sobrescribir el contenido.

Si intenta volver a implementar archivos con los mismos nombres y ubicaciones, es más probable que la redeshule se realice correctamente si especifica el nombre de la aplicación y el grupo de despliegue con el mismo ID de grupo de despliegue subyacente que utilizó anteriormente. CodeDeploy utiliza el ID del grupo de despliegue subyacente para identificar los archivos que se van a eliminar antes de la redistribución.

La implementación de nuevos archivos o la reimplementación de los mismos archivos en las mismas ubicaciones de las instancias pueden producir un error por estas razones:

- Ha especificado un nombre de aplicación distinto para la reimplementación de la misma revisión en las mismas instancias. La nueva implementación produce un error porque aunque el nombre del grupo de implementaciones sea el mismo, el uso de un nombre de aplicación diferente significa que se está usando un ID de grupo de implementaciones subyacente distinto.
- Ha eliminado y ha vuelto a crear un grupo de implementaciones para una aplicación y, a continuación, ha intentado volver a implementar la misma revisión en el grupo de

implementaciones. La redistribución falla porque, aunque el nombre del grupo de implementación sea el mismo, CodeDeploy hace referencia a un ID de grupo de implementación subyacente diferente.

- Eliminó un grupo de aplicaciones y despliegues y CodeDeploy, a continuación, creó un nuevo grupo de aplicaciones y despliegues con los mismos nombres que los que eliminó. A continuación, ha intentado implementar de nuevo una revisión que se había implementado en el grupo de implementaciones anterior en el nuevo grupo con el mismo nombre. Se produce un error en la redistribución porque, aunque los nombres de la aplicación y del grupo de implementaciones son los mismos, CodeDeploy siguen haciendo referencia al ID del grupo de implementación que ha eliminado.
- Ha implementado una revisión en un grupo de implementaciones y después ha implementado la misma revisión en otro grupo de implementaciones en las mismas instancias. La segunda implementación falla porque CodeDeploy hace referencia a un ID de grupo de implementación subyacente diferente.
- Ha implementado una revisión en un grupo de implementaciones y después ha implementado otra revisión en otro grupo de implementaciones en las mismas instancias. Hay al menos un archivo con el mismo nombre y en la misma ubicación en la que el segundo grupo de implementaciones intenta realizar la implementación. La segunda implementación falla porque CodeDeploy no elimina el archivo existente antes de que comience la segunda implementación. Ambas implementaciones hacen referencia a identificadores de grupo de implementaciones diferentes.
- Ha implementado una revisión en CodeDeploy, pero hay al menos un archivo con el mismo nombre y en la misma ubicación. La implementación falla porque, de forma predeterminada, CodeDeploy no elimina el archivo existente antes de que comience la implementación.

Para solucionar estas situaciones, proceda del modo siguiente:

- Elimine los archivos de las ubicaciones e instancias en las que aplicaron anteriormente implementaciones e intente realizar de nuevo la implementación.
- En el AppSpec archivo de la revisión, ya sea en los eventos del ciclo de vida ApplicationStop o del BeforeInstall despliegue, especifique un script personalizado para eliminar los archivos en cualquier ubicación que coincida con los archivos que la revisión va a instalar.
- Implemente o reimplemente los archivos en ubicaciones o instancias que no formaron parte de implementaciones anteriores.
- Antes de eliminar una aplicación o un grupo de despliegues, implementa una revisión que contenga un AppSpec archivo que especifique que no hay archivos que copiar en las instancias.

Para la implementación, especifique el nombre de aplicación y el nombre del grupo de implementaciones que usan los mismos identificadores de aplicación y grupo de implementaciones que los que está a punto de eliminar. (Puede usar el [get-deployment-group](#) comando para recuperar el ID del grupo de implementación). CodeDeploy utiliza el ID y el AppSpec archivo del grupo de despliegue subyacentes para eliminar todos los archivos que instaló en la anterior implementación exitosa.

Las rutas de archivo largas provocan el error “No existe tal archivo o directorio”

En el caso de las implementaciones en instancias de Windows, si tiene una ruta de archivo de más de 260 caracteres en la sección de archivos del archivo `appspect.yml`, es posible que las implementaciones fallen y se produzca un error similar al siguiente:

```
No such file or directory @ dir_s_mkdir - C:\your-long-file-path
```

Este error se produce porque, de forma predeterminada, Windows no permite rutas de archivos de más de 260 caracteres, como se detalla en la [documentación de Microsoft](#).

Para las versiones 1.4.0 o posteriores del CodeDeploy agente, puede habilitar las rutas de archivo largas de dos maneras, según el proceso de instalación del agente:

Si el CodeDeploy agente aún no se ha instalado:

1. En el equipo en el que va a instalar el CodeDeploy agente, active la clave de registro de `LongPathsEnabled` Windows mediante este comando:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"  
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Instale el CodeDeploy agente. Para obtener más información, consulte [Instala el CodeDeploy agente](#).

Si el CodeDeploy agente ya se ha instalado:

1. En el equipo CodeDeploy agente, habilite la clave de registro de `LongPathsEnabled` Windows mediante este comando:

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem"
```

```
-Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

2. Reinicie el CodeDeploy agente para que el cambio de clave de registro surta efecto. Para reiniciar el agente, utilice este comando:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Los procesos de larga duración pueden producir un error en la implementación

En el caso de las implementaciones en instancias de Amazon Linux, Ubuntu Server y RHEL, si tiene un script de implementación que inicia un proceso de ejecución prolongada, es CodeDeploy posible que pase mucho tiempo esperando durante el evento del ciclo de vida de la implementación y, luego, no se realice la implementación. Esto se debe a que si el proceso se ejecuta durante más tiempo del que se espera que los procesos en primer plano y en segundo plano, en ese caso la implementación CodeDeploy se detenga o falle, incluso si el proceso sigue ejecutándose según lo previsto.

Suponga, por ejemplo, que la revisión de la aplicación contiene dos archivos en su raíz, `after-install.sh` y `sleep.sh`. Su AppSpec archivo contiene las siguientes instrucciones:

```
version: 0.0
os: linux
files:
  - source: ./sleep.sh
    destination: /tmp
hooks:
  AfterInstall:
    - location: after-install.sh
      timeout: 60
```

El `after-install.sh` archivo se ejecuta durante el evento del ciclo de vida de la `AfterInstall` aplicación. Este es su contenido:

```
#!/bin/bash
/tmp/sleep.sh
```

El archivo `sleep.sh` contiene lo siguiente, que suspende la ejecución del programa durante tres minutos (180 segundos), lo que simula un proceso de larga duración:


```
#!/bin/bash
sleep 180
```

Cuando `after-install.sh` llamas `sleep.sh`, se `sleep.sh` inicia y se ejecuta durante tres minutos (180 segundos), es decir, dos minutos (120 segundos) después de la hora prevista `sleep.sh` (y, en relación con ello, `after-install.sh`) en la CodeDeploy que dejará de ejecutarse. Tras el tiempo de espera de un minuto (60 segundos), CodeDeploy se detiene y se produce un error en el momento del ciclo de vida de la AfterInstall aplicación, aunque `sleep.sh` sigue ejecutándose según lo previsto. Se muestra el siguiente error:

```
Script at specified location: after-install.sh failed to complete in 60
seconds.
```

No basta con añadir un signo de ampersand (&) en `after-install.sh` para ejecutar `sleep.sh` en segundo plano.

```
#!/bin/bash
# Do not do this.
/tmp/sleep.sh &
```

Si lo hace, puede dejar la implementación en un estado pendiente durante el tiempo de espera predeterminado de una hora para el evento del ciclo de vida de la implementación, después del cual, CodeDeploy se detiene y falla la implementación en el evento del ciclo de vida de la AfterInstall aplicación, como antes.

En `after-install.sh`, llame de la `sleep.sh` siguiente manera, lo que le permitirá CodeDeploy continuar una vez que el proceso comience a ejecutarse:

```
#!/bin/bash
/tmp/sleep.sh > /dev/null 2> /dev/null < /dev/null &
```

En la llamada anterior, `sleep.sh` es el nombre del proceso que desea empezar a ejecutar en segundo plano, redirigiendo `stdout`, `stderr` y `stdin` a `/dev/null`.

Solución de problemas de un evento AllowTraffic de ciclo de vida fallido sin que se haya registrado ningún error en los registros de implementación

En algunos casos, una implementación azul/verde falla durante el AllowTraffic ciclo de vida, pero los registros de implementación no indican la causa del error.

Este error se debe normalmente a que las comprobaciones de estado se han configurado incorrectamente en Elastic Load Balancing para la instancia del Equilibrador de carga clásico, del Equilibrador de carga de aplicación o del Equilibrador de carga de red usada para administrar el tráfico del grupo de implementación.

Para resolver el problema, revise y corrija los errores en la configuración de la comprobación de estado del balanceador de carga.

Para los balanceadores de carga clásicos, consulte [Configurar comprobaciones de estado](#) en la Guía del usuario de Classic Load Balancers y [ConfigureHealthCheck](#) en la versión de referencia de la API de Elastic Load Balancing, versión 2012-06-01.

Para los equilibradores de carga de aplicación, consulte [Comprobaciones de estado de los grupos de destino](#) en la Guía del usuario de los equilibradores de carga de aplicación.

Para los equilibradores de carga de red, consulte [Comprobaciones de estado de los grupos de destino](#) en la Guía del usuario de los equilibradores de carga de red.

Solución de problemas relacionados con un error o un evento del ciclo de vida de ApplicationStop la BeforeBlockTraffic implementación AfterBlockTraffic

Durante una implementación, el CodeDeploy agente ejecuta los scripts especificados para ApplicationStop y AfterBlockTraffic en el AppSpec archivo de la implementación anterior exitosa. BeforeBlockTraffic (Todos los demás scripts se ejecutan desde el AppSpec archivo de la implementación actual). Si alguno de estos scripts contiene un error y no se ejecuta correctamente, la implementación puede producir un error.

Los posibles motivos de estos errores son los siguientes:

- El CodeDeploy agente encuentra el *deployment-group-id_last_successful_install* archivo en la ubicación correcta, pero la ubicación que aparece en el *deployment-group-id_last_successful_install* archivo no existe.

En las instancias de Amazon Linux, Ubuntu Server y RHEL, este archivo debe existir en `/opt/codedeploy-agent/deployment-root/deployment-instructions`.

En las instancias de Windows Server, este archivo se debe almacenar en la carpeta C : `\ProgramData\Amazon\CodeDeploy\deployment-instructions`.

- En la ubicación que aparece en el `deployment-group-id_last_successful_install` archivo, el AppSpec archivo no es válido o los scripts no se ejecutan correctamente.
- El script contiene un error que no se puede corregir, por lo que nunca se ejecuta correctamente.

Utilice la CodeDeploy consola para investigar por qué una implementación podría haber fallado durante alguno de estos eventos. En la página de detalles de la implementación, elija View events (Ver eventos). En la página de detalles de la instancia, en la ApplicationStopAfterBlockTraffic fila o, selecciona Ver registros. BeforeBlockTraffic O usa el AWS CLI para llamar al [get-deployment-instance](#) comando.

Si la causa del error es un script de la última implementación correcta que nunca se ejecutó correctamente, cree una implementación y especifique que se deben ignorar AfterBlockTraffic los errores y. ApplicationStop BeforeBlockTraffic Hay dos formas de hacer esto:

- Utilice la CodeDeploy consola para crear una implementación. En la página Crear despliegue, en Fallo de un evento ApplicationStop del ciclo de vida, selecciona No fallar el despliegue en una instancia si este evento del ciclo de vida de la instancia falla.
- Use AWS CLI para llamar al [create-deployment](#) comando e incluir la `--ignore-application-stop-failures` opción.

Cuando vuelva a implementar la revisión de la aplicación, la implementación continuará aunque algunos de estos eventos de ciclo de vida produzcan un error. Si la nueva revisión incluye scripts fijos para los eventos de ciclo de vida, puede que las implementaciones futuras se realicen correctamente si aplica esta solución.

Solución de problemas de un evento fallido en el ciclo de vida de una DownloadBundle implementación con UnknownError: no abierto para lectura

Si está intentando implementar una revisión de la aplicación desde Amazon S3 y la implementación falla durante el evento del ciclo de vida de la DownloadBundle implementación con el siguiente UnknownError: not opened for reading error:

- Se ha producido un error interno del servicio de Amazon S3. Implemente de nuevo la revisión de la aplicación.

- El perfil de instancia de IAM de su instancia de Amazon S3 no tiene permisos para obtener acceso a la revisión de la aplicación en Amazon S3. Para obtener información sobre las políticas de buckets de Amazon S3, consulte [Enviar una revisión CodeDeploy a Amazon S3 \(solo para implementaciones de EC2/on-premise\)](#) y [Requisitos previos para una implementación](#).
- Las instancias en las que despliega están asociadas a una AWS región (por ejemplo, EE.UU. Oeste (Oregón)), pero el bucket de Amazon S3 que contiene la revisión de la aplicación está asociado a otra AWS región (por ejemplo, EE.UU. Este (Norte de Virginia)). Asegúrese de que la revisión de la aplicación esté en un bucket de Amazon S3 asociado a la misma AWS región que las instancias.

En la página de detalles del evento de la implementación, en la fila Download bundle (Descargar paquete), elija View logs (Ver registros). O utilice el AWS CLI para llamar al [get-deployment-instance](#) comando. Si se produce un error, debería aparecer un error en el resultado con el código de error `UnknownError` y el mensaje de error `not opened for reading`.

Para determinar la causa de este error:

1. Habilite el registro en red en al menos una de las instancias y, a continuación, implemente de nuevo la revisión de la aplicación.
2. Examine el archivo de registro en red para encontrar el error. Los mensajes de error comunes para este problema incluyen la frase "access denied".
3. Después de examinar los archivos de registro, le recomendamos que deshabilite el registro en red para reducir el tamaño de los archivos de registro y la cantidad de información confidencial que puede aparecer en la salida en texto sin formato en la instancia en el futuro.

Para obtener información sobre cómo encontrar el archivo de registro de cables y habilitar y deshabilitar el registro de cables, consulte `:log_aws_wire:` la [referencia de configuración del CodeDeploy agente](#).

Solución de problemas cuando se omiten todos los eventos del ciclo de vida

Si todos los eventos del ciclo de vida de una implementación de EC2 o en las instalaciones se omiten, es posible que reciba un error similar a `The overall deployment failed because too many individual instances failed deployment, too few healthy instances are available for deployment, or some instances in your deployment group`

are experiencing problems. (Error code: HEALTH_CONSTRAINTS). Estas son algunas posibles causas y soluciones:

- Es posible que el CodeDeploy agente no esté instalado o en ejecución en la instancia. Para determinar si el CodeDeploy agente se está ejecutando:
 - Para un servidor de Amazon Linux RHEL o Ubuntu, ejecute lo siguiente:

```
systemctl status codedeploy-agent
```

- Para Windows, ejecute lo siguiente:

```
powershell.exe -Command Get-Service -Name CodeDeployagent
```

Si el CodeDeploy agente no está instalado o en ejecución, consulte [Compruebe que el CodeDeploy agente se esté ejecutando](#).

Es posible que tu instancia no pueda llegar al punto de conexión público CodeDeploy o a Amazon S3 mediante el puerto 443. Pruebe con una de las siguientes acciones:

- Asigne una dirección IP pública a la instancia y utilice su tabla de ruteo para permitir el acceso a Internet. Asegúrese de que el grupo de seguridad asociado con la instancia permite el acceso de salida a través del puerto 443 (HTTPS). Para obtener más información, consulte [Protocolo de comunicación y puerto para el CodeDeploy agente](#).
- Si una instancia se aprovisiona en una subred privada, utilice una gateway NAT en lugar de una gateway de Internet en la tabla de ruteo. Para obtener más información, consulte [Puerta de enlace NAT](#).
- CodeDeploy Es posible que el rol de servicio no tenga los permisos necesarios. Para configurar un rol de servicio de CodeDeploy, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).
- Si utiliza un proxy HTTP, asegúrese de que esté especificado en la `:proxy_uri:` configuración del archivo de configuración del CodeDeploy agente. Para obtener más información, consulte [CodeDeploy referencia de configuración del agente](#).
- Es posible que la firma de fecha y hora de la firma de su instancia de implementación no coincida con la firma de fecha y hora de la solicitud de implementación. Busque un error similar al del archivo `Cannot reach InstanceService: Aws::CodeDeployCommand::Errors::InvalidSignatureException - Signature expired` de registro del CodeDeploy agente. Si ve este error, siga los pasos que se indican en [Solución de errores de despliegue «InvalidSignatureException — Firma caducada: \[hora\]](#)

[es anterior a \[hora\]»](#). Para obtener más información, consulte [Vea los datos de registro de las implementaciones locales CodeDeploy de EC2/](#).

