- Inicio
- Implementación de contenedores de Docker en aplicaciones web de Azure App Service
- Material:
- Objetivos
- Desarrollo
    - Crear un Repositorio en Azure Devops
    - Crear Pipelines CI y CD
        - Importar y ejecutar el pipeline CI
        - Importar y editar el pipeline CD Deploy
        - Editar Pipeline Con Valores de conexion
        - Crear Service Principal y Service Connection
        - Actualizar y Ejecutar el Pipeline

Inicio

Implementación de contenedores de Docker en aplicaciones web de Azure App Service

Autor: Juan Guillermo Álvarez

Institución: SmartData

Fecha: Abril 2025

Este documento presenta una guía práctica para la implementación de contenedores Docker en aplicaciones web utilizando Azure App Service, destacando aspectos técnicos clave y buenas prácticas.

Implementación de contenedores de Docker en aplicaciones web de Azure App Service

Material:

- AZ400_M02_L06_Deploy_Docker_containers_to_Azure_App_Service_web_apps.md
- Repositorio https://github.com/MicrosoftLearning/eShopOnWeb.git
- https://learn.microsoft.com/es-es/rest/api/azure/devops/?view=azure-devops-rest-7.2 apis
- https://github.com/ArmandoTG/aks.git ejercicio 2 aksgd3

Objetivos

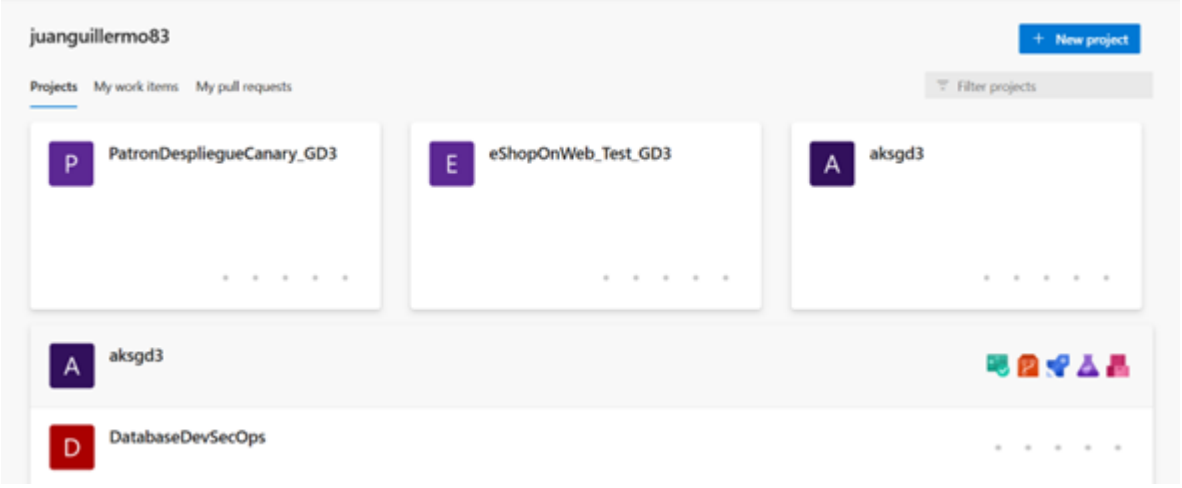
- Crear un proyecto y cargar un repositorio en Azure Devops
- Configurar Pipelines
- Crear una imagen de Docker personalizada mediante un agente de Linux hospedado por Microsoft.
- Crear Un Azure container Registry
- Insertar una imagen en Azure Container Registry.
- Implementar una imagen de Docker como un contenedor en Azure App Service mediante Azure DevOps.

Desarrollo

Crear un Repositorio en Azure Devops

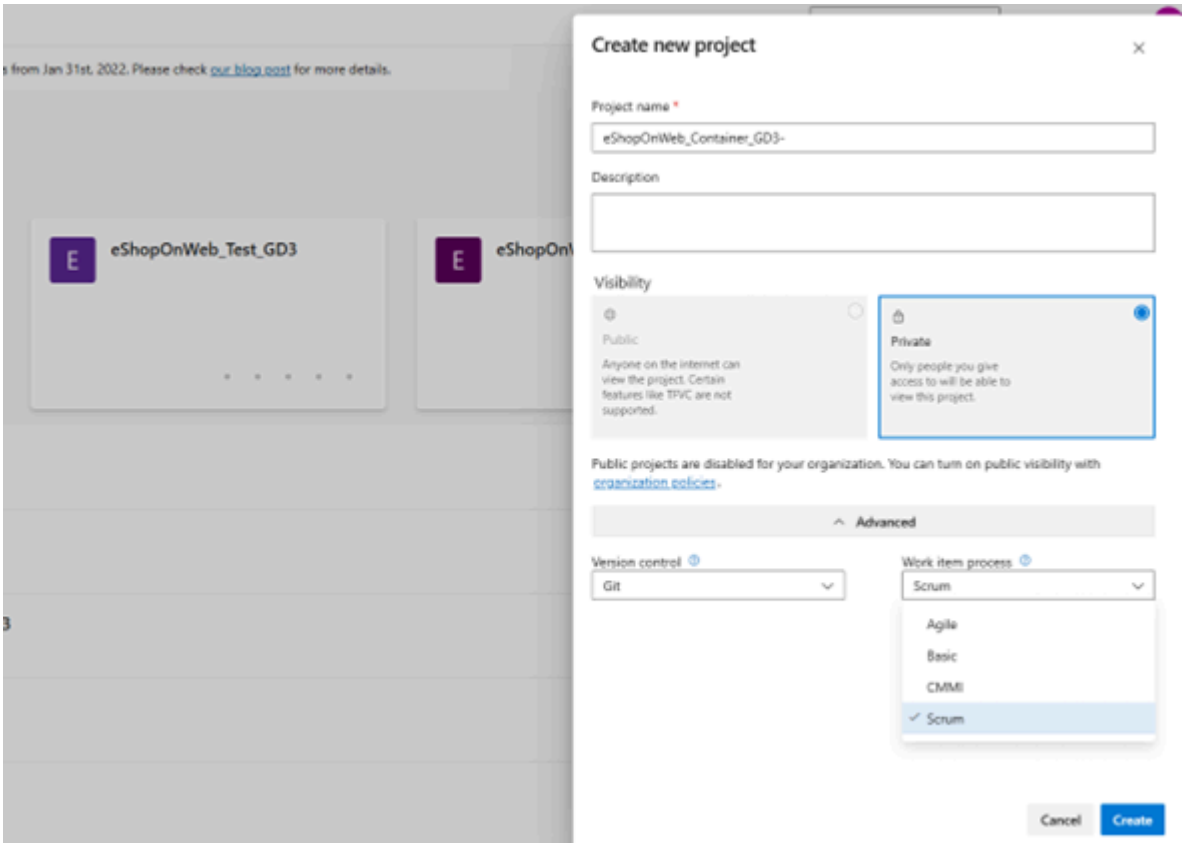
1. Dirigite a Azure Devops y crea un nuevo proyecto

2. Presiona New project

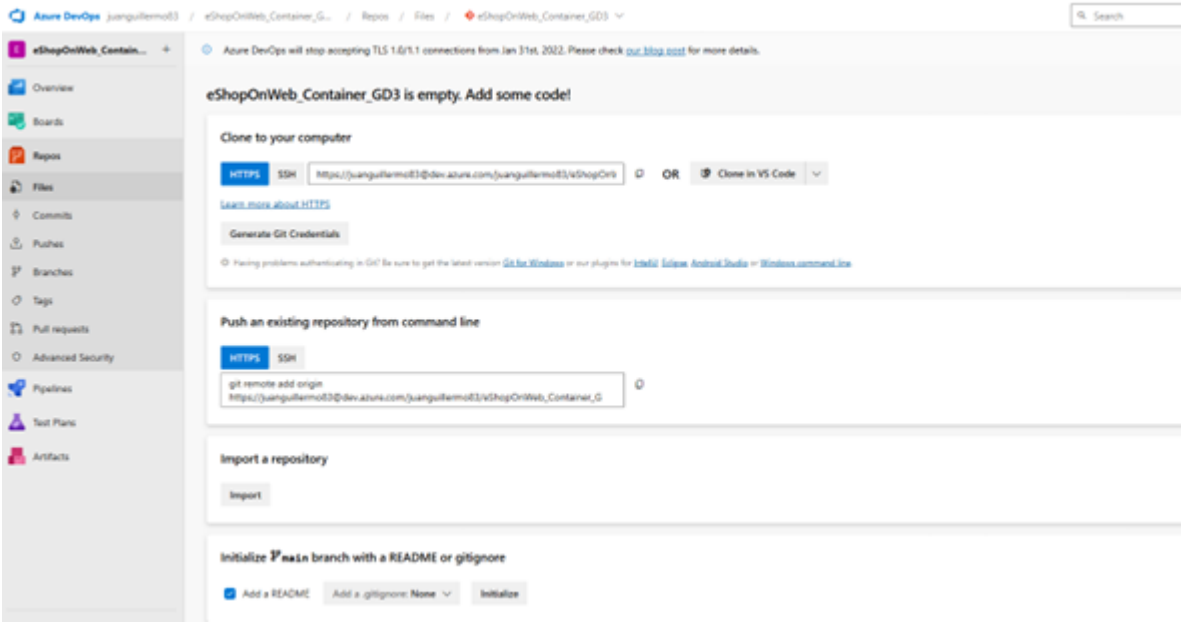


3. Completa el formulario de creación

    - Nombra el proyecto (eShopOnWeb_Container_GD3)
    - Visibility Privada
    - Avanzado: Git y scrum

