

Crear una canalización sencilla en CodePipeline

En este tutorial, se utiliza CodePipeline para implementar el código mantenido en un CodeCommit repositorio en una única instancia de Amazon EC2. La canalización se activa cuando envía un cambio al repositorio de CodeCommit. La canalización implementa los cambios en una instancia de Amazon EC2 que se utiliza CodeDeploy como servicio de implementación.

La canalización tiene dos etapas:

- Una etapa de origen (**Source (Origen)**) para la acción de origen de CodeCommit.
- Una etapa de implementación (**Deploy (Implementar)**) para la acción de implementación de CodeDeploy.

La forma más sencilla de empezar a trabajar con AWS CodePipeline es utilizar el asistente **Create Pipeline (Crear canalización)** de la consola de CodePipeline.

nota

Paso 1: Crear un repositorio de CodeCommit

En primer lugar, debe crear un repositorio en CodeCommit. Su canalización obtendrá el código fuente de este repositorio cuando se ejecute. También creará un repositorio local en el que conservará el código y lo actualizará antes de enviarlo al repositorio de CodeCommit.

Para crear un repositorio de CodeCommit

1. Abra la CodeCommit consola en <https://console.aws.amazon.com/codecommit/>.
2. En el selector de regiones, elija el Región de AWS lugar en el que desea crear el repositorio y la canalización. Para obtener más información, consulte [Regiones de AWS y puntos de enlace](#).
3. En la página **Repositories**, seleccione **Create repository**.

4. En la página **Create repository (Crear repositorio)**, en **Repository name (Nombre del repositorio)**, escriba un nombre para el repositorio (por ejemplo, **MyDemoRepo**).
5. Seleccione **Create** (Crear).

nota

Los demás pasos de este tutorial utilizan **MyDemoRepo** como nombre para el repositorio de CodeCommit. Si elige otro nombre, asegúrese de utilizarlo durante todo el tutorial.

Para configurar un repositorio local

En este paso, configurará un repositorio local para conectarse a su repositorio remoto de CodeCommit.

nota

No es necesario configurar un repositorio local. También puede usar la consola para cargar archivos como se describe en [Paso 2: Agregar código de ejemplo al repositorio de CodeCommit](#).

1. Con el nuevo repositorio abierto en la consola, elija **Clone URL (Clonar URL)** en la parte superior derecha de la página y, a continuación, elija **Clone SSH (Clonar SSH)**. La dirección para clonar el repositorio Git se copia en el portapapeles.
2. En el terminal o en la línea de comandos, acceda a un directorio local en el que desea que se almacene el repositorio local. En este tutorial, utilizamos `/tmp`.
3. Ejecute el siguiente comando para clonar el repositorio, sustituyendo la dirección SSH por la que ha copiado en el paso anterior. Este comando crea un directorio denominado `MyDemoRepo`. Copie una aplicación de ejemplo en este directorio.

```
git clone ssh://git-codecommit.us-west-2.amazonaws.com/v1/repos/MyDemoRepo
```

Paso 2: Agregar código de ejemplo al repositorio de CodeCommit

En este paso, descargará el código de una aplicación de ejemplo creada para un tutorial de ejemplo de CodeDeploy y lo añadirá al repositorio de CodeCommit.

1. A continuación, descargue un ejemplo y guárdelo en una carpeta o directorio del equipo local.
 - a. Elija una de las siguientes opciones. Elija `SampleApp_Linux.zip` si desea seguir los pasos de este tutorial para instancias de Linux.
 - Si desea realizar la implementación en instancias de Amazon Linux mediante CodeDeploy, descargue la aplicación de ejemplo aquí: [SampleApp Linux.zip](#).
 - Si desea realizar la implementación en instancias de Windows Server mediante CodeDeploy, descargue la aplicación de ejemplo aquí: [SampleApp Windows.zip](#).

