

03 - Despliegue Azure DevOps

Tipo	Area Temática	Duración
Guia	Automatización de despliegues en la nube	2 horas

Tema:

Despliegue de Aplicaciones con Azure DevOps

Introducción

En esta práctica, configurarás y gestionarás pipelines de despliegue en Azure DevOps. Aprenderás a configurar un pipeline completo para una aplicación web, integrando Azure DevOps con AWS para el despliegue en la nube. Además, te enfocarás en la ejecución de pruebas unitarias para asegurar la calidad del código antes del despliegue.

Requisitos Previos

1. **Cuenta en Azure DevOps:** Asegúrate de tener acceso a un proyecto en Azure DevOps.
2. **Cuenta en AWS:** Acceso a una cuenta de AWS con permisos para crear y gestionar recursos (EC2, S3, etc.).
3. **Repositorio Git:** Código fuente de una aplicación web disponible en un repositorio Git, que incluya un conjunto de pruebas unitarias.

Paso 1: Configuración Inicial en Azure DevOps

1.1 Crear un Proyecto en Azure DevOps

1. Inicia sesión en Azure DevOps.
2. Navega a "New Project" y crea un nuevo proyecto.
 - **Nombre del Proyecto:** `Despliegue-App-Web`
 - **Descripción:** `Proyecto para despliegue automatizado de una aplicación web`
 - **Visibilidad:** Private (o Public, según prefieras)
3. Haz clic en "Create".

1.2 Configurar Repositorio Git

1. Dentro del proyecto, ve a "Repos" y selecciona "Importar un Repositorio".
2. Ingresa la URL del repositorio Git que contiene tu aplicación web.
3. Haz clic en "Importar" para añadir el código a tu proyecto en Azure DevOps.

Paso 2: Configuración del Pipeline de Build

2.1 Crear un Pipeline de Build

1. Navega a "Pipelines" > "Pipelines" y selecciona "New Pipeline".
2. Elige la opción "Azure Repos Git (YAML)" y selecciona el repositorio que importaste.
3. Selecciona "Starter Pipeline" y reemplaza el contenido YAML con la configuración básica:

```
trigger:
- main

pool:
  vmImage: 'ubuntu-latest'

steps:
- task: NodeTool@0
  inputs:
    versionSpec: '14.x'
```

```

    displayName: 'Instalar Node.js'

- script: |
    npm install
    npm test
    displayName: 'Instalar dependencias y ejecutar pruebas unitarias'

- task: PublishBuildArtifacts@1
  inputs:
    pathToPublish: 'dist/'
    artifactName: 'drop'
    publishLocation: 'Container'
    displayName: 'Publicar Artefactos de Build'

```

1. Guarda el archivo YAML y ejecuta el pipeline haciendo clic en "Run".

2.2 Configurar Triggers Automáticos

1. En el YAML del pipeline, añade la siguiente configuración para que el pipeline se ejecute automáticamente con cada commit en la rama `main`:

```

trigger:
- main

```

2.3 Verificación de Resultados de Pruebas Unitarias

1. Asegúrate de que las pruebas unitarias se ejecuten correctamente como parte del pipeline.
2. Revisa los logs del pipeline para verificar que todas las pruebas pasen sin errores.
3. Si alguna prueba falla, detén el pipeline y corrige los errores antes de continuar.

Paso 3: Configuración del Pipeline de Release

3.1 Crear un Pipeline de Release

1. Navega a "Pipelines" > "Releases" y selecciona "New Pipeline".
2. Selecciona la opción "Empty job".
3. En la sección "Artifacts", selecciona el artefacto que se generó en el pipeline de build.
4. Añade una nueva "Stage" llamada `Deploy to AWS`.

3.2 Configurar Despliegue en AWS

1. Dentro de la etapa `Deploy to AWS`, añade una tarea para desplegar en un bucket de S3:

```
- task: AmazonWebServices.s3Upload@1
  displayName: 'Subir Artefacto a S3'
  inputs:
    awsCredentials: 'YourAWSCredentialsServiceConnection'
    regionName: 'us-west-2'
    bucketName: 'your-s3-bucket'
    sourceFolder: '$(System.DefaultWorkingDirectory)/_Despliegue-App-Web/drop'
    targetFolder: '/'
```

1. Añade una tarea adicional para desplegar la aplicación en un servidor EC2 (si aplica) utilizando AWS CLI o un script personalizado.

3.3 Configurar Aprobaciones y Despliegues Automáticos

1. En la etapa `Deploy to AWS`, selecciona "Pre-deployment conditions" y añade aprobaciones si es necesario.
2. Configura el pipeline para que el despliegue sea automático una vez que el build haya finalizado con éxito.

Paso 4: Monitoreo y Resolución de Problemas

4.1 Monitorear el Pipeline

1. Navega a "Pipelines" > "Pipelines" y selecciona el pipeline en ejecución.
2. Observa las etapas del pipeline para asegurarte de que no haya fallos.

4.2 Resolución de Problemas Comunes

1. Fallo en Pruebas Unitarias:

- Verifica el código fuente para identificar posibles errores en las pruebas.
- Revisa los logs del pipeline para obtener detalles específicos sobre el fallo.
- Asegúrate de que las dependencias necesarias estén correctamente instaladas.

2. Error en la Conexión AWS:

- Verifica que las credenciales de AWS estén configuradas correctamente en Azure DevOps.
- Revisa los logs de la tarea que falló para obtener más detalles.

Conclusión

Esta práctica te ha guiado a través de la configuración de pipelines en Azure DevOps para el despliegue automatizado de una aplicación web. Has aprendido a ejecutar pruebas unitarias como parte del proceso de build, asegurando que solo el código de alta calidad sea desplegado. Ahora deberías ser capaz de configurar pipelines que integren diferentes herramientas y servicios para realizar despliegues continuos y monitorear el rendimiento de tus aplicaciones.

¡Felicidades por completar la práctica!