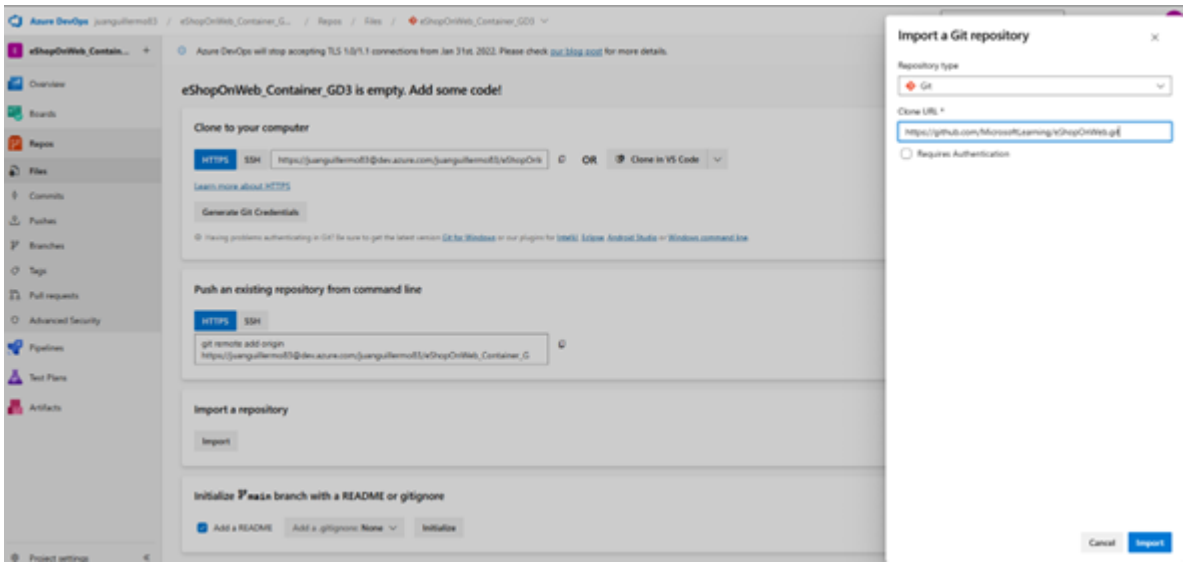


4. Importar

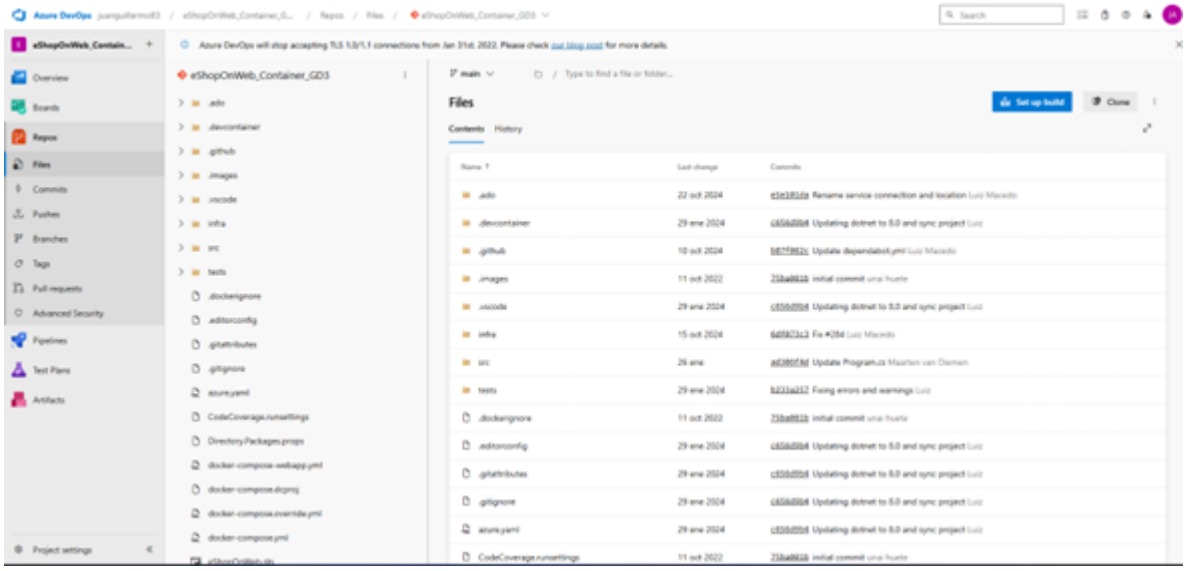


5. Clona el repositorio (usa la url entregada al inicio del documento)

   https://github.com/MicrosoftLearning/eShopOnWeb.git



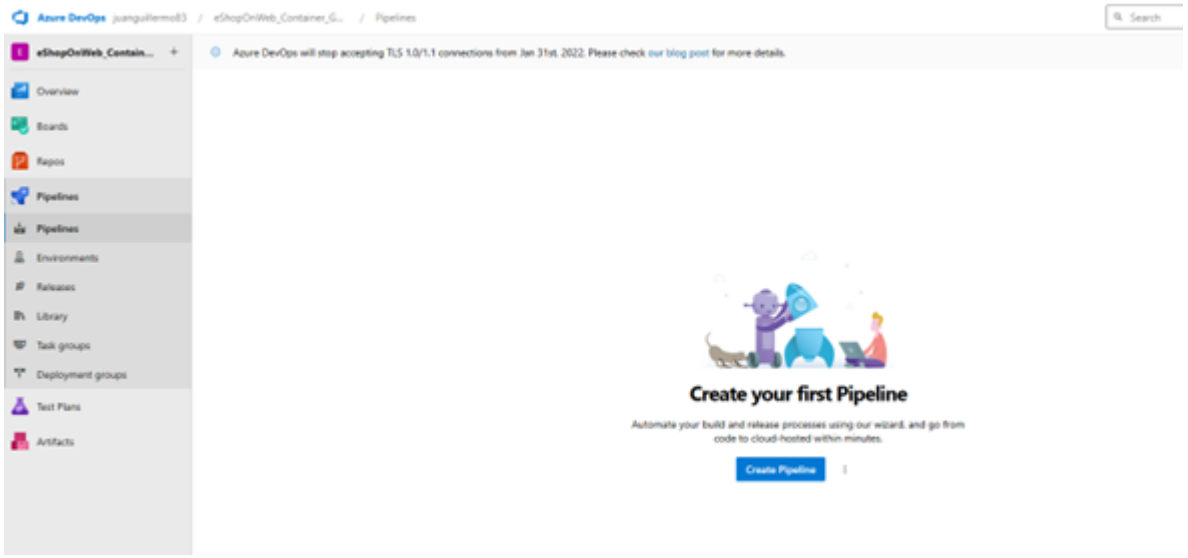
6. Repositorio Cargado



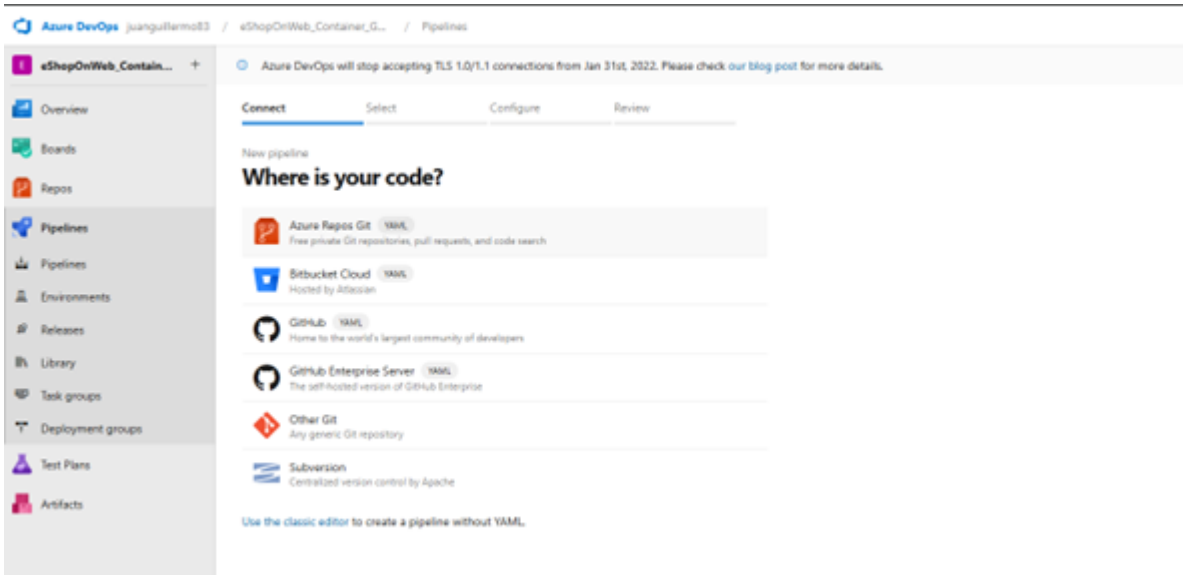
Crear Pipelines CI y CD

Importar y ejecutar el pipeline CI

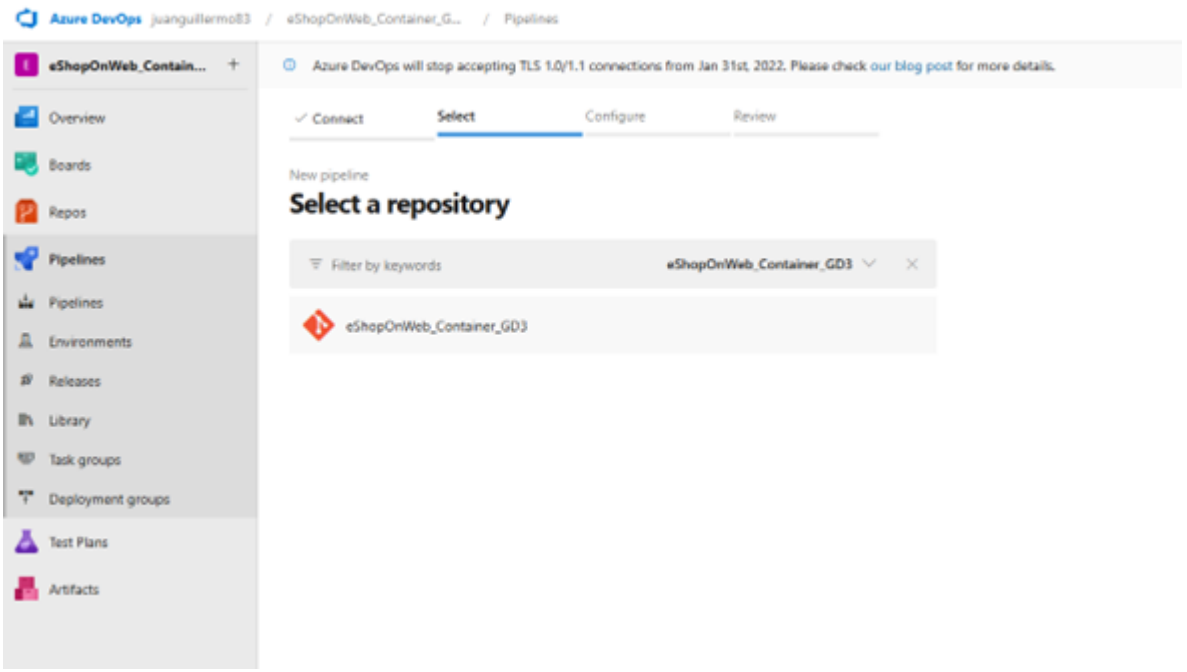
1. Pipelines -> pipelines



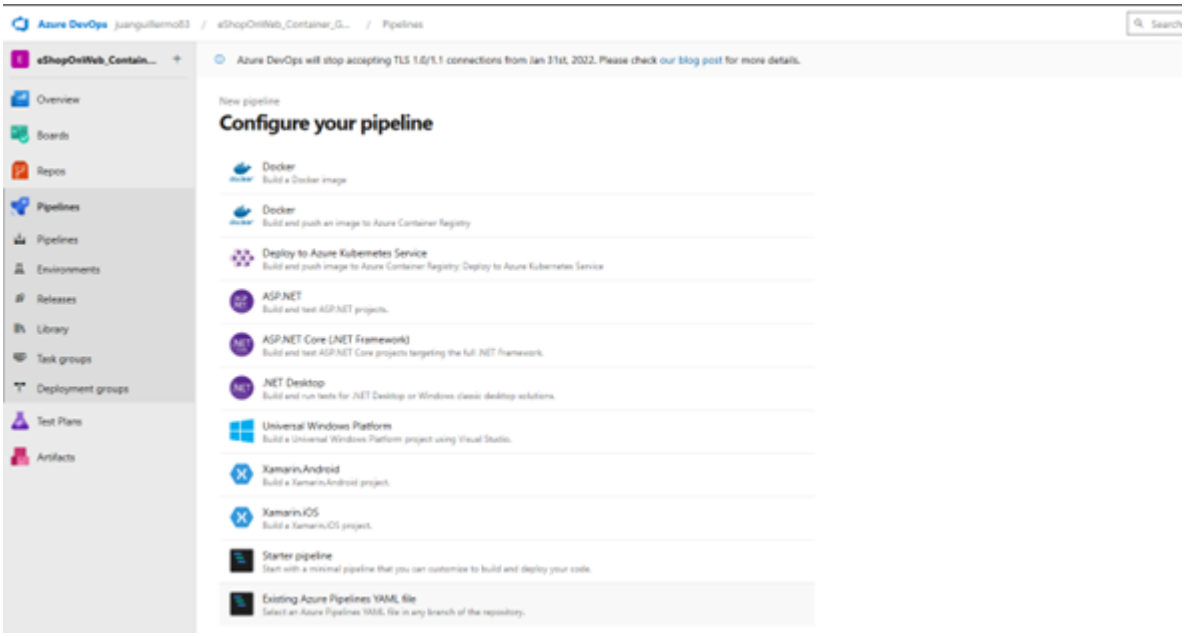