La aplicación de ejemplo contiene los siguientes archivos para su implementación: CodeDeploy

- `appspec.yml`— El archivo de especificación de la aplicación (AppSpec archivo) es un archivo con formato [YAML](#) que se utiliza CodeDeploy para gestionar una implementación. Para obtener más información sobre el AppSpec archivo, consulte la [referencia al CodeDeploy AppSpec archivo](#) en la *Guía del AWS CodeDeploy usuario*.
- `index.html`— El archivo de índice contiene la página principal de la aplicación de ejemplo implementada.
- `LICENSE.txt`— El archivo de licencia contiene información sobre la licencia de la aplicación de ejemplo.
- Archivos para scripts: la aplicación de ejemplo usa scripts para escribir archivos de texto en una ubicación de la instancia. Se escribe un archivo para cada uno de los varios eventos del ciclo de vida de la CodeDeploy implementación de la siguiente manera:
 - `scripts` Carpeta (solo muestra de Linux): la carpeta contiene los siguientes scripts de shell para instalar las dependencias e iniciar y detener la aplicación de ejemplo para la implementación automatizada: `install_dependencies`, `start_server`, `ystop_server`.
 - (Solo ejemplo de Windows) `before-install.bat`: se trata de un script por lotes para el evento del ciclo de vida de la `BeforeInstall` implementación, que se ejecutará para eliminar los archivos antiguos escritos durante las implementaciones anteriores de este

ejemplo y crear una ubicación en la instancia en la que escribir los nuevos archivos.

b. Descargue el archivo comprimido (en zip).

2. Descomprima los archivos de [SampleApp Linux.zip](#) en el directorio local que creó anteriormente (por ejemplo, /tmp/MyDemoRepo o c:\temp\MyDemoRepo).

Asegúrese de colocar los archivos directamente en su repositorio local. No incluya una carpeta SampleApp_Linux. En su máquina Linux, macOS o Unix local, por ejemplo, la jerarquía de directorios y archivos debería tener este aspecto:

```
/tmp
├── MyDemoRepo
│   ├── appspec.yml
│   ├── index.html
│   ├── LICENSE.txt
│   └── scripts
│       ├── install_dependencies
│       ├── start_server
│       └── stop_server
```

3. Para subir archivos a su repositorio, utilice alguno de los métodos siguientes.
 - a. Para utilizar la consola de CodeCommit para cargar los archivos:
 - i. Abra la consola de CodeCommit y elija el repositorio en la lista **Repositories (Repositorios)**.
 - ii. Elija **Add file (Añadir archivo)** y, a continuación, **Upload file (Cargar archivo)**.
 - iii. Seleccione **Choose file (Elegir archivo)** y, a continuación, busque el archivo. Para añadir un archivo a una carpeta, seleccione **Crear archivo** y, a continuación, introduzca el nombre de la carpeta con el nombre del archivo, por ejemplo scripts/install_dependencies. Pegue el contenido del archivo en el nuevo archivo.

Para confirmar el cambio, introduzca su nombre de usuario y la dirección de correo electrónico.

Seleccione **Commit changes (Confirmar cambios)**.

- iv. Repita este paso para cada archivo.

El contenido de su repositorio debe ser a la que se muestran a:

```
| -- appspec.yml
| -- index.html
| -- LICENSE.txt
L-- scripts
    |-- install_dependencies
    |-- start_server
    L-- stop_server
```

- b. Para usar los comandos de git para subir tus archivos, sigue estos pasos:

- i. Cambiar los directorios a su repositorio local:

ii. *(For Linux, macOS, or Unix)* `cd /tmp/MyDemoRepo`

(For Windows) `cd c:\temp\MyDemoRepo`

- iii. Ejecute el siguiente comando para preparar todos los archivos de una vez:

```
git add -A
```

- iv. Ejecute el siguiente comando para confirmar los archivos con un mensaje de confirmación:

```
git commit -m "Add sample application files"
```

- v. Ejecute el siguiente comando para enviar los archivos de su repositorio local al repositorio de CodeCommit:

```
git push
```

4. Los archivos descargados y añadidos a su repositorio local ya se han añadido a la ramificación `main` de su repositorio de CodeCommit `MyDemoRepo` y están listos para ser incluidos en una canalización.

Paso 3: crear una instancia de Amazon EC2 para Linux e instalar el agente CodeDeploy

En este paso, cree la instancia de Amazon EC2 en la que pueda implementar una aplicación de ejemplo. Como parte de este proceso, cree un rol de instancia que permita instalar y administrar el CodeDeploy agente en la instancia. Este agente de CodeDeploy es un paquete de software que permite utilizar una instancia en las implementaciones de CodeDeploy. También adjuntas políticas que permiten a la instancia recuperar los archivos que el CodeDeploy agente utiliza para implementar la aplicación y permitir que SSM administre la instancia.

