

- 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
 - Reflexión sobre el proceso de ejecución

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

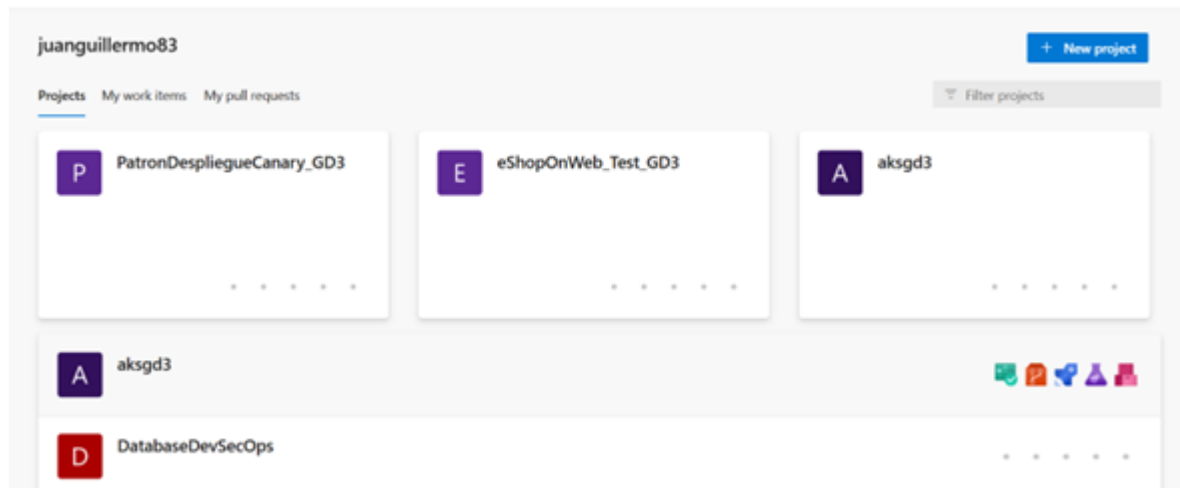
Objetivos

- 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

Create new project

Project name *
eShopOnWeb_Container_GD3-

Description

Visibility

Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.

Private
Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

Advanced

Version control
Git

Work item process
Scrum

Agile
Basic
CMMI
✓ Scrum

Cancel Create

4. Importar

Azure DevOps juanguillemo83 / eShopOnWeb_Container_G... / Repos / Files / eShopOnWeb_Container_GD3

eShopOnWeb_Container_GD3 is empty. Add some code!

Clone to your computer

HTTPS SSH `https://juanguillemo83@dev.azure.com/juanguillemo83/vShopOnWeb_Container_GD3` OR Clone in VS Code

Search more about HTTPS

Generate Git Credentials

Push an existing repository from command line

HTTPS SSH

git remote add origin `https://juanguillemo83@dev.azure.com/juanguillemo83/vShopOnWeb_Container_GD3`