2. Azure Repos Git



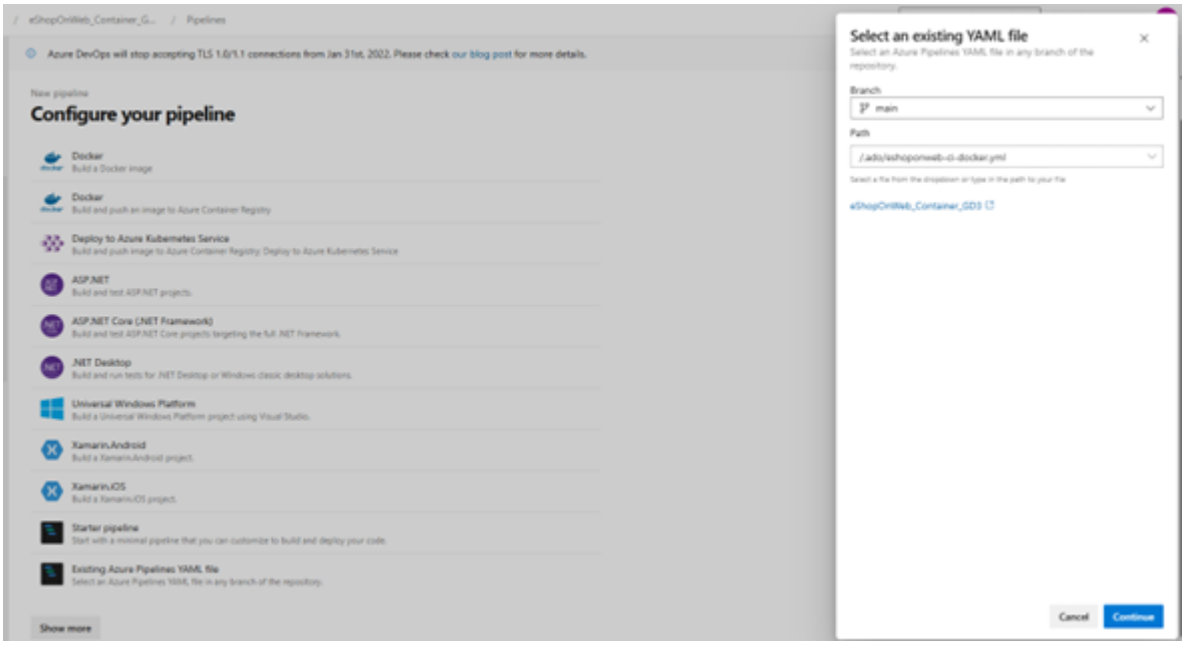
3. Selecciona el Repositorio



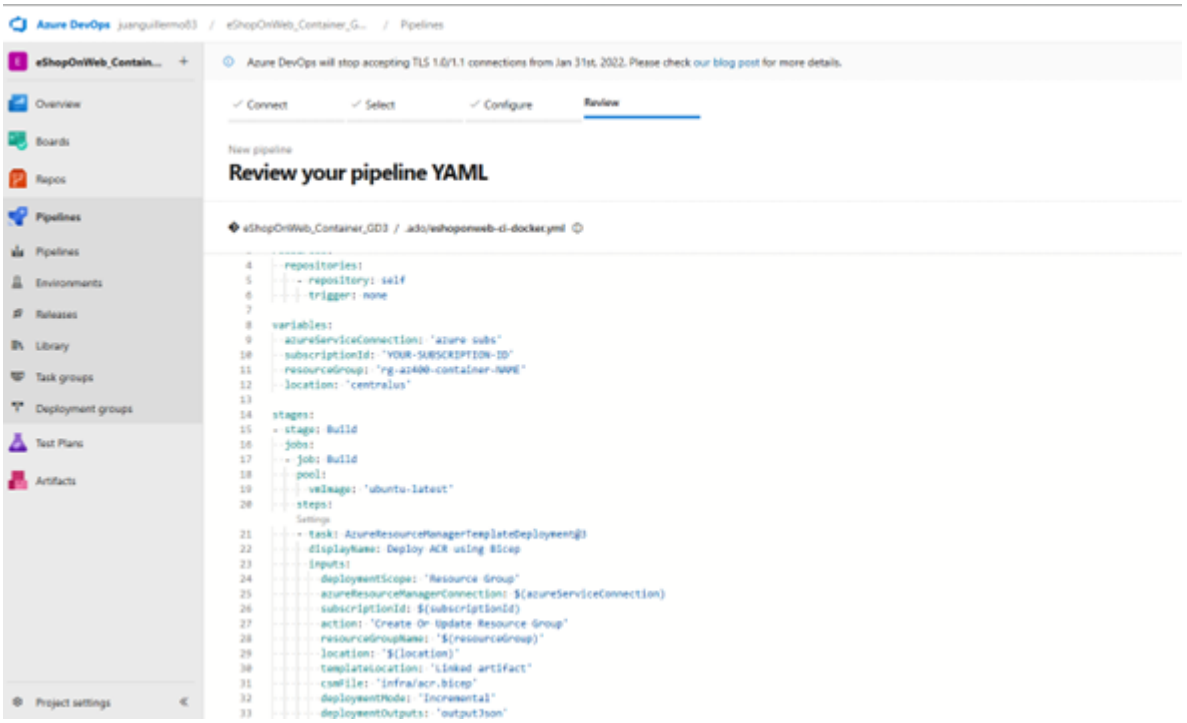
4. Opcion Existente azure pipeline YAML



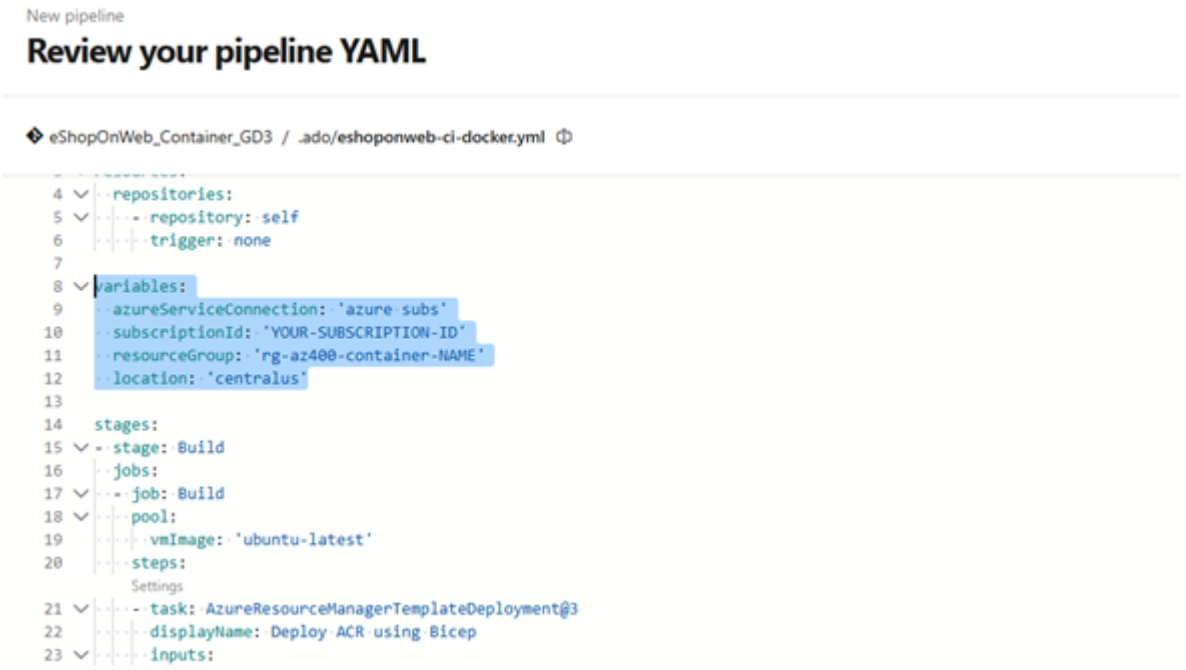
5. Selecciona eshoponweb-ci-docker.yml



6. Actual YAML cargado



7. Actualizar los valores de las Variables, segun nuestros datos de conexion y servicios

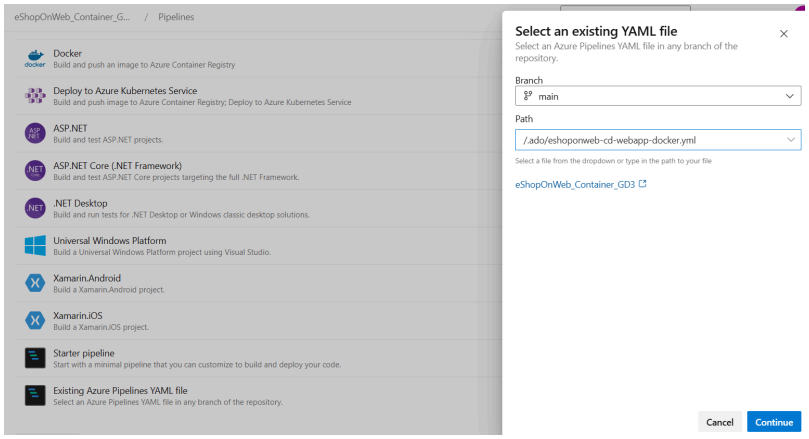


Importar y editar el pipeline CD Deploy