- Es posible que el CodeDeploy agente deje de ejecutarse porque una instancia se está quedando sin memoria o espacio en el disco duro. Intenta reducir el número de implementaciones archivadas en la instancia actualizando la `max_revisions` configuración del CodeDeploy agente. Si hace esto para una instancia de EC2 y el problema continúa, considere la posibilidad de utilizar una instancia más grande. Por ejemplo, si el tipo de instancia es `t2.small`, intente usar una `t2.medium`. Para obtener más información, consulte [Archivos instalados por el agente CodeDeploy](#) , [CodeDeploy referencia de configuración del agente](#) y [Tipos de instancias](#).
- Es posible que la instancia que va a implementar no tenga un perfil de instancia de IAM asociado o que tenga un perfil de instancia de IAM asociado que no tenga los permisos necesarios.
 - Si un perfil de instancia de IAM no está asociado a la instancia, cree uno con los permisos necesarios y, a continuación, asócielo a la instancia.
 - Si un perfil de instancia de IAM ya está asociado a la instancia, asegúrese de que dispone de los permisos necesarios.

Después de confirmar que el perfil de instancia asociado está configurado con los permisos necesarios, reinicie la instancia. Para obtener más información, consulte [Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2](#) y [Roles de IAM para Amazon EC2](#) en la Guía del usuario de Amazon EC2.

PowerShell Los scripts de Windows no utilizan la versión de 64 bits de Windows de forma predeterminada PowerShell

Si un PowerShell script de Windows que se ejecuta como parte de una implementación utiliza una funcionalidad de 64 bits (por ejemplo, porque consume más memoria de la que permite una aplicación de 32 bits o porque llama a bibliotecas que solo se ofrecen en una versión de 64 bits), es posible que el script se bloquee o no se ejecute según lo esperado. Esto se debe a que, de forma predeterminada, CodeDeploy utiliza la versión de 32 bits de Windows PowerShell para ejecutar los PowerShell scripts de Windows que forman parte de la revisión de una aplicación.

Agregue un código como el siguiente al principio de cualquier script que deba ejecutarse con la versión de 64 bits de Windows PowerShell:

```
# Are you running in 32-bit mode?
# (\SysWOW64\ = 32-bit mode)
```

```
if ($PSHOME -like "*SysWOW64*")
{
    Write-Warning "Restarting this script under 64-bit Windows PowerShell."

    # Restart this script under 64-bit Windows PowerShell.
    # (\SysNative\ redirects to \System32\ for 64-bit mode)

    & (Join-Path ($PSHOME -replace "SysWOW64", "SysNative") powershell.exe) -File `
        (Join-Path $PSScriptRoot $MyInvocation.MyCommand) @args

    # Exit 32-bit script.

    Exit $LastExitCode
}

# Was restart successful?
Write-Warning "Hello from $PSHOME"
Write-Warning " (\SysWOW64\ = 32-bit mode, \System32\ = 64-bit mode)"
Write-Warning "Original arguments (if any): $args"

# Your 64-bit script code follows here...
# ...
```

Si bien la información sobre la ruta de archivo de este código puede parecer contradictoria, Windows de 32 bits PowerShell usa una ruta como la siguiente:

```
c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
```

Windows de 64 bits PowerShell usa una ruta como la siguiente:

```
c:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Resolución problemas de implementación de Amazon ECS

Temas

- [Se agota el tiempo de espera mientras se espera a que se establezca la tarea de reemplazo](#)
- [Se agota el tiempo de espera mientras se espera a que continúe una notificación](#)
- [El rol de IAM no tiene permisos suficientes](#)
- [Se agotó el tiempo de espera de la implementación mientras se esperaba una llamada de estado](#)

- [La implementación falló porque una o más de las funciones de validación de eventos del ciclo de vida fallaron](#)
- [No se pudo actualizar el ELB debido al siguiente error: el grupo objetivo del conjunto de tareas principal debe estar detrás del oyente](#)
- [Mi implementación a veces falla cuando uso el escalado automático](#)
- [Solo ALB admite el enrutamiento de tráfico gradual; en su lugar, utilice el enrutamiento de AllAtOnce tráfico cuando cree o actualice un grupo de implementación](#)
- [A pesar de que mi implementación se realizó correctamente, el conjunto de tareas de reemplazo no pasa las comprobaciones de estado de Elastic Load Balancing y mi aplicación no funciona](#)
- [¿Puedo asociar varios equilibradores de carga a un grupo de implementación?](#)
- [¿Puedo realizar despliegues CodeDeploy azules o verdes sin un balanceador de carga?](#)
- [¿Cómo puedo actualizar mi servicio de Amazon ECS con nueva información durante una implementación?](#)

Se agota el tiempo de espera mientras se espera a que se establezca la tarea de reemplazo

Problema: aparece el siguiente mensaje de error al implementar la aplicación Amazon ECS mediante CodeDeploy:

```
The deployment timed out while waiting for the replacement task set to become healthy. This time out period is 60 minutes.
```

Causa posible: este error puede producirse si hay un error en el archivo de definición de tareas o en otros archivos relacionados con la implementación. Por ejemplo, si hay un error tipográfico en el campo `image` del archivo de definición de tareas, Amazon ECS intentará obtener la imagen del contenedor incorrecta y fallará continuamente, lo que provocará este error.

Posibles soluciones y próximos pasos:

- Corrija los errores tipográficos y los problemas de configuración en el archivo de definición de tareas y en otros archivos.
- Consulte el evento relacionado con el servicio de Amazon ECS y descubra por qué las tareas de reemplazo no funcionan correctamente. Para obtener más información sobre los eventos de Amazon ECS, consulte [Eventos de Amazon ECS](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

- Consulte la sección de [solución de problemas de Amazon ECS](#) de la Guía para desarrolladores de Amazon Elastic Container Service para ver los errores relacionados con los mensajes del evento.

Se agota el tiempo de espera mientras se espera a que continúe una notificación

Problema: aparece el siguiente mensaje de error al implementar la aplicación Amazon ECS mediante CodeDeploy:

```
The deployment timed out while waiting for a notification to continue. This time out period is n minutes.
```

Causa posible: este error puede producirse si especificó un tiempo de espera en el campo Especifique cuándo redirigir el tráfico al crear el grupo de implementación, pero la implementación no pudo finalizar antes de que expirara el tiempo de espera.

Posibles soluciones y próximos pasos:

- En su grupo de implementación, establezca la opción Especificar cuándo redirigir el tráfico en un período de tiempo mayor y vuelva a realizar la implementación. Para obtener más información, consulte [Creación de un grupo de implementación para una implementación de Amazon ECS \(consola\)](#).
- En su grupo de implementación, cambie Especificar cuándo redirigir el tráfico a Reenrutar el tráfico inmediatamente y vuelva a implementar. Para obtener más información, consulte [Creación de un grupo de implementación para una implementación de Amazon ECS \(consola\)](#).
- Vuelva a implementar y, a continuación, ejecute el [aws deploy continue-deployment](#) AWS CLI comando con la `--deployment-wait-type` opción establecida en `READY_WAIT`. Asegúrese de ejecutar este comando antes de que venza el tiempo especificado en Especificar cuándo redirigir el tráfico.

El rol de IAM no tiene permisos suficientes

Problema: aparece el siguiente mensaje de error al implementar la aplicación Amazon ECS mediante CodeDeploy:

```
The IAM role role-arn does not give you permission to perform operations in the following AWS service: AWSLambda.
```

Causa posible: este error puede producirse si especificó una función Lambda en la [Hookssección del AppSpec archivo](#), pero no otorgó CodeDeploy permiso al servicio Lambda.

Solución posible: añadir el `lambda:InvokeFunction` permiso a la función de CodeDeploy servicio. Para añadir este permiso, añada al rol una de las siguientes políticas administradas por AWS:

`AWSCodeDeployRoleForECS` o **`AWSCodeDeployRoleForECSLimited`**. Para obtener información sobre estas políticas y cómo añadirlas al rol CodeDeploy de servicio, consulte [Paso 2: Crear un rol de servicio para CodeDeploy](#).

Se agotó el tiempo de espera de la implementación mientras se esperaba una llamada de estado

Problema: aparece el siguiente mensaje de error al implementar la aplicación Amazon ECS mediante CodeDeploy:

```
The deployment timed out while waiting for a status callback. CodeDeploy expects a status callback within one hour after a deployment hook is invoked.
```

Causa posible: este error puede producirse si especificó una función de Lambda en la [Hookssección del AppSpec archivo](#), pero la función de Lambda no pudo llamar a la `PutLifecycleEventHookExecutionStatus` API necesaria para devolver un `Succeeded` estado o `Failed` CodeDeploy

Posibles soluciones y próximos pasos:

- Agregue el `codedeploy:putlifecycleEventHookExecutionStatus` permiso a la función de ejecución de Lambda utilizada por la función de Lambda que especificó en el archivo. AppSpec Este permiso otorga a la función Lambda la capacidad de devolver un estado igual `Succeeded` o `Failed` igual a CodeDeploy Para obtener más información sobre el rol de ejecución de Lambda, consulte [Rol de ejecución de Lambda](#) en la Guía para desarrolladores de AWS Lambda .
- Compruebe el código de la función Lambda y los registros de ejecución para asegurarse de que su función Lambda está llamando a la `PutLifecycleEventHookExecutionStatus` API para informar CodeDeploy sobre si la prueba de validación CodeDeploy del ciclo de vida o. `Succeeded` `Failed` Para obtener información sobre la `putlifecycleEventHookExecutionStatus` API, consulte la referencia [PutLifecycleEventHookExecutionStatus](#) de la AWS CodeDeploy API. Para obtener información sobre los registros de ejecución de Lambda, consulte [Acceder a los CloudWatch registros de Amazon](#) para. AWS Lambda

La implementación falló porque una o más de las funciones de validación de eventos del ciclo de vida fallaron

Problema: aparece el siguiente mensaje de error al implementar la aplicación Amazon ECS mediante CodeDeploy:

```
The deployment failed because one or more of the lifecycle event validation functions failed.
```

Causa posible: este error puede producirse si especificó una función Lambda en la [Hookssección del AppSpec archivo](#), pero la función Lambda regresó Failed a CodeDeploy ella cuando se llamó. PutLifecycleEventHookExecutionStatus Este error indica CodeDeploy que la prueba de validación del ciclo de vida ha fallado.

Posible paso siguiente: compruebe los registros de ejecución de Lambda para ver por qué falla el código de la prueba de validación. Para obtener información sobre los registros de ejecución de Lambda, consulte [Acceder a los CloudWatch registros de Amazon](#) para AWS Lambda

No se pudo actualizar el ELB debido al siguiente error: el grupo objetivo del conjunto de tareas principal debe estar detrás del oyente

Problema: aparece el siguiente mensaje de error al implementar la aplicación Amazon ECS mediante CodeDeploy:

```
The ELB could not be updated due to the following error: Primary taskset target group must be behind listener
```

Causa posible: este error puede producirse si ha configurado un oyente de prueba opcional y está configurado con un grupo de destino incorrecto. Para obtener más información sobre el detector de pruebas en CodeDeploy, consulte [Antes de empezar una implementación de Amazon ECS](#) y [¿Qué sucede durante una implementación de Amazon ECS?](#). Para obtener más información sobre los conjuntos de tareas, consulte [TaskSet](#) la referencia de la API de Amazon Elastic Container Service y [describe-task-set](#) la sección Amazon ECS de la referencia de AWS CLI comandos.

Posible solución: asegúrese de que tanto el oyente de producción como el oyente de prueba de Elastic Load Balancing apunten al grupo de destino que actualmente atiende sus cargas de trabajo. Hay tres lugares para comprobarlo:

- En Amazon EC2, en la configuración de Oyentes y reglas del equilibrador de carga. Para obtener más información, consulte [Oyentes para equilibradores de carga de aplicación](#) en la Guía de

equilibradores de carga de aplicación y [Oyentes para equilibradores de carga de red](#) en la Guía de equilibradores de carga de red.

- En Amazon ECS, en su clúster, en la configuración de Red de su servicio. Para obtener más información, consulte [Consideraciones sobre el Equilibrador de carga de aplicación y el Equilibrador de carga de red](#) en la Guía del usuario de Amazon Elastic Container Service
- En CodeDeploy, en la configuración de su grupo de despliegues. Para obtener más información, consulte [Creación de un grupo de implementación para una implementación de Amazon ECS \(consola\)](#).

Mi implementación a veces falla cuando uso el escalado automático

Problema: Está utilizando Auto Scaling con CodeDeploy y observa que sus despliegues a veces fallan. Para obtener más información acerca de los síntomas de este problema, consulte el tema que dice [Para los servicios configurados para utilizar el escalado automático del servicio y el tipo de implementación azul/verde, el escalado automático no se bloquea durante una implementación, pero la implementación puede fallar en algunas circunstancias](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Causa posible: este problema puede producirse si CodeDeploy los procesos de Auto Scaling entran en conflicto.

Posible solución: Suspenda y reanude los procesos de Auto Scaling durante el CodeDeploy despliegue mediante la RegisterScalableTarget API (o el register-scalable-target AWS CLI comando correspondiente). Para obtener más información, consulte [Suspender y reanudar el escalado para el escalado automático de aplicaciones](#) en la Guía del usuario de escalado automático de aplicaciones.

Note

CodeDeploy no puede llamar RegisterScaleableTarget directamente. Para utilizar esta API, debe configurarla CodeDeploy para enviar una notificación o un evento a Amazon Simple Notification Service (o Amazon CloudWatch). A continuación, debe configurar Amazon SNS (o CloudWatch) para que llame a una función de Lambda y configurar la función de Lambda para que llame a la API. RegisterScalableTarget Se debe llamar a la API de RegisterScalableTarget con el parámetro SuspendedState establecido en true para suspender las operaciones de escalado automático, y en false para reanudarlas.

La notificación o el evento que se CodeDeploy envía debe producirse cuando se inicia un despliegue (para activar la suspensión de las operaciones de Auto Scaling) o cuando un despliegue tiene éxito, falla o se detiene (para activar la reanudación de las operaciones de Auto Scaling).

Para obtener información sobre cómo CodeDeploy configurar la generación de notificaciones o CloudWatch eventos de Amazon SNS, consulte [Supervisión de despliegues con Amazon Events CloudWatch](#) . y. [Monitoring Deployments with Amazon SNS Event Notifications](#)

Solo ALB admite el enrutamiento de tráfico gradual; en su lugar, utilice el enrutamiento de AllAtOnce tráfico cuando cree o actualice un grupo de implementación

Problema: aparece el siguiente mensaje de error al crear o actualizar un grupo de implementación en: CodeDeploy

