***Proyecto\_Previo: Despliegue Automático de una Aplicación MVC Core sin Base de Datos en AWS Elastic Beanstalk utilizando AWS CodePipeline (Sin CodeBuild)***

**Objetivo del Proyecto**

Crear una canalización de integración continua (CI/CD) usando **AWS CodePipeline** para desplegar una aplicación **ASP.NET Core MVC** en **Elastic Beanstalk**, sin usar **CodeBuild**. El proceso de compilación de la aplicación se realizará **localmente**, generando un archivo comprimido .zip de la aplicación que luego será desplegado en **Elastic Beanstalk**.

**Requisitos del Proyecto**

1. **Crear un Proyecto MVC Core**:
   * Utiliza **Visual Studio** para crear un proyecto de tipo **ASP.NET Core Web Application**.
   * Elige la plantilla **Web Application (Model-View-Controller)**.
   * El proyecto **no debe tener una base de datos**, solo debe ser una simple aplicación con algunas vistas como **Home**, **About**, y **Contact**.
2. **Compilación Local**:
   * **Compila el proyecto** en **modo Release** (no en Debug) y asegúrate de que esté listo para producción.
   * Usa el siguiente comando para generar el archivo de salida: ***dotnet publish --configuration Release --output ./publish***
3. **Empaquetado**:

* Una vez compilada la aplicación, localiza la carpeta **publish**
* **Comprime** los archivos en un zip que hay DENTRO de la carpeta, NO LA CARPETA/ en un archivo .zip. Este archivo será el que se utilizará para el despliegue en **Elastic Beanstalk**.

1. **Subir el Proyecto a GitHub**:

* Crea un repositorio en **GitHub** y sube el código fuente de tu proyecto MVC Core. Esto permitirá integrar tu código con **AWS CodePipeline**.

**Pasos en AWS: Configuración de Recursos**

**1. Crear dos Buckets S3 (si no los tienes ya)**

1. Ve a **AWS S3** y crea un **bucket** donde almacenarás el archivo comprimido .zip de tu aplicación.
   * Nombre de los buckets(ejemplo): almacen-artefactos-pipeline, almacen-recursos-beanstalk
   * **Políticas de acceso**: Asegúrate de configurar un rol con los permisos necesarios para que **AWS CodePipeline** pueda acceder a estos bucket.

**2. Crear una Aplicación y Entorno en Elastic Beanstalk**

1. Entra a **AWS Elastic Beanstalk** y crea una **nueva aplicación** para tu proyecto MVC Core.
   * Elige la plataforma **.NET Core on Linux** o **.NET Core on Windows**, según prefieras.
   * Crea un **entorno** de tipo **Web Server**.
   * No es necesario configurar una base de datos, ya que el proyecto no tiene base de datos asociada.

**Configuración de CodePipeline**

**3. Crear la Canalización (Pipeline)**

1. Ve a **AWS CodePipeline** y crea una **nueva pipeline**.
   * Elige un nombre para tu pipeline, como **mvc-core-pipeline**.
   * En **Artifact Store**, selecciona el bucket S3 que creaste previamente: almacen-artefactos-pipeline.

**4. Configurar la Etapa de Origen (Source)**

1. **GitHub**:
   * Configura la fuente del código desde **GitHub**.
   * Conecta tu cuenta de **GitHub** con AWS CodePipeline.
   * Elige el repositorio donde subiste tu código fuente del proyecto MVC Core.
   * **Configura un webhook** para que la pipeline se dispare automáticamente cada vez que hagas un **push** al repositorio de **GitHub**.

**5. Configurar la Etapa de Despliegue (Deploy)**

1. En la **etapa de despliegue**, selecciona **Elastic Beanstalk** como el servicio de destino.
2. **Configura el despliegue**:
   * El artefacto de la fuente será el archivo .zip que subiste a **GitHub**.
   * Elastic Beanstalk tomará este archivo comprimido y lo desplegará en el entorno web que creaste previamente.

**Verificación y Despliegue**

1. **Realiza un Commit** en **GitHub**:
   * Cuando hagas un commit y push del código al repositorio de **GitHub**, esto activará automáticamente la pipeline, que descargará el archivo desde GitHub y lo desplegará en Elastic Beanstalk.
2. **Verifica en AWS CodePipeline**:
   * Ve a **AWS CodePipeline** para verificar que la pipeline se ha ejecutado correctamente.
   * Verifica que haya pasado por las etapas: **Source**, **Deploy**, y que haya llegado a **Elastic Beanstalk**.
3. **Verifica en Elastic Beanstalk**:
   * Una vez completado el despliegue, visita la URL proporcionada por **Elastic Beanstalk** para verificar que tu aplicación está funcionando correctamente.
   * Ejemplo de URL: http://<nombre-de-tu-app>.elasticbeanstalk.com.

**Puntos Importantes:**

1. **Código Fuente**: Asegúrate de que solo subas el código fuente a GitHub. El archivo .zip con el artefacto compilado se subirá a **S3**.
2. **Configuración de Permisos**: Revisa que los permisos de **S3** y **Elastic Beanstalk** estén correctamente configurados para que **AWS CodePipeline** pueda acceder a ellos sin problemas.
3. **Elastic Beanstalk**: Elastic Beanstalk se encargará de la infraestructura y despliegue de la aplicación, permitiéndote centrarte solo en el código.