Para crear un rol de instancia

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de la consola, elija **Roles**.
3. Elija **Create role** (Crear rol).
4. En **Seleccione el tipo de entidad de confianza**, seleccione **Servicio de AWS**. En **Elija un caso de uso**, seleccione **EC2**. En **Select your use case (Seleccionar su caso de uso)**, elija **EC2**. Elija **Next: Permissions** (Siguiente: permisos).
5. Busque y seleccione la política nombrada **AmazonEC2RoleforAWSCodeDeploy**.
6. Busque y seleccione la política nombrada **AmazonSSMManagedInstanceCore**. Elija **Next: Tags** (Siguiente: etiquetas).
7. Elija **Next: Review** (Siguiente: revisar). Escriba el nombre del rol (por ejemplo, **EC2InstanceRole**).

nota

Anote el nombre del rol para utilizarlo en el siguiente paso. Tendrá que elegir este rol cuando cree la instancia.

Elija **Create role** (Crear rol).

Para lanzar una instancia

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En la barra de navegación lateral, selecciona **Instancias** y selecciona **Lanzar instancias** en la parte superior de la página.
3. En **Name (Nombre)**, escriba **MyCodePipelineDemo**. Esto asigna a la instancia una **clave** de etiqueta **Name** y un **valor** de etiqueta de **MyCodePipelineDemo**. Más adelante, se crea una CodeDeploy aplicación que implementa la aplicación de ejemplo en esta instancia. CodeDeploy selecciona las instancias que se van a implementar en función de las etiquetas.
4. En **Imágenes de aplicaciones y sistemas operativos (Amazon Machine Image)**, busque la opción AMI de **Amazon Linux** con el AWS logotipo y asegúrese de que esté seleccionada. (Esta AMI se describe como AMI (HVM) de Amazon Linux 2 y tiene la etiqueta «Apta para la capa gratuita»).
5. En **Tipo de instancia**, elija el t2.micro tipo apta para la capa gratuita como configuración de hardware de su instancia.
6. En **Par de claves (inicio de sesión)**, elija un key pair o cree una.

También puede elegir **Continuar sin un key pair**.

nota

A efectos de este tutorial, puede continuar sin utilizar un par de claves. Si desea usar SSH para conectarse a sus instancias, cree o use un par de claves.

7. En **Configuración de red**, haga lo siguiente.

En **Asignación automática de IP pública**, asegúrese de que el estado sea **Habilitar**.

 - Junto a **Assign a security group (Asignar un grupo de seguridad)**, elija **Create a new security group (Crear un nuevo grupo de seguridad)**.
 - En la fila de **SSH**, en **Tipo de fuente**, elija **Mi IP**.
 - Elija **Agregar grupo de seguridad, HTTP** y, a continuación, en **Tipo de fuente**, elija **Mi IP**.
8. Amplíe **Advanced details** (Detalles avanzados). En el **perfil de instancia de IAM**, elija el rol de IAM que creó en el procedimiento anterior (por ejemplo, **EC2InstanceRole**).
9. En **Resumen**, en **Número de instancias**, introduzca 1..
10. Seleccione **Launch Instance (Lanzar instancia)**.
11. Puede ver el estado del lanzamiento en la página **Instances**. Al lanzar una instancia, su estado inicial es pending. Una vez iniciada la instancia, el estado cambia a running y recibe un nombre de DNS público. (Si la columna **Public**

DNS no se muestra, haga clic en el icono **Show/Hide** y después seleccione **Public DNS**.)

Paso 4: Crear una aplicación en CodeDeploy

En CodeDeploy, una *aplicación* es un recurso que contiene la aplicación de software que desea implementar. Más adelante, utilizará esta aplicación CodePipeline para automatizar las implementaciones de la aplicación de muestra en su instancia de Amazon EC2.

En primer lugar, debe crear un rol que permita a CodeDeploy realizar implementaciones. A continuación, debe crear una aplicación de CodeDeploy.

Para crear un rol de servicio de CodeDeploy

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de la consola, elija **Roles**.
3. Elija **Create role** (Crear rol).
4. En **Seleccionar entidad de confianza**, elija **Servicio de AWS**. En **Use case** (Caso de uso), elija **CodeDeploy**. Elige una **CodeDeploy** de las opciones de la lista. Elija **Siguiente**. La política administrada `AWSCodeDeployRole` ya está asociada al rol.
5. Elija **Siguiente**.
6. Especifique un nombre para el rol (por ejemplo, `CodeDeployRole`) y elija **Create role** (Crear rol).