```
Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing instead when you create/update Deployment group.
```

Causa posible: este error puede producirse si utiliza un equilibrador de carga de red e intenta utilizar una configuración de implementación predefinida distinta de `CodeDeployDefault.ECSAllAtOnce`.

Soluciones posibles:

- Cambie la configuración de implementación predefinida a `CodeDeployDefault.ECSAllAtOnce`. Esta es la única configuración de implementación admitida por los equilibradores de carga de red.

Para obtener más información acerca de las configuraciones de implementación predefinidas, consulte [Configuraciones de implementación predefinidas para una plataforma de informática de Amazon ECS](#).

- Cambie el equilibrador de carga a un equilibrador de carga de red. Los equilibradores de carga de aplicación admiten todas las configuraciones de implementación predefinidas. Para obtener más información acerca de cómo crear un equilibrador de carga de aplicación, consulte [Configure un balanceador de carga, grupos objetivo y agentes de escucha para las implementaciones de CodeDeploy Amazon ECS](#).

A pesar de que mi implementación se realizó correctamente, el conjunto de tareas de reemplazo no pasa las comprobaciones de estado de Elastic Load Balancing y mi aplicación no funciona

Problema: aunque CodeDeploy indica que mi implementación se ha realizado correctamente, el conjunto de tareas de reemplazo no supera las comprobaciones de estado de Elastic Load Balancing y mi aplicación no funciona.

Causa posible: este problema puede producirse si ha realizado una CodeDeploy all-at-once implementación y el conjunto de tareas de reemplazo (verde) contiene un código incorrecto que provoca que las comprobaciones de estado de Elastic Load Balancing no se realicen correctamente. Con la configuración de all-at-once despliegue, las comprobaciones de estado del balanceador de cargas comienzan a ejecutarse en el conjunto de tareas de reemplazo una vez que el tráfico se ha transferido a él (es decir, después de que se produzca el evento CodeDeploy del AllowTraffic ciclo de vida). Por este motivo, observará que las comprobaciones de estado no se superarán en el conjunto de tareas de reemplazo después de desviar el tráfico, pero no antes. Para obtener información sobre los eventos del ciclo de vida que se CodeDeploy generan, consulte [¿Qué sucede durante una implementación de Amazon ECS?](#).

Soluciones posibles:

- Cambie la configuración de despliegue all-at-once a canaria o lineal. En una configuración lineal o lineal, las comprobaciones de estado del balanceador de cargas comienzan a ejecutarse en el conjunto de tareas de reemplazo mientras se CodeDeploy instala la aplicación en el entorno de reemplazo y antes de que se desplace el tráfico (es decir, durante el evento Install del ciclo de vida y antes del AllowTraffic evento). Al permitir que las comprobaciones se ejecuten durante la instalación de la aplicación, pero antes de que se desvíe el tráfico, se detectará un código incorrecto de la aplicación y se producirán errores de implementación antes de que la aplicación esté disponible públicamente.

Para obtener más información acerca de cómo configurar implementaciones de valor controlado o lineales, consulte [Cambie la configuración del grupo de implementación con CodeDeploy](#).

Para obtener información sobre los eventos CodeDeploy del ciclo de vida que se ejecutan durante una implementación de Amazon ECS, consulte [¿Qué sucede durante una implementación de Amazon ECS?](#).

Note

Las configuraciones de implementación lineal y de valor controlado solo son compatibles con los equilibradores de carga de aplicación.

- Si desea conservar la configuración de la all-at-once implementación, configure un detector de pruebas y compruebe el estado de la tarea de reemplazo establecida con el enlace de `BeforeAllowTraffic` ciclo de vida. Para obtener más información, consulte [Lista de enlaces de eventos de ciclo de vida para una implementación de Amazon ECS](#).

¿Puedo asociar varios equilibradores de carga a un grupo de implementación?

No. Si desea utilizar varios balanceadores de carga de aplicaciones o balanceadores de carga de red, utilice las actualizaciones sucesivas de Amazon ECS en lugar de las implementaciones CodeDeploy azul/verde. Para obtener más información acerca de las actualizaciones continuas, consulte [Actualización continua](#) en la Guía para desarrolladores de Amazon Elastic Container Service. Para obtener más información acerca del uso de varios equilibradores de carga con Amazon ECS, consulte [Registro de varios grupos de destino con un servicio](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

¿Puedo realizar despliegues CodeDeploy azules o verdes sin un balanceador de carga?

No, no puedes realizar despliegues CodeDeploy azules o verdes sin un balanceador de carga. Si no puede utilizar un equilibrador de carga, utilice en su lugar la característica de actualizaciones continuas de Amazon ECS. Para obtener más información acerca de las actualizaciones continuas de Amazon ECS, consulte [Actualización continua](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

¿Cómo puedo actualizar mi servicio de Amazon ECS con nueva información durante una implementación?

Para CodeDeploy actualizar su servicio Amazon ECS con un parámetro nuevo mientras lleva a cabo una implementación, especifique el parámetro en la `resources` sección del AppSpec archivo. Solo se admiten algunos parámetros de Amazon ECS CodeDeploy, como el archivo de definición

de tareas y los parámetros del nombre del contenedor. Para obtener una lista completa de los parámetros de Amazon ECS que CodeDeploy se pueden actualizar, consulte [AppSpec sección de «recursos» para las implementaciones de Amazon ECS](#).

Note

Si necesita actualizar su servicio Amazon ECS con un parámetro que no es compatible con CodeDeploy, complete estas tareas:

1. Llame a la API `UpdateService` de Amazon ECS con el parámetro que desee actualizar. Para obtener una lista completa de los parámetros que se pueden actualizar, consulte la referencia [UpdateService](#) de la API de Amazon Elastic Container Service.
2. Para aplicar el cambio a las tareas, cree una nueva implementación azul/verde de Amazon ECS. Para obtener más información, consulte [Creación de una implementación en una plataforma de informática de Amazon ECS \(consola\)](#).

Solución de problemas de implementación de AWS Lambda

Temas

- [AWS Lambda las implementaciones fallan después de detener manualmente una implementación de Lambda que no tiene reversiones configuradas](#)

AWS Lambda las implementaciones fallan después de detener manualmente una implementación de Lambda que no tiene reversiones configuradas

En algunos casos, el alias de una función de Lambda especificada en una implementación podría hacer referencia a dos versiones diferentes del función. Como consecuencia, no se realizan correctamente los intentos posteriores de implementar la función de Lambda. Una implementación de Lambda puede entrar en este estado cuando no tiene configuradas las reversiones y se detiene manualmente. Para continuar, utilice la AWS Lambda consola para asegurarse de que la función no esté configurada para cambiar el tráfico entre dos versiones:

1. Inicie sesión en la AWS Lambda consola AWS Management Console y ábrala en <https://console.aws.amazon.com/lambda/>.

2. En el panel izquierdo, elija **Functions** (Funciones).
3. Seleccione el nombre de la función Lambda que se encuentra en la implementación. CodeDeploy
4. En **Alias**, elija el alias utilizado en la CodeDeploy implementación y, a continuación, elija **Editar**.
5. En **Alias ponderado**, elija **none**. Esto garantiza que el alias no esté configurado para desviar un porcentaje (o peso) del tráfico a más de una versión. Anote la versión seleccionada en **Version** (Versión).
6. Seleccione **Guardar**.
7. Abra la CodeDeploy consola e intente implementar la versión que se muestra en el menú desplegable en el paso 5.

Solución de problemas de grupos de implementación

Cuando se etiqueta una instancia como parte de un grupo de implementaciones, no se implementa automáticamente la aplicación en la nueva instancia

CodeDeploy no implementa automáticamente la aplicación en una instancia recién etiquetada. Debe crear una nueva implementación en el grupo de implementaciones.

Se puede utilizar CodeDeploy para habilitar las implementaciones automáticas en nuevas instancias de EC2 en los grupos de Auto Scaling de Amazon EC2. Para obtener más información, consulte [Integración CodeDeploy con Amazon EC2 Auto Scaling](#).

Solución de problemas de instancias

Temas

- [Las etiquetas deben estar configuradas correctamente](#)
- [AWS CodeDeploy el agente debe estar instalado y ejecutándose en las instancias](#)
- [Las implementaciones no producen error hasta una hora después cuando se termina una instancia durante una implementación](#)
- [Análisis de archivos de registro para investigar los errores de implementación en instancias](#)
- [Cree un nuevo archivo de CodeDeploy registro si se ha eliminado accidentalmente](#)
- [Solución de errores de despliegue «InvalidSignatureException — Firma caducada: \[hora\] es anterior a \[hora\]»](#)

Las etiquetas deben estar configuradas correctamente

Usa el [list-deployment-instances](#) comando para confirmar que las instancias utilizadas para una implementación estén etiquetadas correctamente. Si falta una instancia de EC2 en la salida, utilice la consola de EC2 para confirmar que las etiquetas se han configurado en la instancia. Para obtener más información, consulte [Trabajar con etiquetas en la consola](#) en la Guía del usuario de Amazon EC2.

Note

Si etiqueta una instancia y la utiliza inmediatamente CodeDeploy para implementar una aplicación en ella, es posible que la instancia no se incluya en la implementación. Esto se debe a que pueden pasar varios minutos antes de que CodeDeploy pueda leer las etiquetas. Le recomendamos que espere al menos cinco minutos entre el momento en que etiqueta una instancia y el momento en que intenta realizar una implementación en ella.

AWS CodeDeploy el agente debe estar instalado y ejecutándose en las instancias

Para comprobar que el CodeDeploy agente está instalado y ejecutándose en una instancia, consulte [Compruebe que el CodeDeploy agente se esté ejecutando](#).

Para instalar, desinstalar o volver a instalar el CodeDeploy agente, consulte [Instala el CodeDeploy agente](#).

Las implementaciones no producen error hasta una hora después cuando se termina una instancia durante una implementación

CodeDeploy proporciona un período de una hora para que cada evento del ciclo de vida de la implementación se ejecute hasta su finalización. Esto proporciona tiempo suficiente para los scripts de larga duración.

Si los scripts no se ejecutan hasta completarse mientras se está produciendo un evento del ciclo de vida (por ejemplo, si se termina una instancia o se cierra el CodeDeploy agente), es posible que el estado de la implementación tarde hasta una hora en aparecer como Fallido. Esto es cierto incluso si el periodo de tiempo especificado en el script es inferior a una hora. Esto se debe a que, cuando se termina la instancia, el CodeDeploy agente se cierra y no puede procesar más scripts.

Si se termina una instancia entre eventos de ciclo de vida o antes de que se inicie el primer paso de un evento de ciclo de vida, el tiempo de espera se agota en solo cinco minutos.

Análisis de archivos de registro para investigar los errores de implementación en instancias

Si el estado de una instancia en la implementación es distinto de Succeeded, puede examinar los datos del archivo log para poder identificar el problema. Para obtener información sobre cómo obtener acceso a los datos de registro de una implementación, consulte [Vea los datos de registro de las implementaciones locales CodeDeploy de EC2/](#).

Cree un nuevo archivo de CodeDeploy registro si se ha eliminado accidentalmente

Si eliminas accidentalmente el archivo de registro de despliegue en una instancia, CodeDeploy no se crea un archivo de registro de reemplazo. Para crear un nuevo archivo de registro, inicie sesión en la instancia y, a continuación, ejecute estos comandos:

Para una instancia de Amazon Linux, Ubuntu Server o RHEL, ejecute estos comandos y en este orden, uno cada vez:

```
systemctl stop codedeploy-agent
```

```
systemctl start codedeploy-agent
```

Para una instancia de Windows Server:

```
powershell.exe -Command Restart-Service -Name codedeployagent
```

Solución de errores de despliegue «InvalidSignatureException — Firma caducada: [hora] es anterior a [hora]»

CodeDeploy requiere referencias horarias precisas para realizar sus operaciones. Si la fecha y la hora de la instancia no están configuradas correctamente, es posible que no coincidan con la fecha de firma de la solicitud de despliegue, que se CodeDeploy rechaza.

Para evitar errores relacionados con la configuración de tiempo incorrecta, consulte los siguientes temas:

- [Establecer la hora en una instancia de Linux](#)
- [Establecimiento de la hora para una instancia de Windows](#)

Solucione problemas con los GitHub tokens

El token de GitHub OAuth no es válido

CodeDeploy las aplicaciones creadas después de junio de 2017 utilizan tokens de GitHub OAuth para cada región. AWS El uso de fichas vinculadas a AWS regiones específicas te permite controlar mejor qué CodeDeploy aplicaciones tienen acceso a un GitHub repositorio.

Si recibes un error de GitHub token, es posible que tengas un token antiguo que ahora no es válido.

Para corregir un token de GitHub OAuth no válido

1. Para eliminar el token anterior, utilice uno de los siguientes métodos:
 - Para eliminar el token anterior mediante la API, usa. [DeleteGitHubAccountToken](#)
 - Para eliminar el token antiguo mediante la AWS Command Line Interface:
 - a. Vaya al ordenador en el que reside el token.
 - b. Asegúrese de que AWS CLI está instalado en este equipo. Para obtener instrucciones, consulte [Instalación, actualización y desinstalación de la AWS CLI](#) en la Guía del usuario de la AWS Command Line Interface .
 - c. Introduzca el siguiente comando en el equipo en el que reside el token:

```
aws delete-git-hub-account-token
```

Para obtener más información sobre la sintaxis del comando, consulte [delete-git-hub-account-token](#).

2. Añada un nuevo token de OAuth. Para obtener más información, consulte [Integración CodeDeploy con GitHub](#).

Se ha superado el número máximo de tokens de GitHub OAuth

Al crear una CodeDeploy implementación, la cantidad máxima de GitHub tokens permitidos es 10. Si recibes un error sobre los tokens de GitHub OAuth, asegúrate de tener 10 o menos tokens. Si tiene más de 10 tokens, los primeros tokens que se han creado no son válidos. Por ejemplo, si tiene 11

tokens, el primer token que ha creado no es válido. Si tiene 12 tokens, los dos primeros tokens que ha creado no son válidos. Para obtener información sobre el uso de la CodeDeploy API para eliminar los tokens antiguos, consulta. [DeleteGitHubAccountToken](#)

Solución de problemas de Amazon EC2 Auto Scaling

Temas

- [Solución de problemas generales de Amazon EC2 Auto Scaling](#)
- [«CodeDeployRole no le da permiso para realizar operaciones en el siguiente AWS servicio: AmazonAutoScaling» error](#)
- [Las instancias de un grupo de Amazon EC2 Auto Scaling se aprovisionan y se terminan constantemente antes de que pueda implementarse una revisión](#)
- [La terminación o el reinicio de una instancia de Amazon EC2 Auto Scaling puede producir un error en las implementaciones](#)
- [Evite asociar varios grupos de implementación a un único grupo de Amazon EC2 Auto Scaling](#)
- [Las instancias de EC2 de un grupo de Amazon EC2 Auto Scaling no se pueden lanzar y aparece el error "Heartbeat Timeout"](#)
- [Los enlaces de ciclo de vida de Amazon EC2 Auto Scaling que no coincidan pueden provocar que las implementaciones automáticas en los grupos de Amazon EC2 Auto Scaling se detengan o fallen.](#)
- [Error “La implementación falló porque no se encontró ninguna instancia para su grupo de implementación”](#)

Solución de problemas generales de Amazon EC2 Auto Scaling

Las implementaciones en instancias de EC2 en un grupo de Amazon EC2 Auto Scaling pueden fallar por las razones siguientes:

- Amazon EC2 Auto Scaling lanza y termina de forma continua las instancias de EC2. Si CodeDeploy no puede implementar automáticamente la revisión de la aplicación, Amazon EC2 Auto Scaling lanza y termina las instancias de EC2 de forma continua.

Desasocie el grupo Auto Scaling de Amazon EC2 del grupo de CodeDeploy implementación o cambie la configuración de su grupo de Auto Scaling de Amazon EC2 para que el número deseado de instancias coincida con el número actual de instancias (evitando así que Amazon EC2

Auto Scaling lance más instancias de EC2). Para obtener más información, consulte [Cambie la configuración del grupo de implementación con CodeDeploy](#) o [Escalado manual para Amazon EC2 Auto Scaling](#).

- El agente no responde. CodeDeploy Es posible que el CodeDeploy agente no esté instalado si los scripts de inicialización (por ejemplo, los scripts de inicio en la nube) que se ejecutan inmediatamente después de lanzar o iniciar una instancia de EC2 tardan más de una hora en ejecutarse. CodeDeploy dispone de un tiempo de espera de una hora para que el CodeDeploy agente responda a las implementaciones pendientes. Para solucionar este problema, mueva los scripts de inicialización a la revisión de la aplicación. CodeDeploy
- Una instancia de EC2 de un grupo de Amazon EC2 Auto Scaling se reinicia durante una implementación. La implementación puede fallar si se reinicia una instancia de EC2 durante una implementación o si el CodeDeploy agente se apaga mientras se procesa un comando de implementación. Para obtener más información, consulte [La terminación o el reinicio de una instancia de Amazon EC2 Auto Scaling puede producir un error en las implementaciones](#).
- Varias revisiones de aplicación se implementan simultáneamente en la misma instancia de EC2 en un grupo de Amazon EC2 Auto Scaling. La implementación de varias revisiones de aplicación en la misma instancia de EC2 en un grupo de Amazon EC2 Auto Scaling al mismo tiempo puede producir un error si una de las implementaciones tiene scripts que se ejecutan durante más de unos minutos. No implemente varias revisiones de aplicación en las mismas instancias de EC2 en un grupo de Amazon EC2 Auto Scaling.
- Una implementación produce un error en nuevas instancias de EC2 que se lanzan como parte de un grupo de Amazon EC2 Auto Scaling. En este caso, la ejecución de scripts en una implementación puede evitar el lanzamiento de instancias de EC2 en el grupo de Amazon EC2 Auto Scaling. (Puede parecer que otras instancias de EC2 del grupo de Amazon EC2 Auto Scaling se ejecutan normalmente). Para solucionar este problema, asegúrese de que todos los demás scripts se completan primero:
 - CodeDeploy el agente no está incluido en la AMI: si utiliza el `cfn-init` comando para instalar el CodeDeploy agente al lanzar una nueva instancia, coloque el script de instalación del agente al final de la `cfn-init` sección de la AWS CloudFormation plantilla.
 - CodeDeploy el agente está incluido en la AMI: configure la AMI para que el agente esté en un `Stopped` estado cuando se cree la instancia y, a continuación, incluya un script para iniciar el agente como paso final en la biblioteca de `cfn-init` scripts.

«CodeDeployRole no le da permiso para realizar operaciones en el siguiente AWS servicio: AmazonAutoScaling» error

Las implementaciones que utilizan un grupo de escalado automático creado con una plantilla de lanzamiento requieren los siguientes permisos. Estos permisos se suman a los concedidos por la política `AWSCodeDeployRole` AWS gestionada.

- `EC2:RunInstances`
- `EC2:CreateTags`
- `iam:PassRole`

Si no tiene los permisos indicados, puede recibir este error. Para obtener más información, consulte [Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling](#), [Creación de una plantilla de lanzamiento para un grupo de escalado automático](#) y [Permisos](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Las instancias de un grupo de Amazon EC2 Auto Scaling se aprovisionan y se terminan constantemente antes de que pueda implementarse una revisión

En algunos casos, un error puede impedir que se realice la implementación en instancias recién aprovisionadas de un grupo de Amazon EC2 Auto Scaling. Los resultados son instancias en mal estado e implementaciones infructuosas. Como la implementación no se puede ejecutar o no se completa correctamente, las instancias se terminan en cuanto se crean. La configuración del grupo de Amazon EC2 Auto Scaling provoca que se aprovisionen otro lote de instancias para intentar satisfacer los requisitos de hosts mínimos en buen estado. Este lote se termina también y así una y otra vez.

Entre las causas posibles se incluyen las siguientes:

- Comprobaciones de estado de Amazon EC2 Auto Scaling no superadas.
- Un error en la revisión de la aplicación.

Para solucionar este problema, siga estos pasos:

1. Cree manualmente una instancia de EC2 que no forme parte del grupo de Amazon EC2 Auto Scaling. Etiquete la instancia con una etiqueta de instancia EC2 única.
2. Añada la nueva instancia al grupo de implementaciones afectado.
3. Implemente una nueva revisión de la aplicación sin errores en el grupo de implementaciones.

Esto hace que el grupo de Amazon EC2 Auto Scaling implemente la revisión de la aplicación en instancias futuras en el grupo de Amazon EC2 Auto Scaling.

Note

Tras confirmar que las implementaciones se han realizado correctamente, elimina la instancia que creaste para evitar que se sigan cobrando a tu AWS cuenta.

La terminación o el reinicio de una instancia de Amazon EC2 Auto Scaling puede producir un error en las implementaciones

Si se lanza una instancia de EC2 a través de Amazon EC2 Auto Scaling y la instancia termina o se reinicia, las implementaciones en dicha instancia pueden producir un error por las razones siguientes:

- Durante una implementación en curso, un evento de escalado o cualquier otro evento de terminación provoca que la instancia se desvincule del grupo de Amazon EC2 Auto Scaling y que termine. Como la implementación no se puede completar, produce un error.
- La instancia se reinicia, pero tarda más de cinco minutos en iniciarse. CodeDeploy trata esto como un tiempo de espera. El servicio no puede realizar implementaciones actuales y futuras en la instancia.

Para solucionar este problema:

- En general, asegúrese de que todas las implementaciones se completen antes de que la instancia termine o se reinicie. Asegúrese de que todas las implementaciones comienzan una vez iniciada o reiniciada la instancia.
- Las implementaciones pueden producir un error si especifica una imagen de máquina de Amazon (AMI) base de Windows Server para una configuración de Amazon EC2 Auto Scaling y usa el servicio EC2Config para establecer el nombre de equipo de la instancia. Para solucionar este

problema, en la AMI base de Windows Server, en la pestaña General de Propiedades del servicio EC2, desactive la opción Definir nombre del equipo. Una vez desactivada esta casilla, este comportamiento se deshabilitará para todas las nuevas instancias de Amazon EC2 Auto Scaling de Windows Server lanzadas con esa AMI base de Windows Server. Para las instancias de Amazon EC2 Auto Scaling de Windows Server en las que este comportamiento está habilitado, no es necesario desactivar esta casilla. Simplemente vuelva a aplicar las implementaciones que han producido un error en esas instancias una vez reiniciadas.

Evite asociar varios grupos de implementación a un único grupo de Amazon EC2 Auto Scaling

Como práctica recomendada, debe asociar solo un grupo de implementación a cada grupo de Amazon EC2 Auto Scaling.

Esto se debe a que si Amazon EC2 Auto Scaling escala una instancia que tiene enlaces asociados con varios grupos de implementación, envía las notificaciones para todos los enlaces a la vez. Esto provoca que comiencen a aplicarse las distintas implementaciones en cada instancia al mismo tiempo. Cuando varias implementaciones envían comandos al CodeDeploy agente al mismo tiempo, es posible que se agote el tiempo de espera de cinco minutos entre un evento del ciclo de vida y el inicio del despliegue o el final del evento del ciclo de vida anterior. Si esto ocurre, la implementación no se realiza correctamente, incluso aunque el proceso de implementación se haya ejecutado según lo previsto.

Note

El tiempo de espera predeterminado de un script en un evento de ciclo de vida es de 30 minutos. Puede cambiar el tiempo de espera a un valor diferente en el archivo.

AppSpec Para obtener más información, consulte [Agregue un AppSpec archivo para una implementación local de EC2/](#).

No es posible controlar el orden en el que se producen las implementaciones si se realizan varios intentos de implementación al mismo tiempo.

Por último, si se produce un error de la implementación en cualquier instancia, Amazon EC2 Auto Scaling termina inmediatamente la instancia. Cuando se cierra la primera instancia, el resto de las implementaciones en ejecución empiezan a producir un error. Como el CodeDeploy agente

CodeDeploy tiene un tiempo de espera de una hora para responder a las implementaciones pendientes, cada instancia puede tardar hasta 60 minutos en agotarse.

Para obtener más información acerca de Auto Scaling de Amazon EC2, consulte [Under the Hood: CodeDeploy and Auto Scaling integration](#).

Las instancias de EC2 de un grupo de Amazon EC2 Auto Scaling no se pueden lanzar y aparece el error "Heartbeat Timeout"

Es posible que un grupo de Amazon EC2 Auto Scaling no pueda lanzar nuevas instancias de EC2 y genere un mensaje similar al siguiente:

```
Launching a new EC2 instance <instance-Id>. Status Reason: Instance failed to complete user's Lifecycle Action: Lifecycle Action with token<token-Id> was abandoned: Heartbeat Timeout.
```

Este mensaje suele indicar una de las siguientes posibilidades:

- Se alcanzó el número máximo de implementaciones simultáneas asociadas a una AWS cuenta. Para obtener más información acerca de los límites de implementación, consulte [CodeDeploy cuotas](#).
- El grupo de escalado automático intentó lanzar demasiadas instancias de EC2 demasiado rápido. Se limitaron las llamadas a la API a [RecordLifecycleActionHeartbeat](#) [CompleteLifecycleAction](#) para cada nueva instancia.
- Se eliminó una aplicación antes de CodeDeploy que se actualizaran o eliminaran sus grupos de implementación asociados.

Al eliminar una aplicación o un grupo de despliegues, CodeDeploy intenta limpiar todos los ganchos de Auto Scaling de Amazon EC2 asociados a él, pero es posible que queden algunos ganchos. Si ejecuta un comando para eliminar un grupo de implementaciones, los enlaces restantes se devuelven en la salida. Sin embargo, si ejecuta un comando para eliminar una aplicación, los enlaces restantes no aparecen en la salida.

Por lo tanto, es recomendable que elimine todos los grupos de implementaciones asociados a una aplicación antes de eliminar la aplicación. Puede utilizar el resultado del comando para identificar los enlaces de ciclo de vida que deben eliminarse manualmente.

Si aparece un mensaje de error "Heartbeat Timeout", puede determinar si los enlaces de ciclo de vida restantes son la causa del problema y resolverlo mediante el siguiente procedimiento:

1. Realice una de las siguientes acciones siguientes:

- Ejecute el [delete-deployment-group](#) comando para eliminar el grupo de despliegue asociado al grupo de Auto Scaling que está provocando el tiempo de espera de latidos.
- Llame al [update-deployment-group](#) comando con una lista vacía no nula de nombres de grupos de Auto Scaling para separar todos los ganchos del ciclo de vida de CodeDeploy Auto Scaling administrados.

Por ejemplo, introduzca el siguiente comando: AWS CLI

```
aws deploy update-deployment-group --application-name my-example-app
--current-deployment-group-name my-deployment-group --auto-scaling-
groups
```

Como otro ejemplo, si utiliza la CodeDeploy API con Java, llame `UpdateDeploymentGroup` y `autoScalingGroups` configúrelo en `new ArrayList<String>()`. Esto se establece `autoScalingGroups` en una lista vacía y elimina la lista existente. No utilice `null`, que es la opción predeterminada, porque deja `autoScalingGroups` como está, lo cual no es lo que desea.

Examine el resultado de la llamada. Si el resultado contiene una estructura `hooksNotCleanedUp` con una lista de enlaces de ciclo de vida de Amazon EC2 Auto Scaling, la causa más probable del error serán los enlaces de ciclo de vida restantes.

2. Llame al [describe-lifecycle-hooks](#) comando y especifique el nombre del grupo Auto Scaling de Amazon EC2 asociado a las instancias de EC2 que no se pudieron lanzar. Busque alguna de las siguientes opciones en los resultados:


- Nombres de enlace de ciclo de vida de Amazon EC2 Auto Scaling que corresponden a la estructura de `hooksNotCleanedUp` que identificó en el paso 1.
- Nombres de enlace de ciclo de vida de Amazon EC2 Auto Scaling que contienen el nombre del grupo de implementación asociado al grupo de escalado automático que está fallando.
- Nombres de enlaces del ciclo de vida de Amazon EC2 Auto Scaling que pueden haber provocado el tiempo de espera de la implementación. CodeDeploy

3. Si un enlace pertenece a una de las categorías enumeradas en el paso 2, [delete-lifecycle-hook](#) ejecute el comando para eliminarlo. En la llamada, especifique el grupo de Amazon EC2 Auto Scaling y el enlace de ciclo de vida.

 Important

Elimine únicamente los enlaces que estén causando problemas, tal y como se describe en el paso 2. Si elimina los enlaces viables, es posible que las implementaciones fallen o CodeDeploy que no puedan implementar las revisiones de la aplicación en instancias de EC2 ampliadas.

4. Llame al [create-deployment-group](#) comando [update-deployment-group](#) con los nombres de grupo de Auto Scaling que desee. CodeDeploy vuelve a instalar los ganchos de Auto Scaling con nuevos UUID.

 Note

Si separa un grupo de Auto Scaling de un grupo de despliegues, cualquier CodeDeploy despliegue en curso en el grupo de Auto Scaling podría fallar y las nuevas instancias de EC2 que estén escaladas de forma horizontal por el grupo de Auto Scaling no recibirán las revisiones de su aplicación. CodeDeploy Para que Auto Scaling vuelva a funcionar CodeDeploy, tendrá que volver a conectar el grupo de Auto Scaling al grupo de despliegue y llamar a uno nuevo `CreateDeployment` para iniciar un despliegue en toda la flota.

Los enlaces de ciclo de vida de Amazon EC2 Auto Scaling que no coincidan pueden provocar que las implementaciones automáticas en los grupos de Amazon EC2 Auto Scaling se detengan o fallen.

Amazon EC2 Auto Scaling y CodeDeploy utilizan enlaces de ciclo de vida para determinar qué revisiones de aplicaciones deben implementarse en qué instancias de EC2 después de lanzarlas en los grupos de Auto Scaling de Amazon EC2. Las implementaciones automáticas pueden detenerse o fallar si los enlaces del ciclo de vida y la información sobre estos enlaces no coincide exactamente en Amazon EC2 Auto Scaling y. CodeDeploy

Si las implementaciones en un grupo de Auto Scaling de Amazon EC2 están fallando, compruebe si los nombres de los enlaces del ciclo de vida en Amazon EC2 Auto Scaling coinciden. CodeDeploy Si no es así, utilice estas llamadas de comandos AWS CLI .

En primer lugar, obtenga la lista de enlaces de ciclo de vida para el grupo de Amazon EC2 Auto Scaling y el grupo de implementación:

1. Llame al [describe-lifecycle-hooks](#) comando y especifique el nombre del grupo Auto Scaling de Amazon EC2 asociado al grupo de implementación en. CodeDeploy En el resultado, en la lista LifecycleHooks, anote todos los valores de LifecycleHookName.
2. Llame al [get-deployment-group](#) comando y especifique el nombre del grupo de implementación asociado al grupo Auto Scaling de Amazon EC2. En el resultado, en la lista autoScalingGroups, busque todos los elementos cuyo valor de nombre coincida con el nombre del grupo de Amazon EC2 Auto Scaling y, a continuación, anote el valor de hook correspondiente.

Después, compare los dos conjuntos de nombres de enlaces de ciclo de vida. Si coinciden exactamente, carácter por carácter, entonces este no es el problema. Pruebe con otros pasos de solución de problemas de Amazon EC2 Auto Scaling que se describen en esta sección.

Sin embargo, si los dos conjuntos de nombres de enlaces de ciclo de vida no coinciden exactamente, carácter por carácter, pruebe lo siguiente:

1. Si hay nombres de enlaces de ciclo de vida en el resultado del comando describe-lifecycle-hooks que no están en el resultado del comando get-deployment-group, proceda del modo siguiente:
 - a. Para cada nombre de enlace del ciclo de vida que aparezca en el resultado del describe-lifecycle-hooks comando, llame al [delete-lifecycle-hook](#) comando.
 - b. Llame al [update-deployment-group](#) comando y especifique el nombre del grupo Amazon EC2 Auto Scaling original. CodeDeploy crea nuevos enlaces de ciclo de vida de reemplazo en el grupo Auto Scaling de Amazon EC2 y asocia los enlaces del ciclo de vida al grupo de implementación. Las implementaciones automáticas se deben reanudar ahora cuando se añadan nuevas instancias al grupo de Amazon EC2 Auto Scaling
2. Si hay nombres de enlaces de ciclo de vida en el resultado del comando get-deployment-group que no están en el resultado del comando describe-lifecycle-hooks, proceda del modo siguiente:

- a. Llame al [update-deployment-group](#) comando, pero no especifique el nombre del grupo Amazon EC2 Auto Scaling original.
- b. Vuelva a `update-deployment-group` ejecutar el comando, pero esta vez especifique el nombre del grupo Amazon EC2 Auto Scaling original. CodeDeploy vuelve a crear los enlaces del ciclo de vida que faltan en el grupo Auto Scaling de Amazon EC2. Las implementaciones automáticas se deben reanudar ahora cuando se añadan nuevas instancias al grupo de Amazon EC2 Auto Scaling

Una vez que los dos conjuntos de nombres de enlaces de ciclo de vida coincidan exactamente, carácter por carácter, las revisiones de la aplicación deberían implementarse de nuevo, pero solamente en las nuevas instancias cuando estas se añadan al grupo de Amazon EC2 Auto Scaling. Las implementaciones no se realizan automáticamente en instancias que ya están en el grupo de Amazon EC2 Auto Scaling.

Error “La implementación falló porque no se encontró ninguna instancia para su grupo de implementación”

Lea esta sección si ve el siguiente error: CodeDeploy

```
The deployment failed because no instances were found for your deployment group. Check your deployment group settings to make sure the tags for your EC2 instances or Auto Scaling groups correctly identify the instances you want to deploy to, and then try again.
```

Las causas posibles de este error son:

1. La configuración de su grupo de implementación incluye etiquetas para instancias de EC2, instancias en las instalaciones o grupos de escalado automático que no son correctas. Para solucionar este problema, compruebe que las etiquetas son correctas y, a continuación, vuelva a implementar la aplicación.
2. Su flota se escaló horizontalmente una vez que se inició la implementación. En este escenario, verá instancias en buen estado en el estado `InService` en su flota, pero también verá el error anterior. Para solucionar este problema, vuelva a implementar la aplicación.
3. Su grupo de escalado automático no incluye ninguna instancia que se encuentre en el estado `InService`. En este escenario, al intentar realizar una implementación en toda la flota, la implementación falla y aparece el mensaje de error anterior porque es CodeDeploy necesario

que al menos una instancia esté en ese estado. InService Hay muchos motivos por los que es posible que no tenga instancias en el estado InService. Algunas de ellas son las siguientes:

- Programó (o configuró manualmente) el tamaño del grupo de escalado automático como 0.
- El escalado automático detectó instancias de EC2 defectuosas (por ejemplo, las instancias de EC2 tenían fallos de hardware), por lo que las canceló todas y no dejó ninguna en el estado InService.
- Durante un evento de escalado de 0 a 1, CodeDeploy implementó una revisión que anteriormente se había realizado correctamente (denominada última revisión correcta) y que había dejado de estar en mal estado desde la última implementación. Esto provocó un error en la implementación en la instancia escalada horizontalmente, lo que, a su vez, provocó que el escalado automático cancelara la instancia y no dejara ninguna instancia en el estado InService.

Si descubre que no tiene ninguna instancia en el estado InService, solucione el problema tal y como se describe en el siguiente procedimiento, [To troubleshoot the error if there are no instances in the InService state](#).

Para solucionar el error si no hay ninguna instancia en el estado InService

1. En la consola de Amazon EC2, compruebe la configuración de Capacidad deseada. Si es cero, configúrelo en un número positivo. Espere a que la instancia esté en InService, lo que significa que la implementación se realizó correctamente. Ha solucionado el problema y puede omitir los pasos restantes de este procedimiento de solución de problemas. Para obtener información sobre cómo establecer la configuración de Capacidad deseada, consulte [Configuración de los límites de capacidad en su grupo de escalado automático](#) en la Guía del usuario de Amazon EC2 Auto Scaling.
2. Si el escalado automático sigue intentando lanzar nuevas instancias de EC2 para cumplir con la capacidad deseada, pero no puede cumplir con el escalado horizontal, normalmente se debe a un fallo en el enlace de ciclo de vida del escalado automático. Solucione este problema de la siguiente manera:
 - a. Para comprobar qué evento de enlace de ciclo de vida del escalado automático está fallando, consulte [Verificación de una actividad de escalado para un grupo de escalado automático](#) en la Guía del usuario de Amazon EC2 Auto Scaling.
 - b. Si el nombre del enlace que falla es `CodeDeploy-managed-automatic-launch-deployment-hook-DEPLOYMENT_GROUP_NAME`, vaya a CodeDeploy, busque el grupo de

despliegue y busque el despliegue fallido que se inició con Auto Scaling. A continuación, investigue por qué falló la implementación.

- c. Si entiende por qué falló la implementación (por ejemplo, si se producían CloudWatch alarmas) y puede solucionar el problema sin cambiar la revisión, hágalo ahora.
- d. Si, tras una investigación, determina que CodeDeploy la última revisión correcta ya no está en buen estado y que no hay ninguna instancia en buen estado en su grupo de Auto Scaling, se encuentra en un escenario de bloqueo de despliegue. Para resolver este problema, debe corregir la CodeDeploy revisión defectuosa eliminando temporalmente el enlace CodeDeploy de ciclo de vida del grupo Auto Scaling y, a continuación, reinstalando el enlace y volviendo a implementar una revisión nueva (buena). Para obtener instrucciones, consulte:
 - [To fix the deployment deadlock issue \(CLI\)](#)
 - [To fix the deployment deadlock issue \(console\)](#)

Para solucionar el problema de bloqueo de la implementación (CLI)

1. (Opcional) Bloquee las canalizaciones de CI/CD que están causando el CodeDeploy error para que no se produzcan despliegues inesperados mientras soluciona el problema.
2. Toma nota de tu DesiredCapacity configuración actual de Auto Scaling:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name  
ASG_NAME
```

Puede que tenga que volver a escalar a este número al final de este procedimiento.