Import a repository

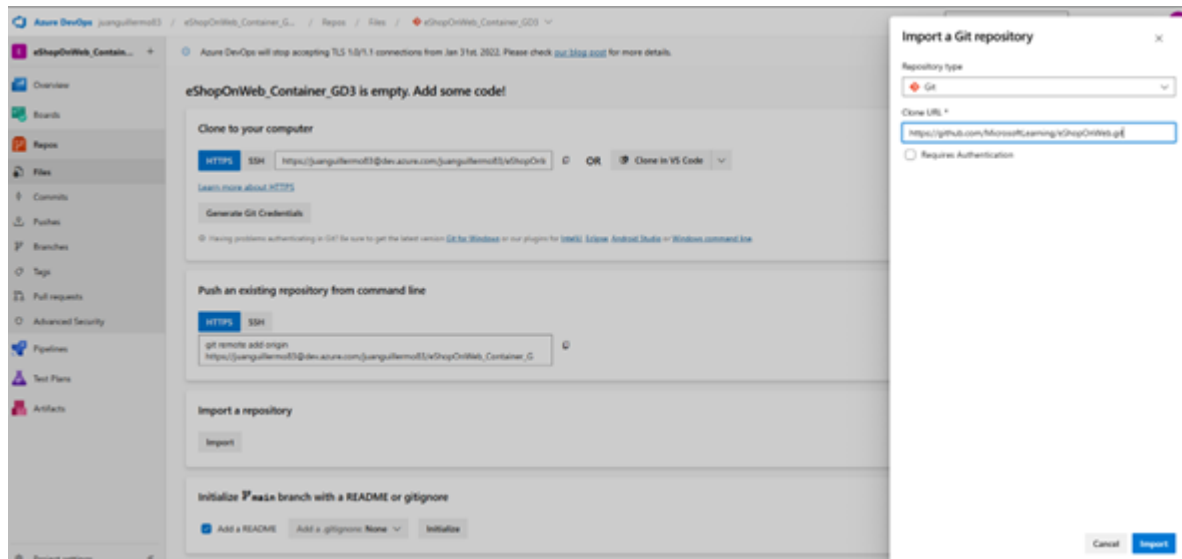
Import

Initialize main branch with a README or gitignore

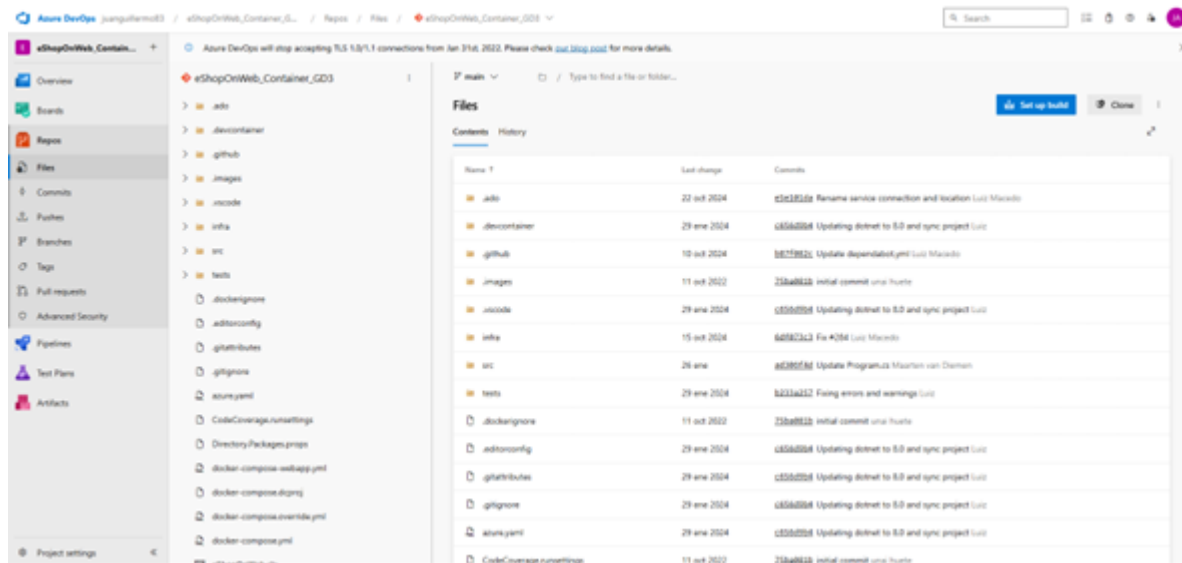
✓ Add a README Add a .gitignore: None Initialize