Para crear una aplicación en CodeDeploy

1. Abra la CodeDeploy consola en <https://console.aws.amazon.com/codedeploy>.
2. Si la página **Aplicaciones** no aparece, en el menú, seleccione **Aplicaciones**.
3. Elija **Create application**.
4. En **Application name (Nombre de aplicación)**, escriba `MyDemoApplication`.
5. En **Compute platform (Plataforma de computación)**, elija **EC2/On-Premises**.
6. Elija **Create application**.

Para crear un grupo de implementaciones en CodeDeploy

Un *grupo de implementación* es un recurso que define la configuración relacionada con la implementación, como las instancias que implementar y con qué rapidez implementarlas.

1. En la página que muestra su aplicación, elija **Create deployment group (Crear grupo de implementaciones)**.
2. En **Deployment group name (Nombre de grupo de implementación)**, escriba `MyDemoDeploymentGroup`.
3. En **Función de servicio**, elija la función de servicio que creó anteriormente (por ejemplo, `CodeDeployRole`).
4. En **Deployment type (Tipo de implementación)**, elija **In-place (In situ)**.
5. En **Environment configuration (Configuración de entorno)**, elija **Amazon EC2 Instances (Instancias de Amazon EC2)**. En el campo **Clave**, introduzca `Name`. En el campo **Valor**, introduce el nombre que usaste para etiquetar la instancia (por ejemplo, `MyCodePipelineDemo`).
6. En **Configuración del agente con AWS Systems Manager**, seleccione **Ahora y programe las actualizaciones**. De este modo, se instalará el agente en la instancia. La instancia de Linux ya está configurada con el agente SSM y ahora se actualizará con el CodeDeploy agente.
7. En **Deployment configuration (Configuración de implementación)**, elija `CodeDeployDefault.OneAtATime`.
8. En **Load Balancer**, asegúrate de que la opción **Habilitar equilibrio de carga** no esté seleccionada. No es necesario configurar un balanceador de carga ni elegir un grupo de destino para este ejemplo.
9. Elija **Create deployment group**.

Paso 5: Crear la primera canalización en CodePipeline

Ya está preparado para crear y ejecutar su primera canalización. En este paso, creará una canalización que se ejecuta automáticamente cuando se inserte código en el repositorio de CodeCommit.

Para crear una canalización de CodePipeline

1. Inicie sesión en la CodePipeline consola AWS Management Console y ábrala en <http://console.aws.amazon.com/codesuite/codepipeline/home>.
Abra la CodePipeline consola en <https://console.aws.amazon.com/codepipeline/>.

2. Elija **Create pipeline**.
3. En **Step 1: Choose pipeline settings (Paso 1: Elegir configuración de canalización)**, en **Pipeline name (Nombre de canalización)**, escriba **MyFirstPipeline**.
4. En **Service role (Rol de servicio)**, elija **New service role (Nuevo rol de servicio)** para permitir que CodePipeline cree un rol de servicio en IAM.
5. En **Advanced settings (Configuración avanzada)**, deje los valores predeterminados y elija **Next (Siguiente)**.
6. En el **Step 2: Add source stage (Paso 2: Añadir etapa de código fuente)**, en **Source provider (Proveedor de código fuente)**, elija **CodeCommit**. En **Repository name (Nombre de repositorio)**, elija el nombre del repositorio de CodeCommit que ha creado en [Paso 1: Crear un repositorio de CodeCommit](#). En **Branch name (Nombre de la ramificación)**, elija **main** y, a continuación, elija **Next step (Paso siguiente)**.

Tras seleccionar el nombre y la rama del repositorio, aparecerá un mensaje con la regla de Amazon CloudWatch Events que se va a crear para esta canalización.

En **Change detection options**, deje los valores predeterminados. Esto permite CodePipeline utilizar Amazon CloudWatch Events para detectar cambios en el repositorio de origen.

Elija **Siguiente**.