3. Establezca la DesiredCapacity configuración de Auto Scaling en 1. Esto es opcional si la capacidad deseada era mayor que 1 al comienzo. Al reducirla a 1, la instancia tardará menos tiempo en aprovisionarse e implementarse más adelante, lo que agiliza la resolución de problemas. Si la capacidad deseada del escalado automático estaba configurada originalmente en 0, debe aumentarla a 1. Esto es obligatorio.


```
como escalado automático: set-desired-capacity auto-scaling-group-name ASG_NAME --  
desired-capacity 1
```


 Note

En el resto de los pasos de este procedimiento se supone que ha configurado su `DesiredCapacity1`.

En este punto, el escalado automático intenta escalar a una instancia. Luego, debido a que el gancho CodeDeploy agregado aún está presente, CodeDeploy intenta desplegarla; la implementación falla; Auto Scaling cancela la instancia; y Auto Scaling intenta volver a lanzar una instancia para alcanzar la capacidad deseada de una, lo que nuevamente falla. Se encuentra en un ciclo de cancelación-relanzamiento.

4. Anule el registro del grupo de escalado automático del grupo de implementación:

 Warning

El siguiente comando lanzará una nueva instancia de EC2 sin software. Antes de ejecutar el comando, asegúrese de que una instancia InService de escalado automático que no ejecute ningún software sea aceptable. Por ejemplo, asegúrese de que el equilibrador de carga asociado a la instancia no envíe tráfico a este host sin software.

 Important

Utilice el CodeDeploy comando que se muestra a continuación para eliminar el gancho. No retire el gancho a través del servicio Auto Scaling, ya que la extracción no será reconocida por CodeDeploy.

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups
```

Después de ejecutar este comando, ocurre lo siguiente:

- a. CodeDeploy anula el registro del grupo Auto Scaling del grupo de despliegue.

- b. CodeDeploy elimina el enlace del ciclo de vida de Auto Scaling del grupo Auto Scaling.
 - c. Como el enlace que estaba provocando una implementación con error ya no está presente, el escalado automático cancela la instancia de EC2 existente e inmediatamente lanza una nueva para escalarla a la capacidad deseada. La nueva instancia debería pasar pronto al estado InService. La nueva instancia no incluye software.
5. Espere a que el estado de la instancia sea InService. Para verificar el estado, utilice el comando siguiente:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
ASG_NAME --query AutoScalingGroups[0].Instances[*].LifecycleState
```

6. Vuelva a añadir el enlace a la instancia de EC2:

⚠ Important

Utilice el CodeDeploy comando que se muestra a continuación para añadir el gancho. No utilice el servicio Auto Scaling para añadir el gancho, ya que la adición no será reconocida por CodeDeploy.

```
aws deploy update-deployment-group --application-name APPLICATION_NAME
--current-deployment-group-name DEPLOYMENT_GROUP_NAME --auto-scaling-
groups ASG_NAME
```

Después de ejecutar este comando, ocurre lo siguiente:

- a. CodeDeploy vuelve a instalar el enlace del ciclo de vida de Auto Scaling en la instancia EC2
 - b. CodeDeploy vuelve a registrar el grupo de Auto Scaling con el grupo de despliegue.
7. Cree una implementación en toda la flota con Amazon S3 o una GitHub revisión que sepa que está en buen estado y desea usar.

Por ejemplo, si la revisión es un archivo .zip de un bucket de Amazon S3 llamado my-revision-bucket con una clave de objeto de httpd_app.zip, introduzca el siguiente comando:

```
aws deploy create-deployment --application-name APPLICATION_NAME
--deployment-group-name DEPLOYMENT_GROUP_NAME --
```

```
revision "revisionType=S3,s3Location={bucket=my-revision-  
bucket,bundleType=zip,key=httpd_app.zip}"
```

Como ahora hay una instancia InService en el grupo de escalado automático, esta implementación debería funcionar y ya no debería aparecer el error La implementación falló porque no se encontró ninguna instancia para su grupo de implementación.


8. Una vez que la implementación se haya realizado correctamente, escale el grupo de escalado automático de nuevo a la capacidad original, si lo escaló anteriormente:

```
aws autoscaling set-desired-capacity --auto-scaling-group-name ASG_NAME  
--desired-capacity ORIGINAL_CAPACITY
```

Para solucionar el problema de bloqueo de la implementación (consola)

1. (Opcional) Bloquee las canalizaciones de CI/CD que están causando el CodeDeploy error para que no se produzcan despliegues inesperados mientras soluciona el problema.
2. Vaya a la consola de Amazon EC2 y tome nota de la configuración de Capacidad deseada del escalado automático. Puede que tenga que volver a escalar a este número al final de este procedimiento. Para obtener información sobre cómo encontrar esta configuración, consulte [Establecer límites de capacidad en su grupo de escalado automático](#).
3. Establezca el número deseado de instancias de EC2 en 1:


Esto es opcional si la capacidad deseada era mayor que 1 al comienzo. Al reducirla a 1, la instancia tardará menos tiempo en aprovisionarse e implementarse más adelante, lo que agiliza la resolución de problemas. Si la Capacidad deseada del escalado automático estaba configurada originalmente en 0, debe aumentarla a 1. Esto es obligatorio.

 Note

En el resto de los pasos de este procedimiento se supone que ha establecido Capacidad deseada en 1.


- a. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/> y elija Auto Scaling Groups (Grupos de escalado automático) en el panel de navegación.
- b. Elija la región apropiada.
- c. Vaya al grupo de escalado automático problemático.

- d. En Detalles del grupo, seleccione Editar.
 - e. Establezca Capacidad deseada en **1**.
 - f. Elija Actualizar.
4. Anule el registro del grupo de escalado automático del grupo de implementación:

 Warning

Los siguientes pasos secundarios lanzarán una nueva instancia de EC2 sin software. Antes de ejecutar el comando, asegúrese de que una instancia InService de escalado automático que no ejecute ningún software sea aceptable. Por ejemplo, asegúrese de que el equilibrador de carga asociado a la instancia no envíe tráfico a este host sin software.

- a. [Abra la consola en https://console.aws.amazon.com/codedeploy/CodeDeploy](https://console.aws.amazon.com/codedeploy/CodeDeploy) .
- b. Elija la región apropiada.
- c. En el panel de navegación, elija Aplicaciones.
- d. Elija el nombre de la CodeDeploy aplicación.
- e. Elija el nombre de su grupo CodeDeploy de implementación.
- f. Elija Editar.
- g. En Configuración del entorno, deseleccione los grupos de Amazon EC2 Auto Scaling.

 Note

La consola no le permite guardar la configuración si no hay ninguna configuración de entorno definida. Para evitar esta comprobación, añada temporalmente una etiqueta de EC2 o On-premises que sepa que no se resolverá en ningún host. Para añadir una etiqueta, seleccione Instancias de Amazon EC2 o Instancia en las instalaciones y añada una etiqueta Clave de **EC2** o **On-premises**. Puede dejar el valor de la etiqueta vacío.

- h. Elija Guardar cambios.

Después de completar estos pasos secundarios, ocurre lo siguiente:

- i. **CodeDeploy anula el registro del grupo Auto Scaling del grupo de despliegue.**

- ii. CodeDeploy elimina el enlace del ciclo de vida de Auto Scaling del grupo Auto Scaling.
 - iii. Como el enlace que estaba provocando una implementación con error ya no está presente, el escalado automático cancela la instancia de EC2 existente e inmediatamente lanza una nueva para escalarla a la capacidad deseada. La nueva instancia debería pasar pronto al estado InService. La nueva instancia no incluye software.
5. Espere a que el estado de la instancia sea InService. Para verificar su estado:
 - a. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
 - b. En el panel de navegación, elija Auto Scaling Groups (Grupos de).
 - c. Elija el grupo de escalado automático.
 - d. En el panel de contenido, elija la pestaña Administración de instancias.
 - e. En Instancias, asegúrese de que la columna Ciclo de vida se indique InService junto a la instancia.
6. Vuelva a registrar el grupo de Auto Scaling con el grupo de CodeDeploy despliegue mediante el mismo método que utilizó para eliminarlo:
 - a. Abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy/>.
 - b. Elija la región apropiada.
 - c. En el panel de navegación, elija Aplicaciones.
 - d. Elija el nombre de la CodeDeploy aplicación.
 - e. Elija el nombre de su grupo CodeDeploy de implementación.
 - f. Elija Editar.
 - g. En Configuración del entorno, seleccione Grupos de Amazon EC2 Auto Scaling y seleccione su grupo de escalado automático de la lista.
 - h. En Instancias de Amazon EC2 o en Instancias en las instalaciones, busque la etiqueta que ha añadido y elimínela.
 - i. Anule la selección de la casilla de verificación situada junto a Instancias de Amazon EC2 o Instancias en las instalaciones.
 - j. Elija Guardar cambios.

Esta configuración vuelve a instalar el enlace de ciclo de vida en el grupo de escalado automático.

7. Cree una implementación en toda la flota con Amazon S3 o una GitHub revisión que sepa que está en buen estado y desea usar.

Por ejemplo, si la revisión es un archivo .zip de un bucket de Amazon S3 llamado my-revision-bucket con una clave de objeto de httpd_app.zip, realice lo siguiente:

- a. En la CodeDeploy consola, en la página del grupo de despliegues, selecciona Crear despliegue.
- b. En Tipo de revisión, elija Mi aplicación está almacenada en Amazon S3.
- c. En Ubicación de la revisión, elija s3://my-revision-bucket/httpd_app.zip.
- d. En Tipo de archivo de revisión, elija .zip.
- e. Elija Crear implementación.

Como ahora hay una instancia InService en el grupo de escalado automático, esta implementación debería funcionar y ya no debería aparecer el error La implementación falló porque no se encontró ninguna instancia para su grupo de implementación.

8. Una vez que la implementación se haya realizado correctamente, escale el grupo de escalado automático de nuevo a la capacidad original, si lo escaló anteriormente:
 - a. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/> y elija Auto Scaling Groups (Grupos de escalado automático) en el panel de navegación.
 - b. Elija la región apropiada.
 - c. Vaya al grupo de escalado automático.
 - d. En Detalles del grupo, seleccione Editar.
 - e. Vuelva a establecer Capacidad deseada en su valor original.
 - f. Elija Actualizar.

Códigos de error para AWS CodeDeploy

En este tema se proporciona información de referencia sobre CodeDeploy los errores.

Código de error	Descripción
	Se produjo un error en el despliegue debido a un problema con el CodeDeploy agente.

Código de error	Descripción
AGENT_ISSUE	<p>Asegúrese de que el agente está instalado y se ejecuta en todas las instancias de este grupo de implementaciones.</p> <p>Más información:</p> <ul style="list-style-type: none">• Compruebe que el CodeDeploy agente se esté ejecutando• Instala el CodeDeploy agente• Trabajando con el CodeDeploy agente
AUTO_SCALING_IAM_ROLE_PERMISSIONS	<p>El rol de servicio asociado a su grupo de implementación no tiene el permiso necesario para llevar a cabo operaciones en el siguiente servicio de AWS .</p> <p>Más información:</p> <ul style="list-style-type: none">• Paso 2: Crear un rol de servicio para CodeDeploy• Creación de un rol para delegar permisos a un servicio de AWS
HEALTH_CONSTRAINTS	<p>La implementación global falló porque fallaron demasiadas instancias individuales de la implementación, hay muy pocas instancias en buen estado disponibles para la implementación o algunas instancias de su grupo de implementaciones tienen problemas.</p> <p>Más información:</p> <ul style="list-style-type: none">• Instance Health• Solución de problemas de instancias• Resolución de problemas de implementación de EC2/en las instalaciones

Código de error	Descripción
HEALTH_CONSTRAINTS_INVALID	<p>La implementación no puede comenzar porque no hay el número mínimo de instancias en buen estado, tal como define la configuración de implementación. Puede reducir el número de instancias en buen estado necesario actualizando la configuración de implementación o aumentando el número de instancias de este grupo de implementaciones.</p> <p>Más información:</p> <ul style="list-style-type: none">• Instance Health• Trabajar con instancias para CodeDeploy
IAM_ROLE_MISSING	<p>La implementación falló porque no existe ningún rol de servicio con el nombre especificado para el grupo de implementaciones. Asegúrese de que utiliza el nombre de rol de servicio correcto.</p> <p>Más información:</p> <ul style="list-style-type: none">• Paso 2: Crear un rol de servicio para CodeDeploy• Cambie la configuración del grupo de implementación con CodeDeploy

Código de error	Descripción
IAM_ROLE_PERMISSIONS	<p>CodeDeploy no tiene los permisos necesarios para asumir una función, o la función de IAM que está utilizando no le da permiso para realizar operaciones en un AWS servicio.</p> <p>Más información:</p> <ul style="list-style-type: none">• Paso 1: Configurar• Paso 2: Crear un rol de servicio para CodeDeploy• Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2

Código de error	Descripción
NO_INSTANCES	<p data-bbox="829 226 1455 306">Esto podría deberse a una de las siguientes causas.</p> <ul data-bbox="829 352 1507 1549" style="list-style-type: none"><li data-bbox="829 352 1507 814">• Si se trata de una implementación azul/verde de EC2/en las instalaciones, si utiliza etiquetas de Amazon EC2, es posible que estas etiquetas no estén configuradas correctamente. En su grupo de CodeDeploy y implementación, asegúrese de que estén incluidos en las instancias azules y en las instancias verdes. Puede confirmar que las instancias están etiquetadas correctamente a través de la consola de Amazon EC2.<li data-bbox="829 840 1507 1302">• Si utiliza un grupo de Amazon EC2 Auto Scaling, puede que el grupo de escalado automático no tenga capacidad suficiente. Asegúrese de que su grupo de escalado automático tenga capacidad suficiente para su implementación. Puede confirmar que su grupo de Amazon EC2 Auto Scaling tiene capacidad comprobando el número de instancias en buen estado mediante la consola de Amazon EC2.<li data-bbox="829 1327 1507 1549">• Si se trata de una implementación azul/verde de EC2/en las instalaciones, es posible que las flotas azul y verde no tengan el mismo tamaño. Asegúrese de que ambas flotas tengan el mismo tamaño. <p data-bbox="829 1629 1073 1661">Más información:</p> <ul data-bbox="829 1707 1365 1745" style="list-style-type: none"><li data-bbox="829 1707 1365 1745">• Tagging Instances for Deployments

Código de error	Descripción
	<ul style="list-style-type: none">• Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling• Creación de una aplicación para una implementación azul/verde (consola)
OVER_MAX_INSTANCES	<p>La implementación falló porque desea implementar más instancias de las que están autorizadas para su cuenta. Para reducir el número de instancias que desea implementar, actualice la configuración de etiqueta de este grupo de implementaciones o elimine algunas de las instancias de destino. Como alternativa, puedes ponerte en contacto con nosotros AWS Support para solicitar un aumento del límite.</p> <p>Más información:</p> <ul style="list-style-type: none">• Cambie la configuración del grupo de implementación con CodeDeploy• CodeDeploy cuotas• Solicitar un aumento del límite
THROTTLED	<p>Se produjo un error en la implementación porque se realizaron más solicitudes de las que permite un rol de IAM. AWS CodeDeploy Intenta reducir el número de solicitudes.</p> <p>Más información:</p> <ul style="list-style-type: none">• Tasa de solicitudes de la API de consulta

Código de error	Descripción
UNABLE_TO_SEND_ASG	<p>La implementación falló porque el grupo de implementación no está configurado correctamente con su grupo de Amazon EC2 Auto Scaling. En la CodeDeploy consola, elimine el grupo Auto Scaling de Amazon EC2 del grupo de implementación y, a continuación, vuelva a añadirlo.</p> <p>Más información:</p> <ul style="list-style-type: none">• Under the Hood: CodeDeploy e integración con Auto Scaling

Temas relacionados de

[Solución de problemas CodeDeploy](#)

CodeDeploy recursos

Los siguientes recursos relacionados pueden ayudarle a medida que trabaja con ellos CodeDeploy.