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

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



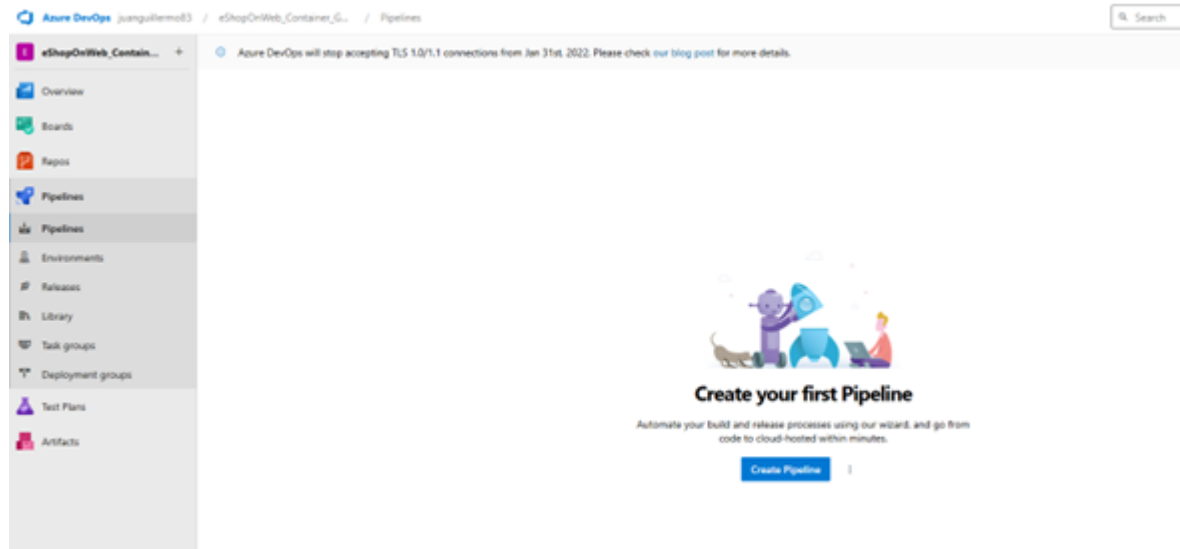
6. Repositorio Cargado



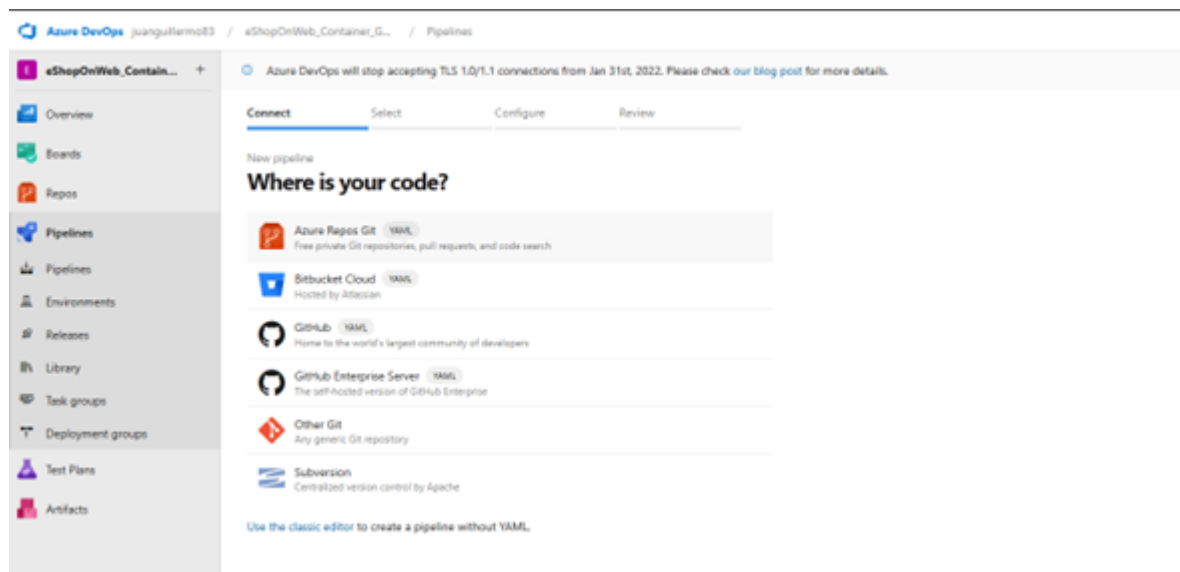
Crear Pipelines CI y CD

Importar y ejecutar el pipeline CI