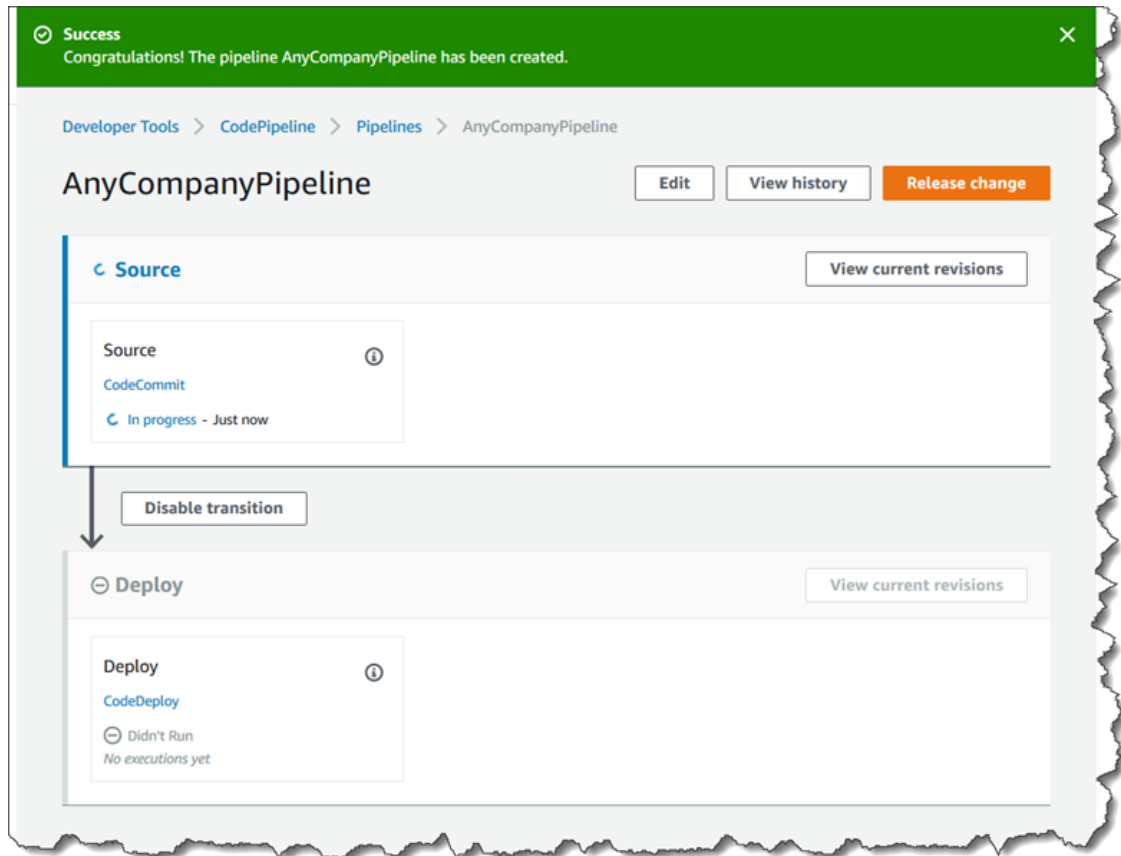
7. En **Step 3: Add build stage (Paso 3: Añadir etapa de compilación)**, elija **Skip build stage (Omitir etapa de compilación)** y, a continuación, acepte el mensaje de advertencia eligiendo **Skip (Omitir)** una vez más. Elija **Siguiente**.

nota

En este tutorial, va a implementar código que no requiere ningún servicio de compilación, por lo que puede omitir este paso. Sin embargo, si es necesario compilar el código fuente antes de implementarlo en instancias, puede configurar [CodeBuild](#) en este paso.

8. En el **Step 4: Add deploy stage (Paso 4: Añadir etapa de implementación)**, en **Deploy provider (Proveedor de implementación)**, elija **CodeDeploy**. En **Application name (Nombre de aplicación)**, escriba **MyDemoApplication**. En **Deployment group (Grupo de implementación)**, elija **MyDemoDeploymentGroup** y, a continuación, elija **Next step (Paso siguiente)**.
9. En **Step 5: Review**, revise la información y después elija **Create pipeline**.

10. La canalización comienza a ejecutarse después de crearla. Descarga el código del repositorio de CodeCommit y crea una implementación de CodeDeploy en la instancia EC2. Puede ver los mensajes de progreso y de éxito y error a medida que el CodePipeline ejemplo implementa la página web en la instancia de Amazon EC2 de la CodeDeploy implementación.



¡Enhorabuena! Acaba de crear una canalización sencilla en CodePipeline.

A continuación, podrá verificar los resultados.

Para comprobar que la canalización se ha ejecutado correctamente

1. Vea el progreso inicial de la canalización. El estado de cada etapa cambia de **No executions yet** (Sin ejecuciones) a **In Progress** (En curso) y después a **Succeeded** (Realizado correctamente) o **Failed** (Error). La canalización debería completar la primera ejecución en unos minutos.
2. Una **vez** que aparezca el estado de la canalización, en el área de estado de la etapa de **implementación**, elija **CodeDeploy**. Se abrirá la consola de CodeDeploy. Si no se muestra **Succeeded (Correcto)** consulte [Solución de problemas de CodePipeline](#).