Guías de referencia y recursos de soporte

- [AWS CodeDeploy Referencia de la API](#): descripciones, sintaxis y ejemplos de uso sobre CodeDeploy las acciones y los tipos de datos, incluidos los parámetros y códigos de error más comunes.
- [CodeDeploy Preguntas frecuentes técnicas](#): las principales preguntas de los clientes sobre CodeDeploy.
- [AWS centro de soporte](#): el centro para crear y gestionar sus AWS Support casos. También incluye enlaces a otros recursos útiles como foros, preguntas técnicas frecuentes, estado de los servicios y AWS Trusted Advisor.
- [AWS planes de soporte](#): la página web principal para obtener información sobre AWS Support los planes.
- [Comuníquese con nosotros](#): un punto de contacto central para consultas relacionadas con la AWS facturación, la cuenta, los eventos, el uso indebido y otros 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.

Muestras

- [CodeDeploy muestras en GitHub](#): ejemplos y escenarios de plantillas para CodeDeploy.
- CodeDeploy Complemento de [Jenkins: complemento](#) de Jenkins para. CodeDeploy
- [CodeDeploy agente](#): versión de código abierto del agente. CodeDeploy

Blogs

- [AWS DevOps blog](#): información para desarrolladores, administradores de sistemas y arquitectos.

AWS kits y herramientas de desarrollo de software

Los siguientes AWS SDK y herramientas respaldan el desarrollo de soluciones con CodeDeploy:

- [AWS SDK for .NET](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto\)](#)
- [AWS SDK for Ruby](#)
- AWS Toolkit for Eclipse — Partes [1](#), [2](#) y [3](#).
- [AWS Tools for Windows PowerShell](#)— Un conjunto de PowerShell cmdlets de Windows que exponen la funcionalidad del AWS SDK for .NET PowerShell entorno.
- [CodeDeploy cmdlets en AWS Tools for PowerShell](#): conjunto de PowerShell cmdlets de Windows que muestran la funcionalidad del entorno. CodeDeploy PowerShell
- [AWS Command Line Interface](#)— Una sintaxis de línea de comandos uniforme para acceder a los servicios. AWS CLI Utiliza un único proceso de configuración para permitir el acceso a todos los servicios compatibles.
- [AWS herramientas para desarrolladores](#): enlaces a herramientas y recursos para desarrolladores que proporcionan documentación, ejemplos de código, notas de la versión y otra información para ayudarlo a crear aplicaciones innovadoras con CodeDeploy y AWS.

Historial de documentos

En la siguiente tabla se describen los principales cambios realizados en esta guía del usuario para admitir funciones nuevas y mejoradas desde la última versión de la Guía del CodeDeploy usuario.

- Versión de la API: 2014-10-06

Cambio	Descripción	Fecha
Se agregó texto alternativo (texto alternativo)	Todas las imágenes de esta guía se actualizaron para incluir texto alternativo. Los lectores de pantalla leen el texto alternativo y hace que nuestra documentación sea más accesible para los usuarios invidentes.	22 de mayo de 2024
CodeDeploy versión Agent v1.7.0	El AWS CodeDeploy agente se actualizó a la versión 1.7.0. Para obtener más información, consulte el historial de versiones del CodeDeploy agente .	6 de marzo de 2024
Comandos modificados	Los sudo <code>service codedeploy-agent status/start/stop</code> comandos ya no se recomiendan porque no se utilizan <code>systemd</code> para la gestión de los procesos del CodeDeploy y agente, lo cual es una práctica recomendada. Para asegurarse de que se utilice <code>systemd</code> , emplee	12 de enero de 2024

los comandos `systemctl`, como se muestra en el siguiente ejemplo: `systemctl start codedeploy-agent`. Los siguientes temas se actualizaron con `systemctl` comandos: [instalar el CodeDeploy agente para Amazon Linux o RHEL](#), [instalar el CodeDeploy agente para Ubuntu Server](#), [solucionar todos los errores omitidos en los eventos del ciclo de vida](#) y [crear un nuevo archivo de CodeDeploy registro en caso de que se haya eliminado accidentalmente](#).

[Temas añadidos](#)

Se agregaron los temas [Administrar el proceso del CodeDeploy agente y Hacer referencia a los archivos de referencia a los scripts de eventos del ciclo de vida](#).

12 de enero de 2024

[CodeDeploy Ahora es compatible con configuraciones zonales](#)

Se actualizó el CodeDeploy tema [Crear una configuración de despliegue con información de configuración zonal](#).

7 de diciembre de 2023

[CodeDeploy ahora admite despliegues de terminación](#)

Se ha añadido el tema [Habilitar implementaciones de terminación durante eventos de reducción horizontal de Auto Scaling](#) para describir la característica de implementación de terminación. También se actualizó la [sección AppSpec «enlaces» para un despliegue local de EC2/local](#), los temas [Creación de un grupo de despliegues para un despliegue local \(consola\)](#) y [Creación de un grupo de despliegues para un despliegue local \(consola\) de EC2/local \(consola\) para tener en cuenta esta función](#).

7 de diciembre de 2023

[Formato JSON corregido](#)

Se ha corregido el formato de los ejemplos de código JSON en el tema de la [sección «recursosAppSpec» \(solo Amazon ECS y AWS Lambda despliegues\)](#).

3 de diciembre de 2023

[Se ha añadido un tema de resolución de problemas](#)

Se ha añadido un tema de [Resolución de problemas de implementación de Amazon ECS](#).

24 de octubre de 2023

[Se actualizó el nombre del AppSpec archivo](#)

Se actualizó la [referencia del CodeDeploy AppSpec archivo](#) para indicar que el AppSpec archivo debe tener un nombre `appspec.yml` para las implementaciones locales de EC2.

5 de octubre de 2023

[CodeDeploy ahora admite varios balanceadores de carga](#)

Se actualizaron los [temas Crear un grupo de despliegues para un despliegue local \(consola\)](#), [Crear un grupo de despliegues para un despliegue azul/verde de EC2/local \(consola\)](#) y [Configurar un balanceador de carga en Elastic Load Balancing para despliegues de Amazon CodeDeploy EC2](#) para indicar la compatibilidad con varios [balanceadores de carga](#).

26 de septiembre de 2023

[Se han actualizado las regiones en el tema sobre VPC](#)

Se actualizó la tabla del tema [Uso CodeDeploy con Amazon Virtual Private Cloud](#) para mostrar más compatibilidad regional. En concreto, las regiones Asia Pacífico (Hyderabad), Asia Pacífico (Melbourne), Europa (Milán), Europa (España) y Europa (Zúrich) se han actualizado para mostrar la compatibilidad del punto de conexión del agente.

22 de septiembre de 2023

[Se han actualizado las regiones en el tema del kit de recursos](#)

Se agregaron las siguientes AWS regiones a los [nombres de los buckets del kit de recursos por](#) sección: Asia Pacífico (Osaka), Asia Pacífico (Hyderabad), Canadá (Central), Europa (España), Europa (Zúrich) y Oriente Medio (Emiratos Árabes Unidos). También se han actualizado las políticas de IAM con estas regiones y cualquier otra que faltara.

22 de septiembre de 2023

[Se han abreviado los temas sobre la instalación y actualización del agente](#)

Se han abreviado [los temas Instalar el CodeDeploy agente para Windows Server y Actualizar el CodeDeploy agente en Windows Server](#). Se han eliminado las URL redundantes de los buckets de Amazon S3 y los comandos de copia de Amazon S3.

21 de septiembre de 2023

[Se ha añadido la región Asia-Pacífico \(Yakarta\)](#)

Se ha añadido Asia-Pacífico (Yakarta) a [Nombres de los buckets del kit de recursos por región](#).

21 de septiembre de 2023

[CodeDeploy actualizó una política AWS administrada existente](#)

Se actualizó la política AWSCodeDeployRole gestionada. Para obtener más información, consulte [Actualizaciones de AWS a las políticas administradas de AWS](#).

16 de agosto de 2023

Se ha añadido un límite	Se agregó un límite al tema de CodeDeploy límites . El límite es Número máximo de alarmas asociadas con el grupo de implementación.	15 de agosto de 2023
Se ha corregido un paso relacionado con los equilibradores de carga	Se han corregido las instrucciones de la sección Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones (consola) . El paso del equilibrador de carga ahora está marcado como opcional.	3 de agosto de 2023
Se ha aclarado la redacción del tema de Amazon ECS	Se ha aclarado la redacción de Tutorial: Implementación de una aplicación en Amazon ECS . La redacción ahora indica que está implementando una aplicación. Anteriormente, la redacción indicaba que estaba implementando un servicio de Amazon ECS.	3 de agosto de 2023
CodeDeploy ahora está disponible en la región de Israel (Tel Aviv)	CodeDeploy ahora está disponible en la región de Israel (Tel Aviv) (il-central-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	31 de julio de 2023

Actualización de tema	Se ha actualizado el tema Resolución de problemas de implementación de EC2/en las instalaciones con un consejo sobre el uso de un manual de procedimientos para realizar tareas de resolución de problemas de forma automática.	7 de julio de 2023
Actualización de tema	Se actualizó la sección «recursosAppSpec » del tema de las implementaciones de Amazon ECS con más información sobre la definición de tareas (ARN).	7 de julio de 2023
Actualización de tema	Se actualizó el tema Paso 1: Instalar y configurar la instancia local con información AWS CLI sobre la solución de problemas.	7 de julio de 2023
Actualización de tema	Se ha actualizado el tema Prevención del suplente confuso entre servicios con información sobre las implementaciones azul/verde de Amazon ECS a través de AWS CloudFormation.	6 de julio de 2023

Actualización de tema	El tema Prevención del suplente confuso entre servicios se ha actualizado con información sobre las implementaciones azul/verde de Amazon ECS a través de AWS CloudFormation.	6 de julio de 2023
Actualización de tema	Se ha actualizado el tema Configuraciones de implementación predefinidas para una plataforma de informática EC2/en las instalaciones . Se ha añadido una nota sobre el comportamiento de la configuración de implementación de CodeDeployDefault.HalfAtATime predefinida con los grupos de escalado automático.	29 de junio de 2023
Actualización de tema	Se actualizó el AWS CodeDeploy tema Seguridad de la infraestructura para indicar las nuevas versiones mínima y recomendada del protocolo de seguridad de la capa de transporte (TLS).	28 de junio de 2023
Límite actualizado	Se ha modificado el límite siguiente: “Número máximo de horas durante las que una implementación local de EC2/en las instalaciones puede ejecutarse”. Para obtener más información, consulte Límites	27 de junio de 2023

Actualización de tema	El tema Paso 3: Limitar los permisos del CodeDeploy y usuario se actualizó con instrucciones detalladas.	31 de mayo de 2023
CodeDeploy actualizó una política AWS gestionada existente	Se actualizó la política AWSCodeDeployFullAccess gestionada. Para obtener más información, consulte Actualizaciones de AWS a políticas administradas de AWS .	16 de mayo de 2023
CodeDeploy versión v1.6.0 del agente	El AWS CodeDeploy agente se actualizó a la versión 1.6.0. Para obtener más información, consulte Historial de versiones del CodeDeploy agente .	30 de marzo de 2023
CodeDeploy versión v1.5.0 del agente	El AWS CodeDeploy agente se actualizó a la versión 1.5.0. Para obtener más información, consulte Historial de versiones del CodeDeploy agente .	3 de marzo de 2023
Actualización de la plataforma de informática de Amazon ECS	Las implementaciones en una plataforma de informática de Amazon ECS ahora se admiten en la región de Asia-Pacífico (Yakarta).	8 de febrero de 2023

CodeDeploy actualizó una política AWS gestionada existente	Se actualizó la política AWSCodeDeployRole gestionada. Para obtener más información, consulte Actualizaciones de AWS a las políticas administradas de AWS .	3 de febrero de 2023
Actualización de tema	El tema Uso CodeDeploy con Amazon Virtual Private Cloud se actualizó con AWS regiones nuevas y modificadas.	2 de febrero de 2023
Actualizaciones de temas	CodeDeploy ya está disponible en la región Asia Pacífico (Melbourne) (ap-southeast-4). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de estas nuevas regiones.	26 de enero de 2023
Actualizaciones en las prácticas recomendadas de seguridad	La CodeDeploy sección Primeros pasos y algunas otras secciones se actualizaron para adaptarla a las prácticas recomendadas de AWS seguridad.	23 de enero de 2023
CodeDeploy versión v1.4.1 del agente	El AWS CodeDeploy agente se actualizó a la versión 1.4.1. Para obtener más información, consulte el historial de versiones del CodeDeploy agente .	6 de diciembre de 2022

[Se ha añadido un tema de resolución de problemas](#)

Se ha añadido un tema sobre cómo solucionar los errores causados por las largas rutas de archivo utilizadas con el CodeDeploy agente para Windows. Para obtener más información, consulte [Las rutas de archivo largas provocan el error «No existe tal archivo o directorio»](#).

6 de diciembre de 2022

[Se ha cambiado un límite](#)

Se modificó el siguiente límite: «Número máximo de configuraciones de implementación personalizadas asociadas a una AWS cuenta». El límite ahora es 200. Para obtener más información sobre los límites, consulte el tema [Límites](#).

7 de septiembre de 2022

[CodeDeploy versión v1.4.0 del agente](#)

El AWS CodeDeploy agente se actualizó a la versión 1.4.0. Para obtener más información, consulte el [historial de versiones del CodeDeploy agente](#).

31 de agosto de 2022

[Se han corregido algunos límites.](#)

Se corrigieron los siguientes límites: «El número máximo de despliegues simultáneos asociados a una AWS cuenta» es ahora de 1000. «Número máximo de instancias de una única implementación» es ahora 1000. «Número máximo de instancias que pueden utilizar las implementaciones simultáneas en curso y que están asociadas con una cuenta» es ahora 1000. El «número máximo de configuraciones de despliegue personalizadas asociadas a una AWS cuenta» es ahora de 100. Para obtener más información sobre los límites, consulte el tema [Límites](#).

8 de agosto de 2022

[Se agregó una tabla que muestra los CodeDeploy puntos finales compatibles en cada región.](#)

Para obtener más información, consulte [Uso CodeDeploy con Amazon Virtual Private Cloud](#).

20 de abril de 2022

[Se ha añadido un nuevo límite para las implementaciones azules/verdes de Amazon ECS.](#)

El número máximo de horas entre la implementación de una revisión y el desvío del tráfico al entorno de sustitución durante una implementación azul/verde de Amazon ECS es ahora 120. Para obtener más información, consulte [Implementaciones](#) en el tema [Límites](#).

12 de abril de 2022

Se ha añadido un tema sobre cómo prevenir el problema del suplente confuso	Para obtener más información, consulte AWS Identity and Access Management para AWS CodeDeploy .	14 de marzo de 2022
CodeDeploy actualizó una política AWS gestionada existente	Se ha actualizado el rol AmazonEC2RoleforAWSCodeDeployLimited . Para obtener más información, consulte Actualizaciones a las políticas administradas de AWS .	22 de noviembre de 2021
CodeDeploy actualizó una política AWS gestionada existente	Se ha actualizado AWS CodeDeployRole . Para obtener más información, consulte Actualizaciones a las políticas administradas de AWS .	18 de mayo de 2021
CodeDeploy versión v1.3.2 del agente	El AWS CodeDeploy agente se actualizó a la versión 1.3.2. Para obtener más información, consulte el historial de versiones del CodeDeploy agente .	6 de mayo de 2021
CodeDeploy admite la actualización de instancias de Amazon EC2 desactualizadas	CodeDeploy ahora admite la actualización automática de instancias de Amazon EC2 desactualizadas. Para obtener más información, consulte Configuración de opciones avanzadas para un grupo de implementación .	23 de febrero de 2021

CodeDeploy versión v1.3.1 del agente	El AWS CodeDeploy agente se actualizó a la versión 1.3.1. Para obtener más información, consulte el historial de versiones del CodeDeploy agente .	22 de diciembre de 2020
CodeDeploy versión v1.3.0 del agente	El AWS CodeDeploy agente se actualizó a la versión 1.3.0. Para obtener más información, consulte el historial de versiones del CodeDeploy agente .	10 de noviembre de 2020
CodeDeploy versión v1.2.1 del agente	El AWS CodeDeploy agente se actualizó a la versión 1.2.1. Para obtener más información, consulte el historial de versiones del CodeDeploy agente .	23 de septiembre de 2020
CodeDeploy es compatible con los puntos de conexión de Amazon VPC con tecnología AWS PrivateLink	Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus AWS recursos, puede establecer una conexión privada entre su VPC y CodeDeploy. Puede utilizar esta conexión para poder CodeDeploy comunicarse con los recursos de su VPC sin tener que pasar por la Internet pública. Para obtener más información, consulte Uso CodeDeploy con Amazon Virtual Private Cloud .	11 de agosto de 2020

[Límites CodeDeploy de servicio actualizados](#)

Se ha actualizado el límite tanto para el número de aplicaciones por cuenta como para el número de grupos de implementación por aplicación a 1000. Para obtener más información sobre los límites CodeDeploy del servicio, consulta [CodeDeploy los límites](#).

6 de agosto de 2020

[CodeDeploy versión Agente v1.1.2](#)

El AWS CodeDeploy agente se actualizó a la versión 1.1.2. Para obtener más información, consulte el [historial de versiones del CodeDeploy agente](#).