1. Repite los pasos 1 al 4 de la importacion de un pipeline ⬆️ Volver a la sección "Crear Pipelines CI y CD"

2. Selecciona Archivo YAML de Azure Pipelines existente

    - Seleccione la rama principal y el archivo /.ado/eshoponweb-cd-webapp-docker.yml

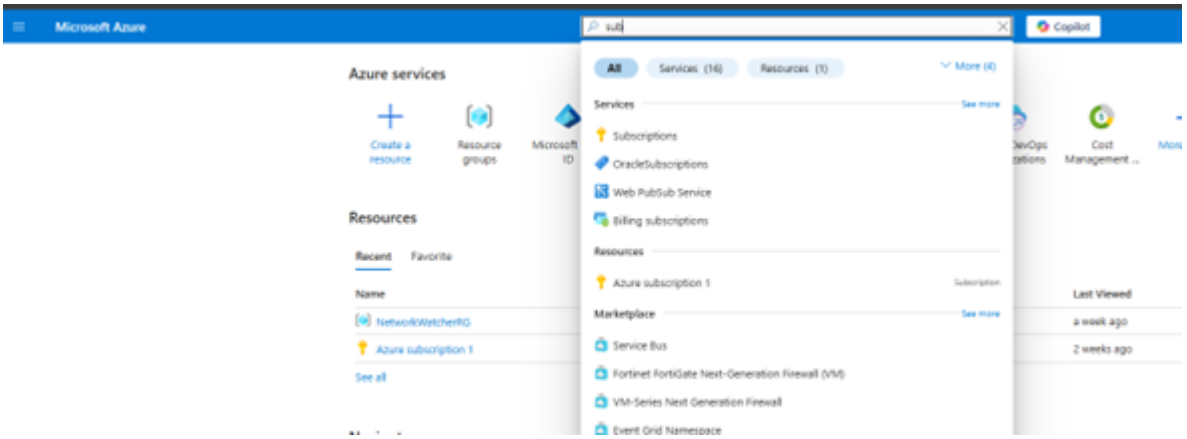


- presiona continuar

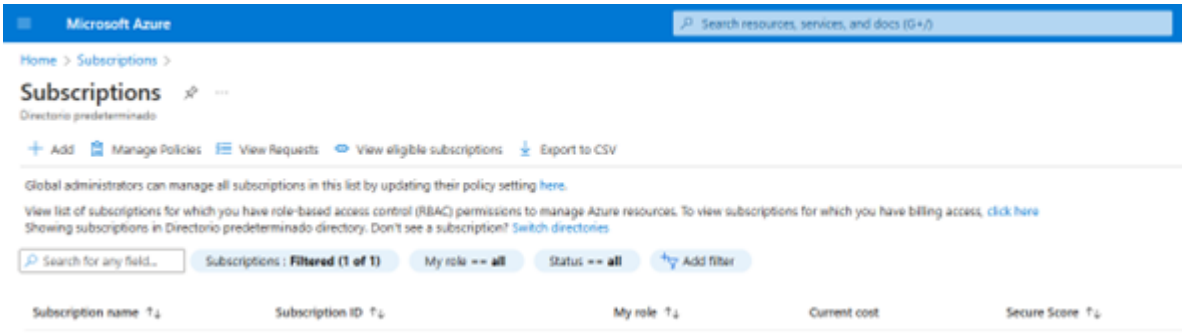
Editar Pipeline Con Valores de conexion

1.
```
azureServiceConnection: 'azure subs'
subscriptionId: 'xxxxx-xxxx-xxxx-xxxx-xxxxxx'
resourceGroup: 'rg-az400-container-NAME'
location: 'centralus'
```