3. En la pestaña **Implementaciones**, elija el ID de la implementación. En la página de la implementación, en la sección **Eventos del ciclo de vida de la implementación**, elija el ID de la instancia. Se abrirá la consola de EC2.
4. En la pestaña **Description (Descripción)**, en **Public DNS (DNS público)**, copie la dirección (por ejemplo, `ec2-192-0-2-1.us-west-2.compute.amazonaws.com`) y después péguela en la barra de direcciones de su explorador web.

Se muestra la página web de la aplicación de ejemplo que descargaste y subiste a tu CodeCommit repositorio.

Para obtener más información sobre las etapas, las acciones y cómo funcionan las canalizaciones, consulte [CodePipeline conceptos](#).

Paso 6: Modificar código del repositorio de CodeCommit

Se ha configurado la canalización de modo que se ejecute siempre que se realicen cambios en el código de su repositorio de CodeCommit. En este paso, va a realizar cambios en el archivo HTML que forma parte de la aplicación CodeDeploy de ejemplo en el repositorio de CodeCommit. Cuando envíe estos cambios, la canalización se ejecuta de nuevo y los cambios que realice estarán visibles en la dirección web a la que accedió anteriormente.

1. Cambiar los directorios a su repositorio local:
 2. *(For Linux, macOS, or Unix)* `cd /tmp/MyDemoRepo`
(For Windows) `cd c:\temp\MyDemoRepo`
 3. Use un editor de texto para modificar el archivo `index.html`:
 4. *(For Linux or Unix)* `gedit index.html`
 5. *(For OS X)* `open -e index.html`
(For Windows) `notepad index.html`
 6. 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.
 7. `<!DOCTYPE html>`

```
8. <html>
9. <head>
10.   <title>Updated Sample Deployment</title>
11.   <style>
12.     body {
13.       color: #000000;
14.       background-color: #CCFFCC;
15.       font-family: Arial, sans-serif;
16.       font-size:14px;
17.     }
18.
19.     h1 {
20.       font-size: 250%;
21.       font-weight: normal;
22.       margin-bottom: 0;
23.     }
24.
25.     h2 {
26.       font-size: 175%;
27.       font-weight: normal;
28.       margin-bottom: 0;
29.     }
30.   </style>
```

```
31. </head>

32. <body>

33.   <div align="center"><h1>Updated Sample
      Deployment</h1></div>

34.   <div align="center"><h2>This application was updated
      using CodePipeline, CodeCommit, and CodeDeploy.</h2></div>

35.   <div align="center">

36.     <p>Learn more:</p>

37.     <p><a
      href="https://docs.aws.amazon.com/codepipeline/latest/user
      guide/">CodePipeline User Guide</a></p>

38.     <p><a
      href="https://docs.aws.amazon.com/codecommit/latest/usergu
      ide/">CodeCommit User Guide</a></p>

39.     <p><a
      href="https://docs.aws.amazon.com/codedeploy/latest/usergu
      ide/">CodeDeploy User Guide</a></p>

40.   </div>

41. </body>

</html>
```

42. Confirme e integre los cambios con su repositorio de CodeCommit ejecutando los siguientes comandos, uno a la vez:

```
git commit -am "Updated sample application files"

git push
```

Para comprobar que la canalización se ha ejecutado correctamente

1. Vea el progreso inicial de la canalización. El estado de cada etapa cambia de **No executions yet** (Sin ejecuciones) a **In Progress** (En curso) y después

a **Succeeded** (Realizado correctamente) o **Failed** (Error). La ejecución de la canalización debería completarse en unos minutos.

2. Cuando aparezca **Succeeded (Correcto)** en el estado de la acción, actualice la página de demostración a la que ha accedido anteriormente en el navegador.

Aparece la página web actualizada.

Paso 7: Limpiar recursos

Puede utilizar algunos de los recursos que ha creado en este tutorial para otros tutoriales de esta guía. Por ejemplo, puede reutilizar la implementación y la aplicación de CodeDeploy. Sin embargo, una vez completado este o cualquier otro tutorial, debe eliminar la canalización y los recursos que utiliza para que no se le cobre por el uso continuado de esos recursos. Primero, elimine la canalización, luego la CodeDeploy aplicación y su instancia de Amazon EC2 asociada y, por último, el CodeCommit repositorio.