4 de agosto de 2020

[CodeDeploy versión 1.1.0 del agente e integración con Amazon EC2 Systems Manager](#)

La versión 1.1.0 del CodeDeploy agente ya está disponible; para obtener más información, consulte el [historial de versiones del agente](#). CodeDeploy Ahora puede usar Amazon EC2 Systems Manager para administrar automáticamente la instalación y las actualizaciones de CodeDeploy sus agentes en sus instancias de Amazon EC2 o locales. Para obtener más información, consulte [Instalación del CodeDeploy agente mediante Amazon EC2 Systems Manager](#).

30 de junio de 2020

[CodeDeploy admite la gestión de las implementaciones azul/verde de Amazon ECS con AWS CloudFormation](#)

Ahora puede utilizarla AWS CloudFormation para gestionar las implementaciones azul/verde de Amazon ECS mediante CodeDeploy. Para generar la implementación, defina los recursos verdes y azules y especifique la configuración de enrutamiento y estabilización del tráfico que se va a utilizar en AWS CloudFormation. Para obtener más información, consulte [Crear una implementación azul/verde de Amazon ECS](#) mediante AWS CloudFormation.

19 de mayo de 2020

[CodeDeploy admite el cambio ponderado del tráfico para las implementaciones azul/verde de Amazon ECS](#)

CodeDeploy ahora admite el cambio de tráfico ponderado para las implementaciones azul/verde de Amazon ECS. Puede elegir o crear una configuración de implementación para especificar el número de intervalos de cambio de tráfico de la implementación y el porcentaje de tráfico que se va a cambiar en cada intervalo. Se ha actualizado el tema siguiente para reflejar este cambio: [Configuraciones de implementación en una plataforma de informática de Amazon ECS](#).

6 de febrero de 2020

[Se han actualizado los temas de seguridad, autenticación y control de acceso](#)

La información de seguridad , autenticación y control de acceso se CodeDeploy ha organizado en un nuevo capítulo de seguridad. Para obtener más información, consulte [Seguridad](#).

26 de noviembre de 2019

[CodeDeploy admite las reglas de notificación](#)

Ahora puede usar reglas de notificación para notificar a los usuarios los cambios importantes en las implementaciones. Para obtener más información, consulte [Creación de una regla de notificación](#).

5 de noviembre de 2019

[Temas actualizados](#)

CodeDeploy ya está disponible en la región Asia Pacífico (Hong Kong) (ap-east-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región. Debe habilitar explícitamente el acceso a esta región. Para obtener más información, consulte [Gestión de AWS regiones](#).

25 de abril de 2019

Temas actualizados

AWS CodeDeploy ahora admite despliegues azul/verde de una aplicación containerizada en un servicio de Amazon ECS. Una CodeDeploy y aplicación que utiliza la nueva plataforma informática Amazon ECS despliega una aplicación en contenedores en un nuevo conjunto de tareas de reemplazo en el mismo servicio de Amazon ECS. Se han agregado y actualizado varios temas para reflejar este cambio, como la [descripción general de las plataformas AWS CodeDeploy informáticas](#), las [implementaciones en una plataforma informática de Amazon ECS](#), la [estructura de AppSpec archivos para las implementaciones de Amazon ECS](#) y la [creación de una aplicación para la implementación de un servicio de Amazon ECS \(consola\)](#).

27 de noviembre de 2018

Agente actualizado CodeDeploy

El AWS CodeDeploy agente se actualizó a la versión 1.0.1.1597. Para obtener más información, consulte [Historial de versiones del agente](#). CodeDeploy

15 de noviembre de 2018

Consola actualizada	Los procedimientos de esta guía se actualizaron para adaptarlos al nuevo diseño de la CodeDeploy consola.	30 de octubre de 2018
Nueva versión mínima admitida del CodeDeploy agente	La versión mínima admitida del AWS CodeDeploy agente ahora es la 1.7.x. Para obtener más información, consulte el historial de versiones del CodeDeploy agente .	7 de agosto de 2018

Actualizaciones anteriores

En la siguiente tabla, se describen los cambios importantes de cada versión de la Guía del usuario de AWS CodeDeploy anteriores a junio de 2018.

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	CodeDeploy ya está disponible en la región Europa (París) (eu-west-3). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	19 de diciembre de 2017
Temas actualizados	CodeDeploy ya está disponible en la región de China (Ningxia). Para utilizar servicios en la región de China (Pekín) o China (Ningxia), debe disponer de una cuenta y de credenciales para estas regiones. Las cuentas y credenciales de otras AWS regiones no funcionan en las regiones de Beijing y Ningxia, y viceversa. La información sobre algunos recursos para las regiones de China, como los nombres de los paquetes del kit de CodeDeploy recursos y los procedimientos de instalación	11 de diciembre de 2017

Cambio	Descripción	Fecha de modificación
	<p>de los CodeDeploy agentes, no se incluye en esta edición de la Guía del CodeDeploy usuario.</p> <p>Para obtener más información:</p> <ul style="list-style-type: none"> • CodeDeploy en <u>Cómo empezar AWS en la región de China (Pekín)</u> • CodeDeploy Guía del usuario para las regiones de China (versión en inglés versión en chino) 	
Temas actualizados	<p>CodeDeploy ahora admite el despliegue de una función Lambda. La implementación de AWS Lambda desvía el tráfico entrante de una función de Lambda existente a una versión actualizada de la función de Lambda. Puede elegir o crear una configuración de despliegue para especificar el número de intervalos de cambio de tráfico en el despliegue y el porcentaje de tráfico que se desplazará en cada intervalo. AWS Lambda Las implementaciones son compatibles con el modelo de aplicaciones AWS sin servidor (AWS SAM), por lo que puede usar una preferencia de implementación de AWS SAM para administrar la forma en que se desplaza el tráfico durante una implementación. AWS Lambda Se han añadido y actualizado varios temas para reflejar este cambio, como Descripción general de las plataformas informáticas CodeDeploy, Implementaciones en una plataforma de informática de AWS Lambda, Creación de una implementación en la plataforma de informática de AWS Lambda (consola), Creación de una aplicación para la implementación de una función de AWS Lambda (consola) y Agregar un AppSpec archivo para una implementación de AWS Lambda.</p>	28 de noviembre de 2017

Cambio	Descripción	Fecha de modificación
Nuevo tema	CodeDeploy Ahora admite las implementaciones directamente en una máquina o instancia local en la que esté instalado el CodeDeploy agente. Puede probar una implementación de forma local y, si tiene errores, utilizar los registros de errores del CodeDeploy agente para depurarlos. También puede usar una implementación local para probar la integridad de una revisión de la aplicación, el contenido de un AppSpec archivo, etc. Para obtener más información, consulte Utilice el CodeDeploy agente para validar un paquete de despliegue en una máquina local.	16 de noviembre de 2017
Temas actualizados	CodeDeploy La compatibilidad con los balanceadores de carga de Elastic Load Balancing en los grupos de implementación se ha ampliado para incluir balanceadores de carga de red tanto para las implementaciones azul/verde como para las implementaciones in situ. Ahora puede elegir un Equilibrador de carga de aplicación, un Equilibrador de carga clásico o un Equilibrador de carga de red para su grupo de implementación. Los equilibradores de carga son obligatorios para las implementaciones azul/verde y opcionales para las implementaciones locales. Varios temas se han actualizado para reflejar esta compatibilidad adicional, tales como Integrating CodeDeploy with Elastic Load Balancing , Creación de una aplicación para una implementación local (consola) , Requisitos previos para una implementación , Integrating CodeDeploy with Elastic Load Balancing y Creación de una aplicación para una implementación local (consola) .	12 de septiembre de 2017

Cambio	Descripción	Fecha de modificación
Temas actualizados	<p>CodeDeploy La compatibilidad con los balanceadores de carga de Elastic Load Balancing en los grupos de implementación se ha ampliado para incluir los balanceadores de carga de aplicaciones tanto para las implementaciones azul/verde como para las implementaciones in situ. Ahora puede elegir entre un Equilibrador de carga de aplicación y un Equilibrador de carga clásico para su grupo de implementación. Los equilibradores de carga son obligatorios para las implementaciones azul/verde y opcionales para las implementaciones locales. Los temas Integrating CodeDeploy with Elastic Load Balancing , Cree una aplicación con CodeDeploy y Cree un grupo de despliegue con CodeDeploy se han actualizado para reflejar esta compatibilidad adicional.</p>	10 de agosto de 2017

Cambio	Descripción	Fecha de modificación
Temas nuevos y actualizados	<p>CodeDeploy ahora admite el uso de varios grupos de etiquetas para identificar las uniones e intersecciones de las instancias que se van a incluir en un grupo de implementaciones. Si utiliza un único grupo de etiquetas, cualquier instancia que se identifique mediante al menos una etiqueta del grupo se incluye en el grupo de implementaciones. Si utiliza varios grupos de etiquetas, solo se incluyen las instancias que se identifican por al menos una etiqueta en cada uno de los grupos de etiquetas. Para obtener más información sobre el nuevo método de añadir instancias a un grupo de implementaciones, consulte Tagging Instances for Deployments. Entre otros temas que se han actualizado para reflejar esta compatibilidad se incluyen Creación de una aplicación para una implementación local (consola), Creación de una aplicación para una implementación azul/verde (consola), Creación de un grupo de implementación para una implementación local (consola), Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones (consola), Deployments y Paso 5: Crear una aplicación y un grupo de implementación en Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub.</p>	31 de julio de 2017
Tema actualizado	<p>Se han agregado dos métodos adicionales para instalar el CodeDeploy agente en instancias de Windows Server. Instale el CodeDeploy agente para Windows Server</p> <p>Además de PowerShell los comandos de Windows, ahora están disponibles las instrucciones para descargar el archivo de instalación mediante un enlace HTTPS directo y mediante un comando de copia de Amazon S3. Una vez que el archivo se haya descargado o copiado en una instancia, puede ejecutar la instalación manualmente.</p>	12 de julio de 2017

Cambio	Descripción	Fecha de modificación
Temas actualizados	<p>CodeDeploy ha mejorado la forma en que administra las conexiones a GitHub las cuentas y los repositorios. Ahora puede crear y almacenar hasta 25 conexiones a GitHub cuentas para asociar CodeDeploy aplicaciones a GitHub repositorios. Cada conexión puede admitir múltiples repositorios. Puede crear conexiones a un máximo de 25 GitHub cuentas diferentes o crear más de una conexión a una sola cuenta. Tras conectar una aplicación a una GitHub cuenta, CodeDeploy gestiona los permisos de acceso necesarios sin que tengas que realizar ninguna otra acción por tu parte. Las actualizaciones que reflejan esta compatibilidad se realizaron en Especifique la información sobre una revisión almacenada en un GitHub repositorio, Integración CodeDeploy con GitHub y Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub.</p>	30 de mayo de 2017
Temas actualizados	<p>En el pasado, si el CodeDeploy agente detectaba archivos en una ubicación de destino que no formaban parte de la revisión de la aplicación correspondiente a la última implementación correcta, de forma predeterminada, fallaba en la implementación actual. CodeDeploy ahora ofrece opciones sobre la forma en que el agente gestiona estos archivos: no realizar la implementación, conservar el contenido o sobrescribirlo. Cree una implementación con CodeDeploy se actualizó para reflejar esta compatibilidad y Comportamiento de la restauración con contenido existente se agregó la nueva sección. Vuelva a implementar y revertir una implementación con CodeDeploy</p>	16 de mayo de 2017

Cambio	Descripción	Fecha de modificación
Temas actualizados	<p>Ahora se puede asignar un balanceador de carga clásico en Elastic Load Balancing a un grupo de implementación mediante la CodeDeploy consola o. AWS CLI Durante una implementación local, el equilibrador de carga impide que el tráfico de Internet se dirija a una instancia mientras se está implementando, y la vuelve a poner a disposición una vez finalizada dicha implementación. Se han actualizado varios temas para reflejar esta nueva compatibilidad, incluidos Integración con otros AWS servicios, Integrating CodeDeploy with Elastic Load Balancing, Creación de una aplicación para una implementación local (consola), Creación de un grupo de implementación para una implementación local (consola) y la AppSpec sección de «ganchos». Se ha añadido una sección nueva a la guía de solución de problemas, Solución de problemas relacionados con un error o un evento del ciclo de vida de ApplicationStop la BeforeBlockTraffic implementación AfterBlockTraffic.</p>	27 de abril de 2017

Cambio	Descripción	Fecha de modificación
Temas actualizados	<p>Ahora se puede asignar un balanceador de carga clásico en Elastic Load Balancing a un grupo de implementación mediante la CodeDeploy consola o. AWS CLI Durante una implementación local, el equilibrador de carga impide que el tráfico de Internet se dirija a una instancia mientras se está implementando, y la vuelve a poner a disposición una vez finalizada dicha implementación. Se han actualizado varios temas para reflejar esta nueva compatibilidad, incluidos Integración con otros AWS servicios, Integrating CodeDeploy with Elastic Load Balancing, Creación de una aplicación para una implementación local (consola), Creación de un grupo de implementación para una implementación local (consola) y la AppSpec sección de «ganchos». Se ha añadido una sección nueva a la guía de solución de problemas, Solución de problemas relacionados con un error o un evento del ciclo de vida de ApplicationStop la BeforeBlockTraffic implementación AfterBlockTraffic.</p>	1 de mayo de 2017

Cambio	Descripción	Fecha de modificación
Temas actualizados	<p>CodeDeploy ya está disponible en la región de China (Pekín).</p> <p>Para utilizar servicios en la región de China (Pekín) o China (Ningxia), debe disponer de una cuenta y de credenciales para estas regiones. Las cuentas y credenciales de otras AWS regiones no funcionan en las regiones de Beijing y Ningxia, y viceversa.</p> <p>La información sobre algunos recursos para las regiones de China, como los nombres de los paquetes del kit de CodeDeploy recursos y los procedimientos de instalación de los CodeDeploy agentes, no se incluye en esta edición de la Guía del CodeDeploy usuario.</p> <p>Para obtener más información:</p> <ul style="list-style-type: none">• CodeDeploy en <u>Cómo empezar AWS en la región de China (Pekín)</u>• CodeDeploy Guía del usuario para las regiones de China (versión en inglés versión en chino)	29 de marzo de 2017

Cambio	Descripción	Fecha de modificación
Temas nuevos y actualizados	<p>Se han introducido varios temas nuevos para reflejar la nueva CodeDeploy compatibilidad con las implementaciones azules y verdes, un método de implementación en el que las instancias de un grupo de implementación (el entorno original) se sustituyen por un conjunto diferente de instancias (el entorno de reemplazo). Información general de la implementación azul/verde proporciona una explicación detallada de la metodología azul/verde utilizada por CodeDeploy. Los temas adicionales nuevos incluyen Creación de una aplicación para una implementación azul/verde (consola), Creación de un grupo de implementación para una implementación azul/verde de EC2/en las instalaciones (consola) y Configurar un balanceador de carga en Elastic Load Balancing para las implementaciones de CodeDeploy Amazon EC2.</p> <p>Además, se han actualizado numerosos temas, incluidos Cree una implementación con CodeDeploy, Trabajar con configuraciones de implementación en CodeDeploy, Cree una aplicación con CodeDeploy, Trabajar con grupos de despliegue en CodeDeploy, Trabajar con despliegues en CodeDeploy y la AppSpec sección de «ganchos».</p>	25 de enero de 2017
Temas nuevos y actualizados	<p>En un tema nuevo Usa el register-on-premises-instance comando (ARN de sesión de IAM) para registrar una instancia local, se describe cómo autenticar y registrar instancias locales mediante credenciales temporales que se actualizan periódicamente y que se generan a través de ellas. AWS Security Token Service Este enfoque proporciona una mejor compatibilidad para admitir grandes flotas de instancias on-premises que solamente el uso de credenciales estáticas de un usuario de IAM en cada instancia. Working with On-Premises Instances también se ha actualizado para reflejar esta nueva compatibilidad.</p>	28 de diciembre de 2016

Cambio	Descripción	Fecha de modificación
Temas actualizados	CodeDeploy ya está disponible en la región Europa (Londres) (eu-west-2). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	13 de diciembre de 2016
Temas actualizados	CodeDeploy ahora está disponible en la región de Canadá (Central) (ca-central-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	8 de diciembre de 2016
Temas actualizados	CodeDeploy ya está disponible en la región Este de EE. UU. (Ohio) (us-east-2). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	17 de octubre de 2016
Temas nuevos	Una nueva sección, Autenticación y control de acceso, proporciona información completa sobre el uso AWS Identity and Access Management (IAM) y ayuda CodeDeploy a proteger el acceso a los recursos mediante el uso de credenciales. Estas credenciales proporcionan los permisos necesarios para acceder a AWS los recursos, como recuperar las revisiones de las aplicaciones de los buckets de Amazon S3 y leer las etiquetas de las instancias de Amazon EC2.	11 de octubre de 2016