2. Ir a azure Subscriptios



3. en la segunda columna se despliega el valor de ID de la suscripcion



4. Nombre del resourceGroup

    a. rg-az400-RG1

5. Location

    a. 'centralus'

Crear Service Principal y Service Connection

1. Obtener el nombre de la suscripcion

    ```
    subscriptionName=$(az account show --query name --output tsv)
    ```

2. Obtener el ID de la suscripcion

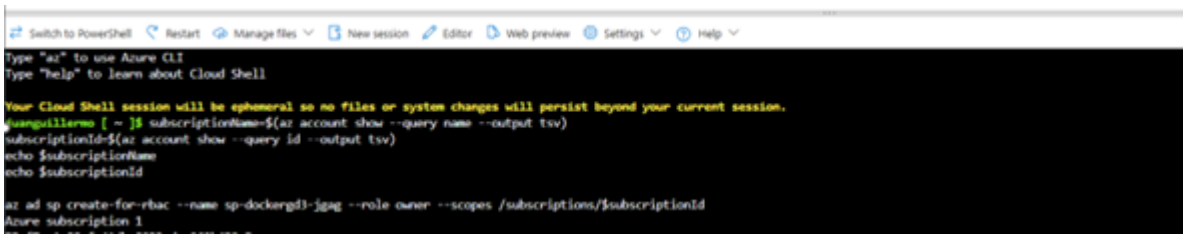
    ```
    subscriptionId=$(az account show --query id --output tsv)
    ```

3. Chequear que los valores cargaron en las variables

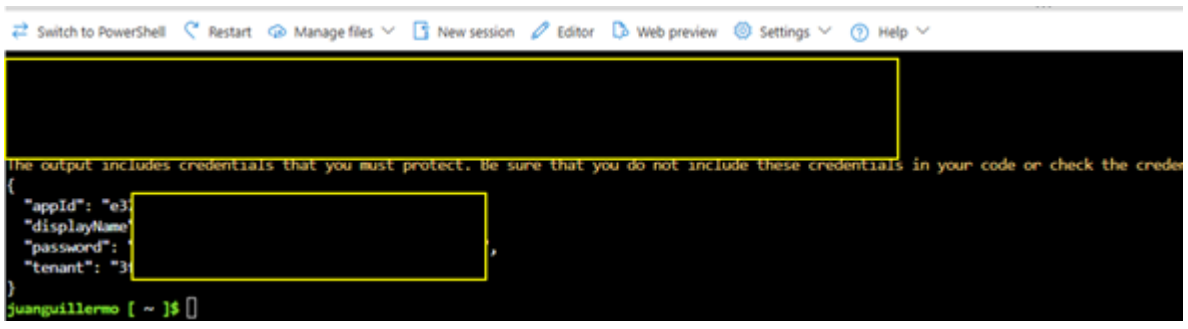
    ```
    echo $subscriptionName
    echo $subscriptionId
    ```

4. Crear un service principal

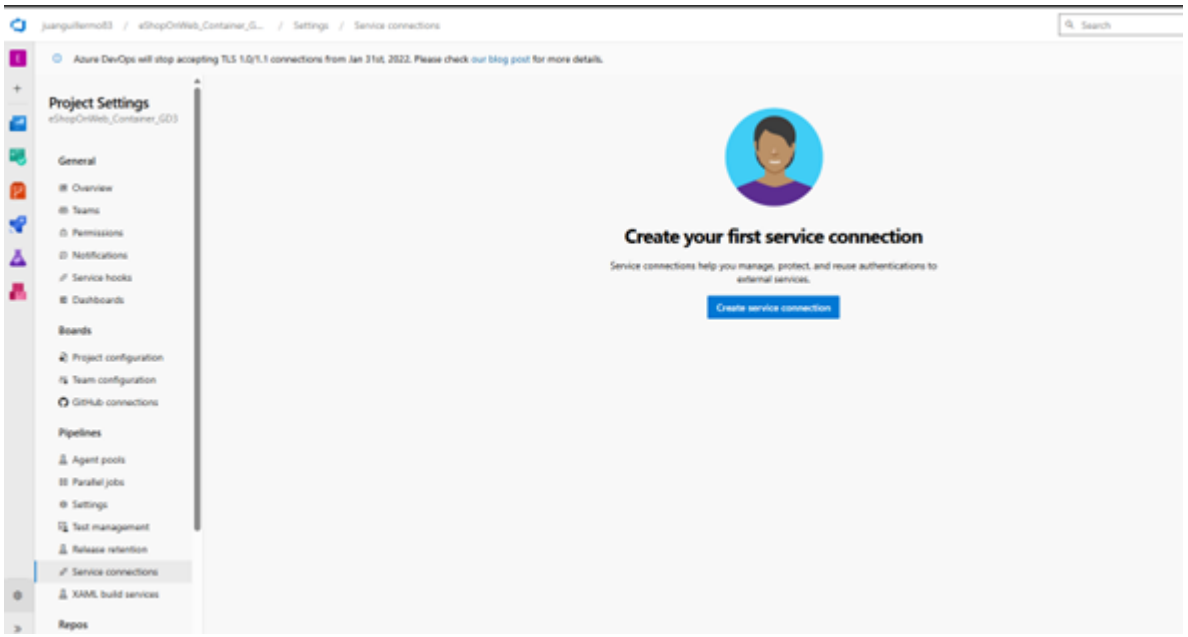
    ```
    az ad sp create-for-rbac --name sp-dockergd3-<XXX> --role owner --scopes /subscriptions/$subscriptionId
    ```



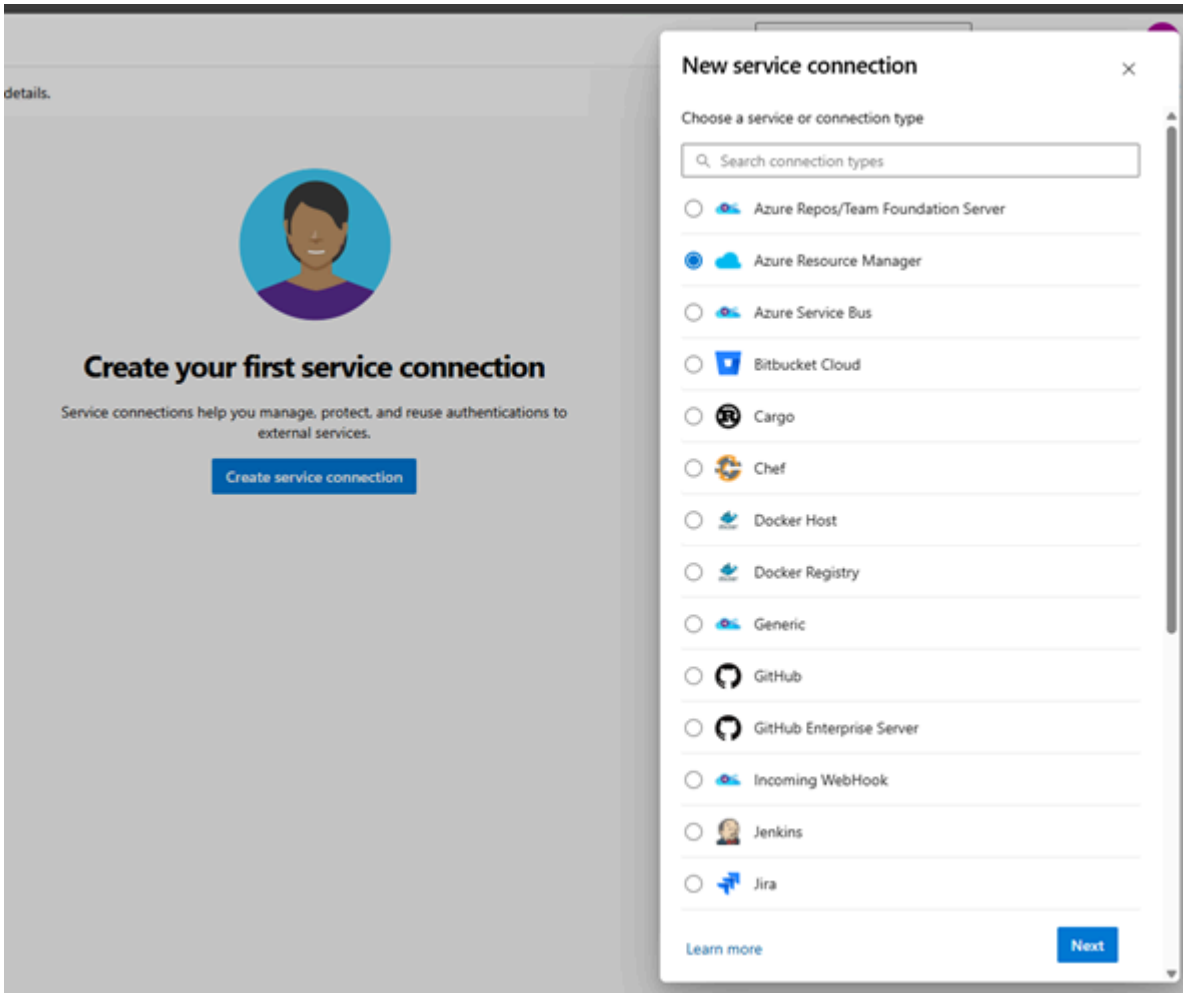
5. Guardar el resultado



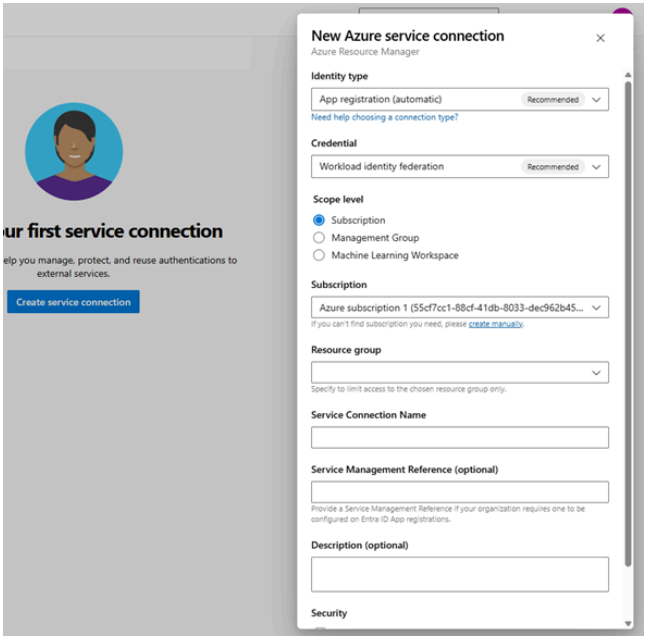
6. Ir a Azure devops para asociar el Service Principal con el Service Connection
    - project settings -> service connectios



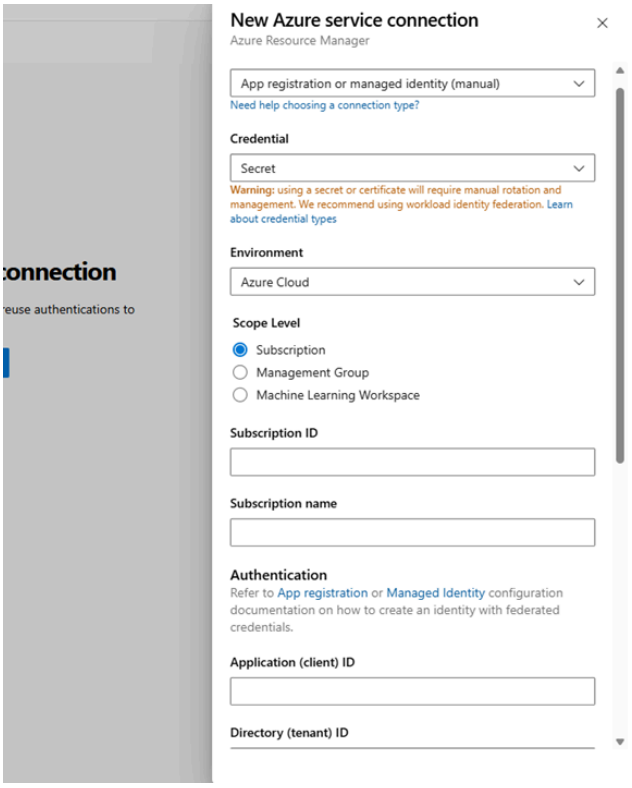
7. Crear service Connections en el menu debes agregar "Azure Resource Manager"



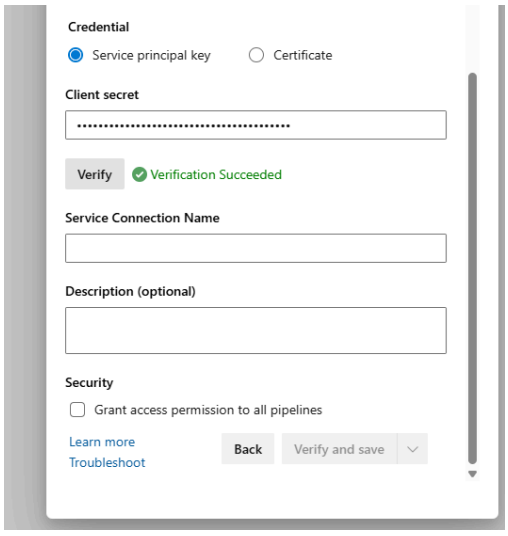
8. Presiona "NEXT", continua llenando el formulario



a. Identity type : App registration or managed identity (manual) b. credential : secret



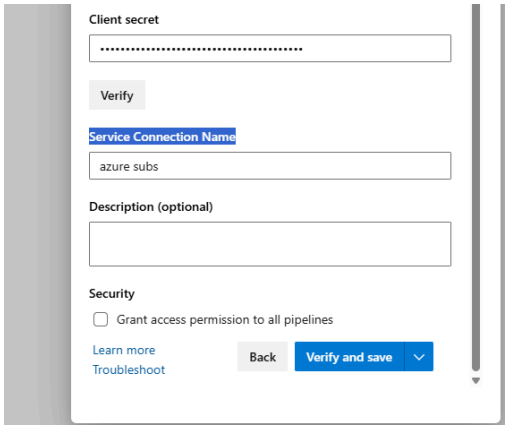
c. completa el formulario con los datos obtenido en la creacion del service principal d. Verificar



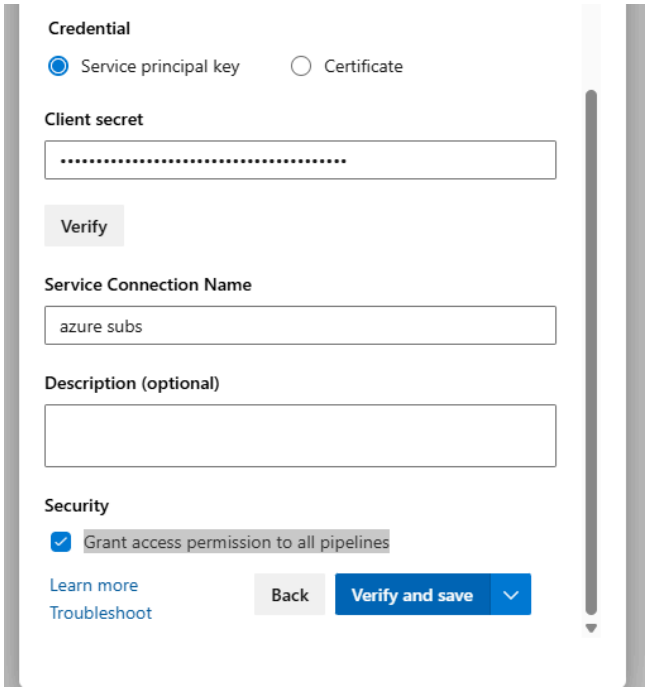
e. Service Connection Name

"azure subs" Este nombre debe ser el mismo del pipeline

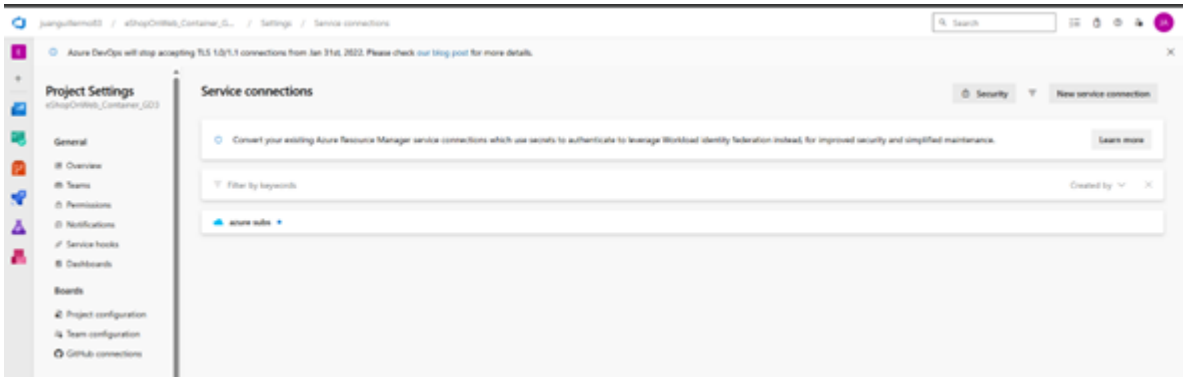
```
variables:
  azureServiceConnection: 'azure subs'
  resourceGroup: 'rg-az400-RG1'
  location: 'centralus'
```