Cambio	Descripción	Fecha de modificación
Tema actualizado	<p>Actualice el CodeDeploy agente en Windows Server se ha actualizado para reflejar la disponibilidad del nuevo actualizador de CodeDeploy agentes para Windows Server. Cuando se instala en una instancia de Windows Server, el actualizador comprobará periódicamente si existen nuevas versiones. Cuando detecte una nueva versión del agente, el actualizador desinstalará la versión actual, si la hay, antes de instalar la última versión.</p>	4 de octubre de 2016
Temas actualizados	<p>CodeDeploy ahora se integra con CloudWatch las alarmas de Amazon, lo que permite detener un despliegue si se produce un cambio en el estado de una alarma específica durante varios períodos consecutivos, tal y como se especifica en el umbral de alarma.</p> <p>CodeDeploy ahora también es compatible con la reversión automática de un despliegue si se cumplen determinadas condiciones, como un fallo en el despliegue o la activación de una alarma.</p> <p>Se han actualizado una serie de temas para reflejar estos cambios, incluidos Cree una aplicación con CodeDeploy, Cree un grupo de despliegue con CodeDeploy, Cambie la configuración del grupo de implementación con CodeDeploy, Deployments, Vuelva a implementar y revertir una implementación con CodeDeploy y Integraciones de productos y servicios con CodeDeploy, junto con un tema nuevo, Supervisión de las implementaciones con CloudWatch alarmas en CodeDeploy.</p>	15 de septiembre de 2016

Cambio	Descripción	Fecha de modificación
Temas nuevos y actualizados	CodeDeploy ahora ofrece integración con Amazon CloudWatch Events. Ahora puede usar CloudWatch Events para iniciar una o más acciones cuando se detectan cambios en el estado de una implementación o en el estado de una instancia que pertenece a un grupo de CodeDeploy implementaciones. Puede incorporar acciones que invoquen AWS Lambda funciones, que publiquen en transmisiones de Kinesis o temas de Amazon SNS, que envíen mensajes a las colas de Amazon SQS o que, a su vez, activen acciones de alarma. CloudWatch Para obtener más información, consulte Supervisión de despliegues con Amazon Events CloudWatch .	9 de septiembre de 2016
Actualizaciones de temas	Los temas Integrating CodeDeploy with Elastic Load Balancing y se Integración con otros AWS servicios han actualizado para incluir una opción adicional de equilibrio o de carga. CodeDeploy ahora es compatible con las versiones Classic Load Balancer y Application Load Balancer disponibles en Elastic Load Balancing.	11 de agosto de 2016
Actualizaciones de temas	CodeDeploy ya está disponible en la región Asia Pacífico (Bombay) (ap-south-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	27 de junio de 2016

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	<p>CodeDeploy ya está disponible en la región Asia Pacífico (Seúl) (ap-northeast-2). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.</p> <p>El índice se ha reorganizado y ahora incluye secciones para instancias, configuraciones de implementación, aplicaciones, grupos de implementaciones, revisiones e implementaciones. Se ha añadido una nueva sección de CodeDeploy tutoriales. A fin de facilitar su uso, se han dividido varios temas largos, como CodeDeploy AppSpec referencia de archivo y Solución de problemas CodeDeploy, en temas más cortos. La información de configuración del CodeDeploy agente se ha trasladado a un tema nuevo, CodeDeploy referencia de configuración del agente.</p>	15 de junio de 2016
Temas nuevos y actualizados	<p>Códigos de error para AWS CodeDeploy proporciona información sobre algunos de los mensajes de error que pueden mostrarse cuando CodeDeploy las implementaciones fallan.</p> <p>Se han actualizado las siguientes secciones en Solución de problemas CodeDeploy para facilitar la resolución de problemas de implementación:</p> <ul style="list-style-type: none"> • Las instancias de EC2 de un grupo de Amazon EC2 Auto Scaling no se pueden lanzar y aparece el error "Heartbeat Timeout" • Evite asociar varios grupos de implementación a un único grupo de Amazon EC2 Auto Scaling 	20 de abril de 2016

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	<p>CodeDeploy ya está disponible en la región Sudamérica (São Paulo) (sa-east-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.</p> <p>Trabajando con el CodeDeploy agente se actualizó para reflejar la nueva opción de configuración: <code>max_revision:</code>, que se utiliza para especificar el número de revisiones de aplicaciones de un grupo de despliegues que se desea que el agente archive. CodeDeploy</p>	10 de marzo de 2016

Cambio	Descripción	Fecha de modificación
Temas nuevos y actualizados	<p>CodeDeploy ahora permite añadir activadores a un grupo de implementaciones para recibir notificaciones sobre eventos relacionados con las implementaciones o las instancias de ese grupo de implementaciones. Estas notificaciones se envían a los destinatarios que están suscritos a un tema de Amazon Simple Notification Service que forma parte de la acción del disparador. También puede utilizar los datos JSON que se crean cuando se activa un disparador en un flujo de trabajo de notificación personalizado. Para obtener más información, consulte Monitoring Deployments with Amazon SNS Event Notifications.</p> <p>Se han actualizado los procedimientos para reflejar el rediseño de la página Application details.</p> <p>Se ha actualizado la sección Las implementaciones no producen error hasta una hora después cuando se termina una instancia durante una implementación en Solución de problemas CodeDeploy.</p> <p>CodeDeploy cuotas se ha actualizado para reflejar los límites revisados del número de grupos de implementación que se pueden asociar a una sola aplicación, los valores permitidos para una configuración con un mínimo de instancias en buen estado y las versiones requeridas del AWS SDK for Ruby.</p>	17 de febrero de 2016

Cambio	Descripción	Fecha de modificación
Temas nuevos y actualizados	<p>CodeDeploy ahora está disponible en la región EE.UU. Oeste (norte de California) (us-west-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la incorporación de esta nueva región.</p> <p>Elige un tipo de CodeDeploy repositorio enumera y describe los tipos de repositorios que ahora admite CodeDeploy. Este tema nuevo se actualizará a medida que se introduzca a la compatibilidad para otros tipos de repositorio.</p> <p>Gestión de las operaciones CodeDeploy de los agentes se actualizó con información sobre el nuevo <code>.version</code> archivo agregado a las instancias para indicar la versión actual del CodeDeploy agente, así como información sobre las versiones compatibles del agente.</p> <p>Se ha añadido a la guía del usuario resalte de la sintaxis para muestras de código, que incluyen ejemplos de JSON y YAML.</p> <p>Añadir un archivo de especificaciones de la aplicación a una revisión de CodeDeploy se ha reorganizado como step-by-step instrucciones.</p>	20 de enero de 2016
Nuevo tema	<p>Implementación de una aplicación en otra cuenta de AWS describe los requisitos de configuración y el proceso para iniciar implementaciones que pertenecen a otra cuenta de la organización, sin necesitar un conjunto completo de credenciales para esa otra cuenta. Esto resulta muy útil para las organizaciones que utilizan múltiples cuentas con fines diferentes, como una asociada a entornos de desarrollo y prueba y otra asociada a un entorno de producción.</p>	30 de diciembre de 2015

Cambio	Descripción	Fecha de modificación
Actualización de tema	Se ha rediseñado el tema Integraciones de productos y servicios con CodeDeploy . Ahora incluye una sección con ejemplos de integración de la comunidad, con listas de publicaciones de blog y ejemplos de vídeo relacionados con las CodeDeploy integraciones.	16 de diciembre de 2015
Actualizaciones de temas	CodeDeploy ya está disponible en la región Asia Pacífico (Singapur) (ap-southeast-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	9 de diciembre de 2015
Actualizaciones de temas	Trabajando con el CodeDeploy agente se actualizó para reflejar la nueva <code>:proxy_uri:</code> opción en el archivo de configuración del CodeDeploy agente. CodeDeploy AppSpec referencia de archivo se ha actualizado con información acerca del uso de una nueva variable de entorno, <code>DEPLOYMENT_GROUP_ID</code> , a los que los scripts de enlace pueden tener acceso durante un evento del ciclo de vida de la implementación.	1 de diciembre de 2015
Actualización de tema	Paso 2: Crear un rol de servicio para CodeDeploy se actualizó para reflejar un nuevo procedimiento de creación de un rol de servicio CodeDeploy e incorporar otras mejoras.	13 de noviembre de 2015
Actualizaciones de temas	CodeDeploy ya está disponible en la región de Europa (Fráncfort) (eu-central-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región. El tema Solución de problemas CodeDeploy se ha actualizado con información acerca de cómo garantizar que la configuración de hora en las instancias sea precisa.	19 de octubre de 2015

Cambio	Descripción	Fecha de modificación
Temas nuevos	<p>AWS CloudFormation plantillas de CodeDeploy referenci ase publicó para reflejar el nuevo AWS CloudFormation apoyo a las CodeDeploy acciones.</p> <p>Se ha creado un tema de Primary Components y se ha introducido la definición de una revisión de destino.</p>	1 de octubre de 2015
Actualizaciones de temas	<p>Cree un grupo de despliegue con CodeDeploy se ha actualizado para reflejar la capacidad para ubicar instancias para un grupo de implementaciones que utiliza búsquedas de comodín.</p> <p>Instance Health se ha actualizado para aclarar el concepto de mínimo de instancias en buen estado.</p>	31 de agosto de 2015
Actualizaciones de temas	CodeDeploy ya está disponible en la región Asia Pacífico (Tokio) (ap-northeast-1). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de esta nueva región.	19 de agosto de 2015

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	<p>CodeDeploy ahora admite despliegues en instancias locales de Red Hat Enterprise Linux (RHEL) e instancias de Amazon EC2. Para obtener más información, consulte los temas siguientes:</p> <ul style="list-style-type: none">• Sistemas operativos compatibles con el agente CodeDeploy• Trabajar con instancias para CodeDeploy• Tutorial: Implementación WordPress en una instancia de Amazon EC2 (Amazon Linux o Red Hat Enterprise Linux y Linux, macOS o Unix)• Tutorial: Implemente una aplicación en una instancia local con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux)	23 de junio de 2015
Actualización de tema	<p>CodeDeploy ahora proporciona un conjunto de variables de entorno que sus scripts de implementación pueden usar durante las implementaciones. Estas variables de entorno incluyen información como el nombre de la CodeDeploy y aplicación actual, el grupo de implementación y el evento del ciclo de vida de la implementación, así como el identificador de CodeDeploy implementación actual. Para obtener más información, consulte el final de la sección AppSpec sección de «ganchos» en la CodeDeploy AppSpec referencia de archivo.</p>	29 de mayo de 2015

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	<p>CodeDeploy ahora proporciona un conjunto de políticas AWS gestionadas en IAM que puede utilizar en lugar de crear manualmente las políticas equivalentes por su cuenta. Entre ellos se incluyen:</p> <ul style="list-style-type: none">• Una política que permite a un usuario registrar CodeDeploy únicamente las revisiones y, después, implementarlas. CodeDeploy• Política para proporcionar a los usuarios acceso total a CodeDeploy los recursos.• Política para proporcionar a un usuario acceso de solo lectura a CodeDeploy los recursos.• Una política para adjuntar a un rol de servicio para que CodeDeploy pueda identificar las instancias de Amazon EC2 por sus etiquetas de Amazon EC2, etiquetas de instancias locales o nombres de grupos de Amazon EC2 Auto Scaling e implementar las revisiones de aplicaciones en consecuencia. <p>Para obtener más información, consulte la sección Ejemplos de políticas administradas por los clientes en Autenticación y control de acceso.</p>	29 de mayo de 2015
Actualizaciones de temas	<p>CodeDeploy ya está disponible en la región de Europa (Irlanda) (eu-west-1) y en la región de Asia Pacífico (Sídney) (ap-southeast-2). Se actualizaron varios temas, incluidos los que contienen instrucciones para configurar el CodeDeploy agente, para reflejar la disponibilidad de estas nuevas regiones.</p>	7 de mayo de 2015

Cambio	Descripción	Fecha de modificación
Temas nuevos	<p>CodeDeploy ahora admite despliegues en instancias locales e instancias de Amazon EC2. Los siguientes temas se han añadido para describir la nueva compatibilidad.</p> <ul style="list-style-type: none">• Working with On-Premises Instances• Tutorial: Implemente una aplicación en una instancia local con CodeDeploy (Windows Server, Ubuntu Server o Red Hat Enterprise Linux)• Working with On-Premises Instances	2 de abril de 2015
Nuevo tema	<p>CodeDeploy recursos se ha añadido.</p>	2 de abril de 2015
Actualización de tema	<p>Solución de problemas CodeDeploy se ha actualizado:</p> <ul style="list-style-type: none">• Una nueva sección Los procesos de larga duración pueden producir un error en la implementación describe los pasos que puede tomar para identificar y solucionar los errores de implementación ocasionados por procesos de larga duración.• La Solución de problemas generales de Amazon EC2 Auto Scaling sección se actualizó para mostrar que CodeDeploy ha aumentado la lógica de tiempo de espera de Amazon EC2 Auto Scaling para el CodeDeploy y agente de cinco minutos a una hora.• Una nueva sección Los enlaces de ciclo de vida de Amazon EC2 Auto Scaling que no coincidan pueden provocar que las implementaciones automáticas en los grupos de Amazon EC2 Auto Scaling se detengan o fallen. describe los pasos que puede realizar para identificar y solucionar errores de las implementaciones automáticas en los grupos de Amazon EC2 Auto Scaling.	2 de abril de 2015

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	<p>Se han actualizado los siguientes temas para reflejar las nuevas recomendaciones para crear políticas personalizadas y adjuntarlas a usuarios y roles en IAM:</p> <ul style="list-style-type: none"> • Configurar una instancia de Amazon EC2 con la que trabajar CodeDeploy • Paso 4: Crear un perfil de instancia de IAM para las instancias de Amazon EC2 • Paso 2: Crear un rol de servicio para CodeDeploy <p>Se han añadido dos secciones a Solución de problemas CodeDeploy:</p> <ul style="list-style-type: none"> • Lista de comprobación de solución de problemas generales • PowerShell Los scripts de Windows no utilizan la versión de 64 bits de Windows de forma predeterminada PowerShell <p>La sección AppSpec sección de «ganchos» en la CodeDeploy AppSpec referencia de archivo se ha actualizado para describir con mayor precisión los eventos del ciclo de vida disponibles.</p>	12 de febrero de 2015
Actualizaciones de temas	<p>Se ha añadido una nueva sección a Solución de problemas CodeDeploy: Las instancias de EC2 de un grupo de Amazon EC2 Auto Scaling no se pueden lanzar y aparece el error "Heartbeat Timeout".</p> <p>Se agregó una CloudBees sección a Integraciones de productos y servicios con CodeDeploy</p>	28 de enero de 2015

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	<p>Se han añadido las siguientes secciones a Solución de problemas CodeDeploy:</p> <ul style="list-style-type: none"> • El uso de algunos editores de texto para crear AppSpec archivos y scripts de shell puede provocar errores en las implementaciones • El uso de Finder en macOS para empaquetar una revisión de la aplicación puede producir un error en la implementación • Solución de problemas relacionados con un error o un evento del ciclo de vida de ApplicationStop la BeforeBlockTraffic implementación AfterBlockTraffic • Solución de problemas de un evento fallido en el ciclo de vida de una DownloadBundle implementación con UnknownError: no abierto para lectura • Solución de problemas generales de Amazon EC2 Auto Scaling 	20 de enero de 2015
Temas nuevos	<p>La sección Integraciones de productos y servicios con CodeDeploy se ha actualizado para incluir los siguientes temas:</p> <ul style="list-style-type: none"> • Integración CodeDeploy con Amazon EC2 Auto Scaling • Tutorial: Se usa CodeDeploy para implementar una aplicación en un grupo de Auto Scaling • Monitoring Deployments • Integrating CodeDeploy with Elastic Load Balancing • Integración CodeDeploy con GitHub • Tutorial: Se utiliza CodeDeploy para implementar una aplicación desde GitHub 	9 de enero de 2015

Cambio	Descripción	Fecha de modificación
Actualizaciones de temas	<ul style="list-style-type: none"> • Se ha añadido la sección Implemente automáticamente desde CodePipeline con CodeDeploy en Integración CodeDeploy con GitHub. Ahora puedes activar automáticamente un despliegue desde un GitHub repositorio siempre que se modifique el código fuente de ese repositorio. • Se ha añadido la sección Solución de problemas de Amazon EC2 Auto Scaling en Solución de problemas CodeDeploy. Esta nueva sección describe cómo solucionar problemas comunes en la implementación a grupos de Amazon EC2 Auto Scaling. • Se ha añadido la nueva sección secundaria "Ejemplos de archivos" a la sección AppSpec Sección de «archivos» (solo para despliegues de EC2/on-premise) de CodeDeploy AppSpec referencia de archivo. Esta nueva subsección incluye varios ejemplos de cómo usar la <code>files</code> sección de un AppSpec archivo para indicar CodeDeploy que se copien archivos o carpetas específicos en ubicaciones específicas de una instancia de Amazon EC2 durante una implementación. 	8 de enero de 2015
Nuevo tema	Monitoring Deployments se agregó. CodeDeploy está integrado con AWS CloudTrail un servicio que captura las llamadas a la API realizadas por o en su nombre CodeDeploy en su AWS cuenta y entrega los archivos de registro a un bucket de Amazon S3 que especifique.	17 de diciembre de 2014
Versión pública inicial	Esta es la primera versión pública de la Guía CodeDeploy del usuario.	12 de noviembre de 2014

AWS Glosario

Para obtener la AWS terminología más reciente, consulte el [AWS glosario](#) de la Glosario de AWS Referencia.

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.