f. Marcar Grant access permission to all pipelines

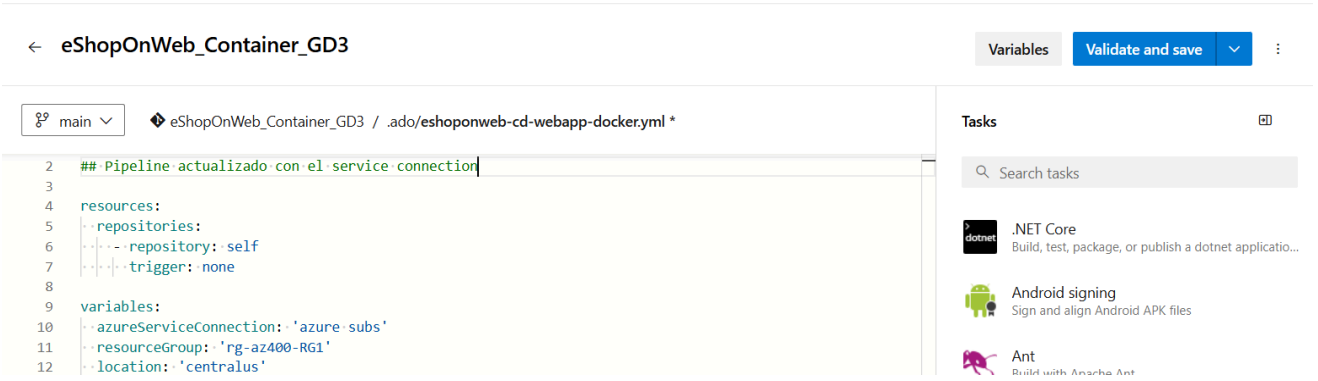


g. Service Connection ha sido creado

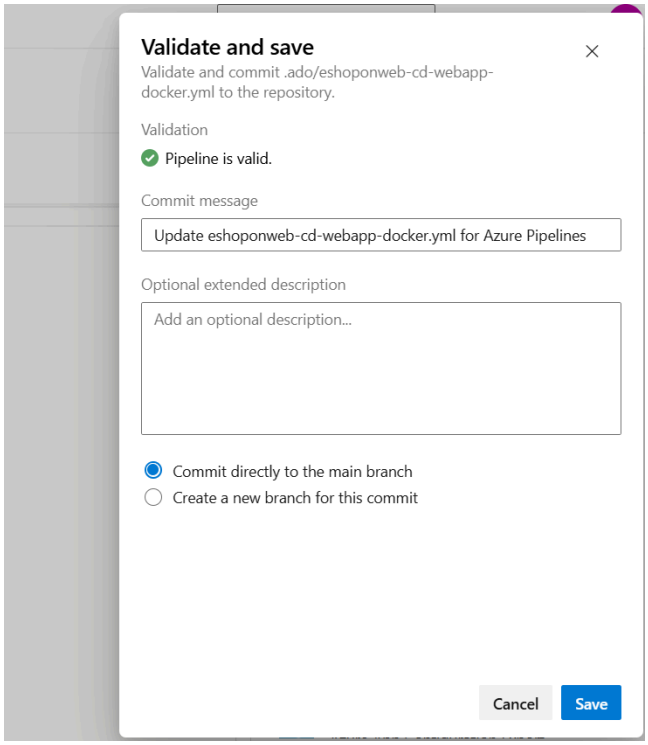


Actualizar y Ejecutar el Pipeline

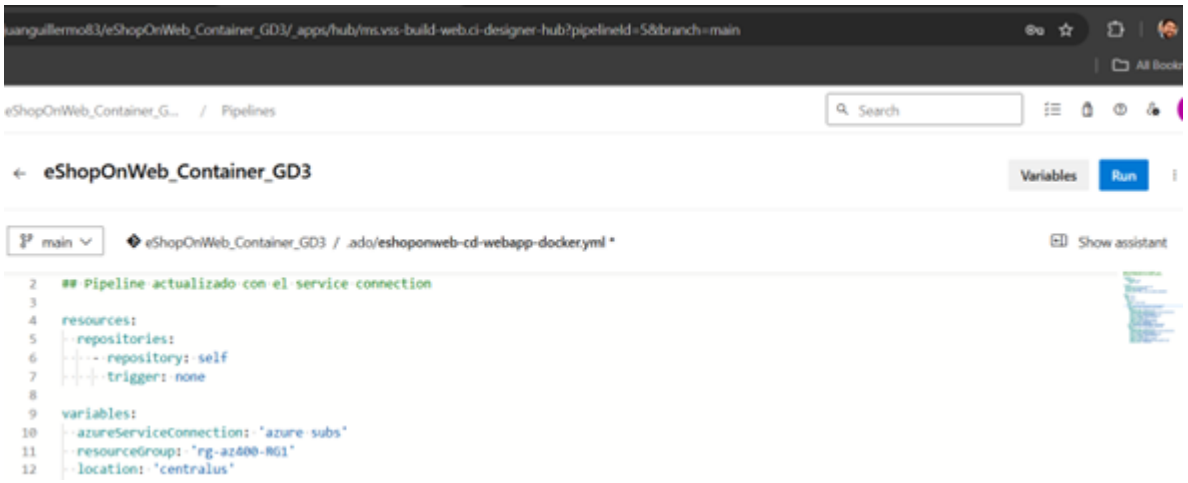
1. Actualizamos



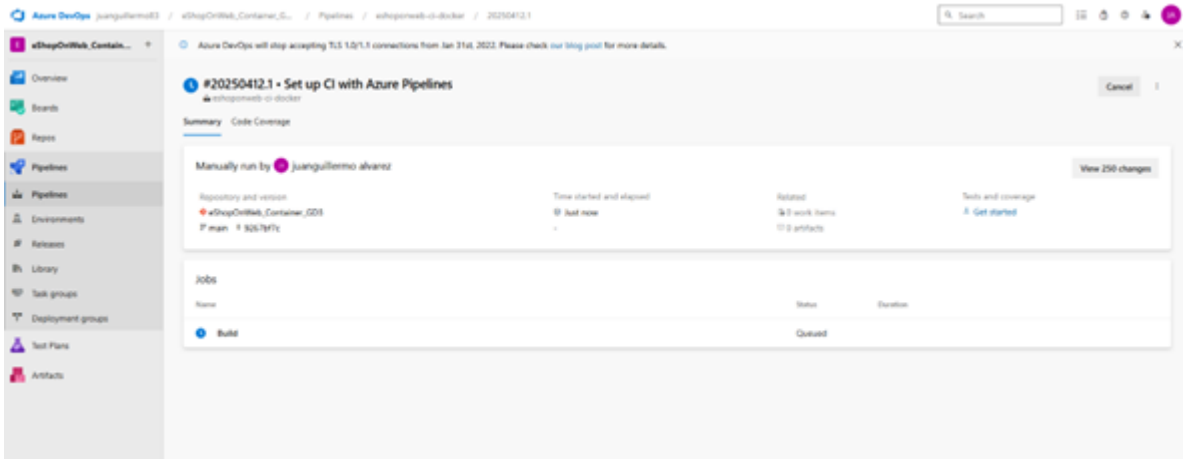
2. Save and validate



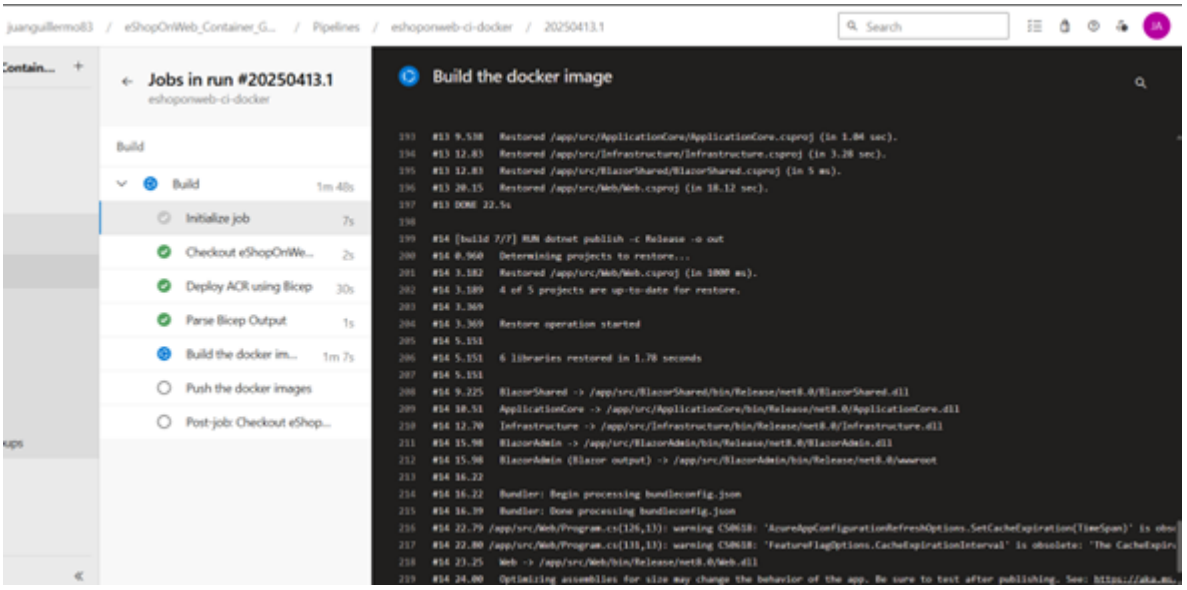
3. Ejecutar el pipeline en RUN



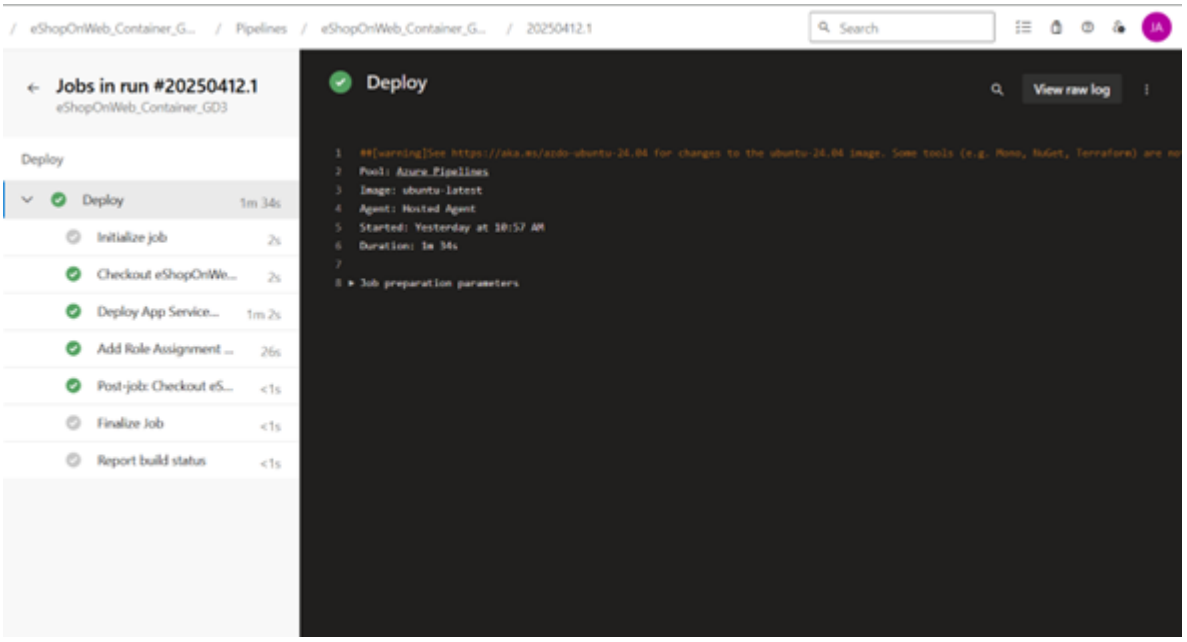
4. ejecucion



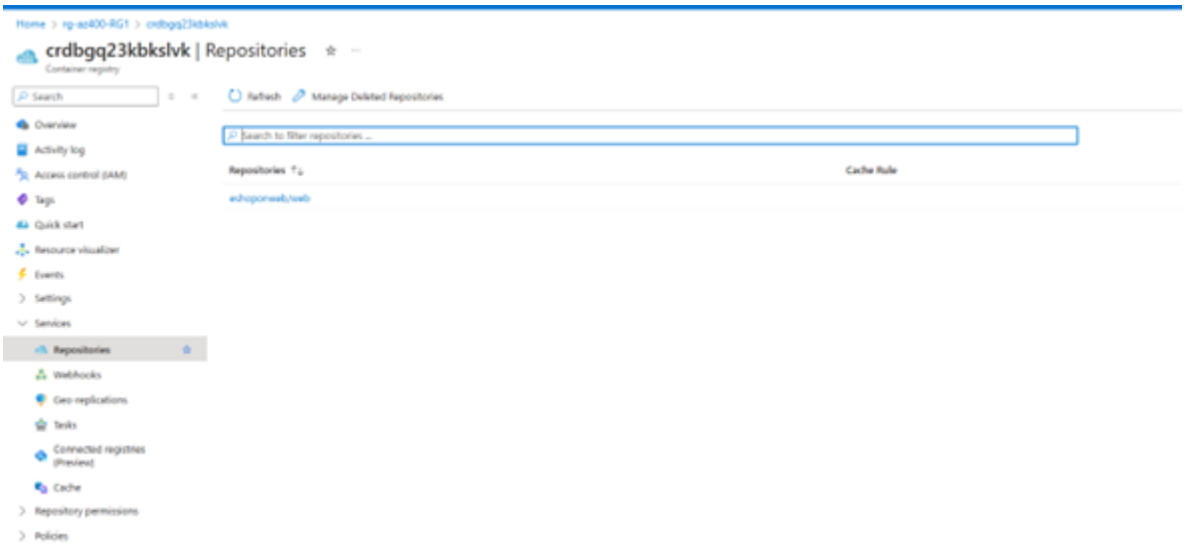
5. Build CI



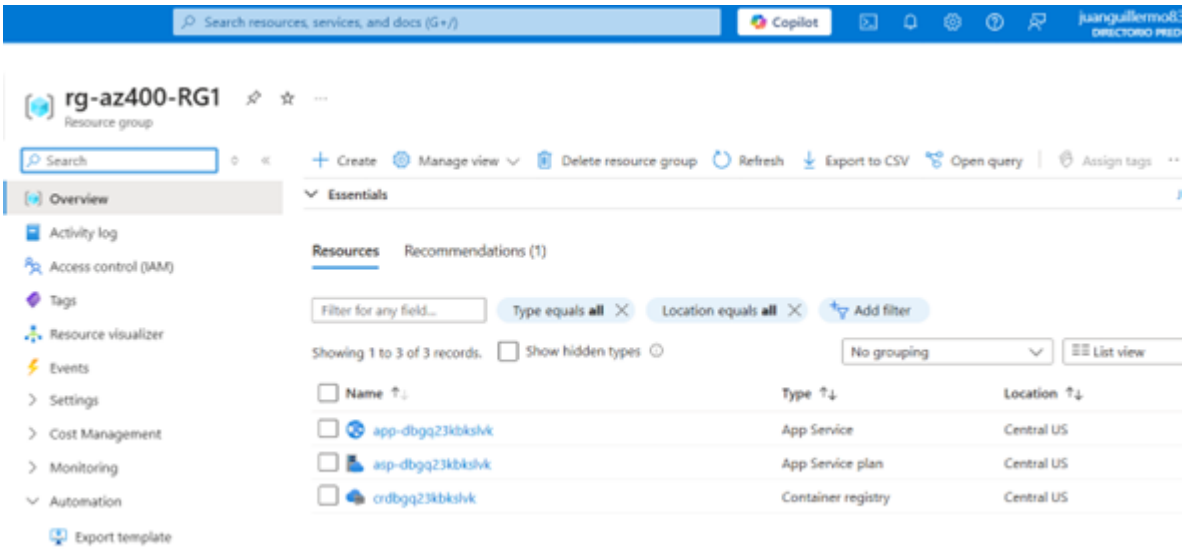
6. Deploy



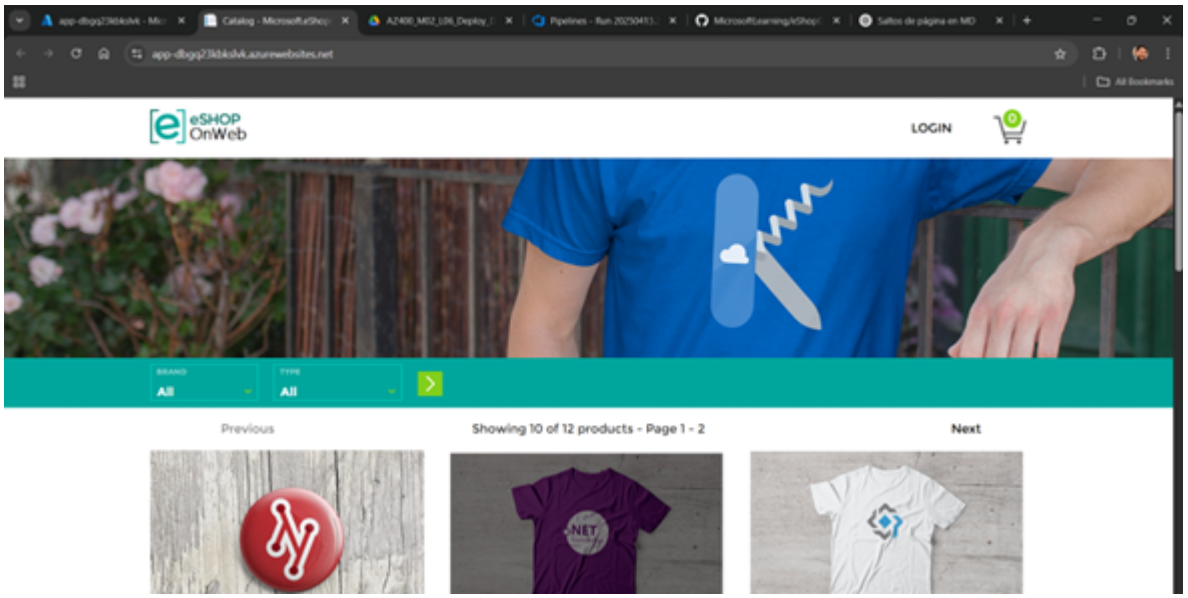
7. Verificar los recursos



8. ingresar al app service



9. ir al browser



Juan Guillermo Álvarez
Ingeniero DevSecOps
Valparaíso, Chile – Abril 2025
linkedin.com/in/juan-guillermo-alvarez-gallardo