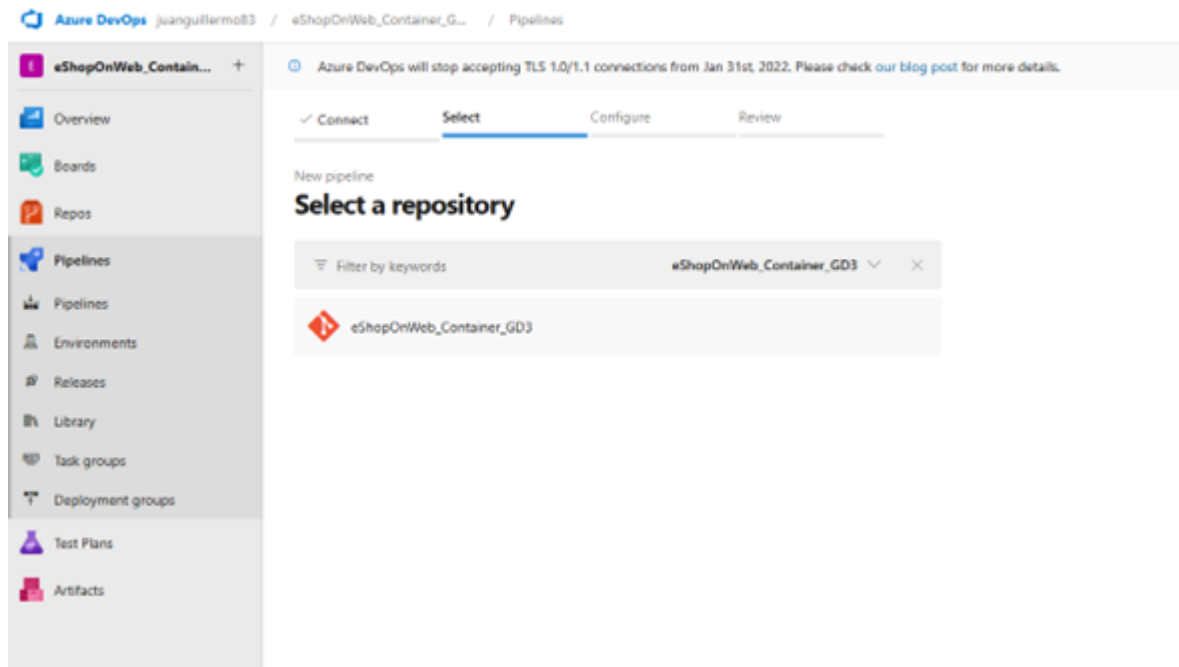
1. Pipelines -> pipelines



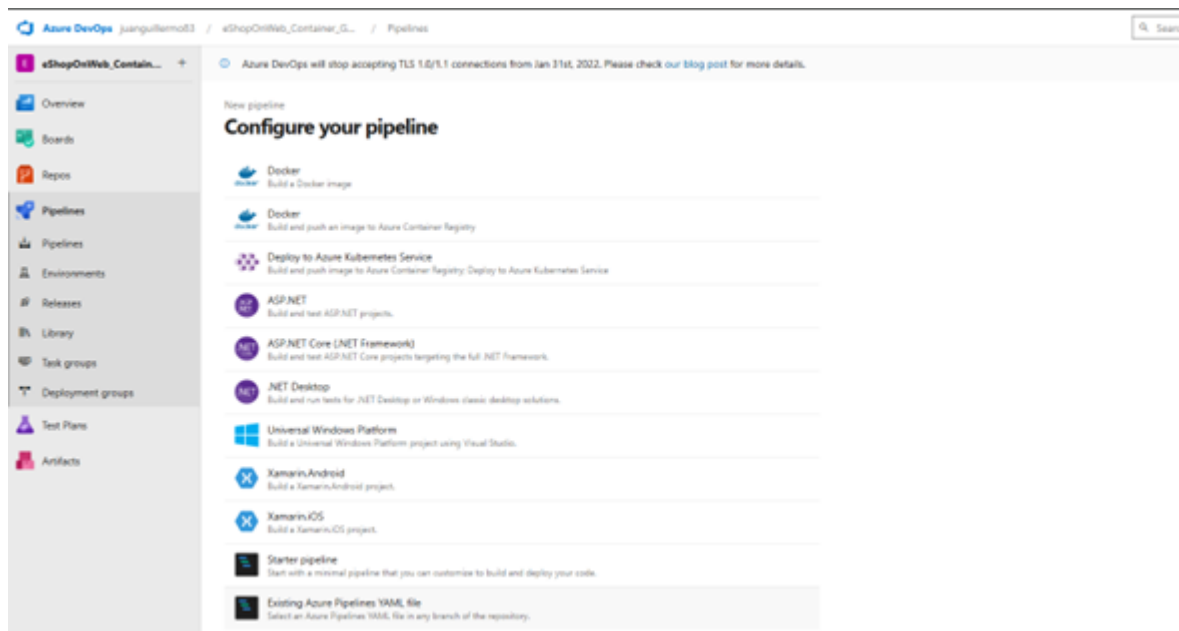
2. Azure Repos Git



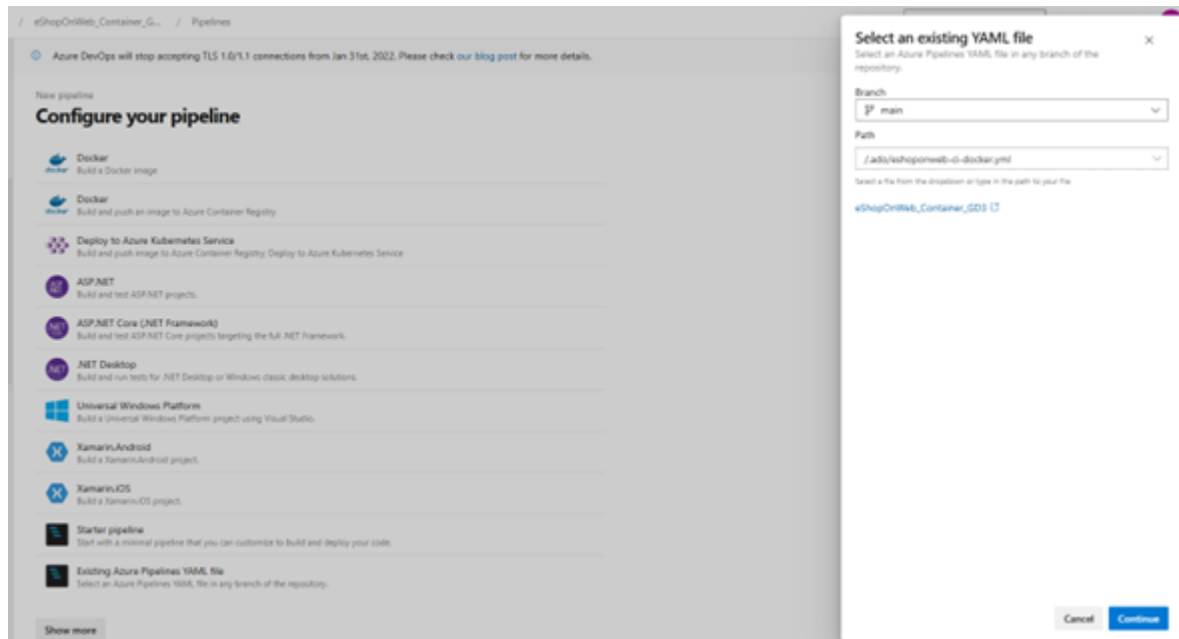
3. Selecciona el Repositorio



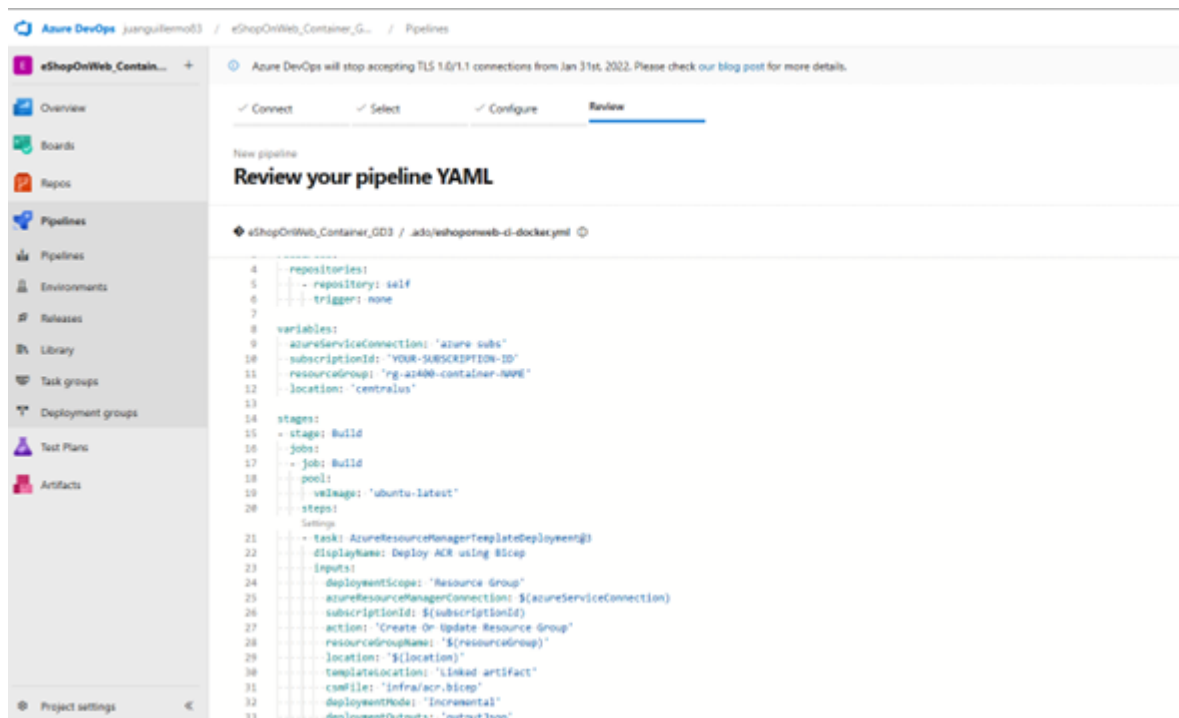
4. Opcion Existente azure pipeline YAML



5. Selecciona eshoponweb-ci-docker.yml



6. Actual YAML cargado



7. Actualizar los valores de las Variables, según nuestros datos de conexión y servicios

New pipeline

Review your pipeline YAML

📌 eShopOnWeb_Container_GD3 / .ado/eshoponweb-ci-docker.yml ⓘ

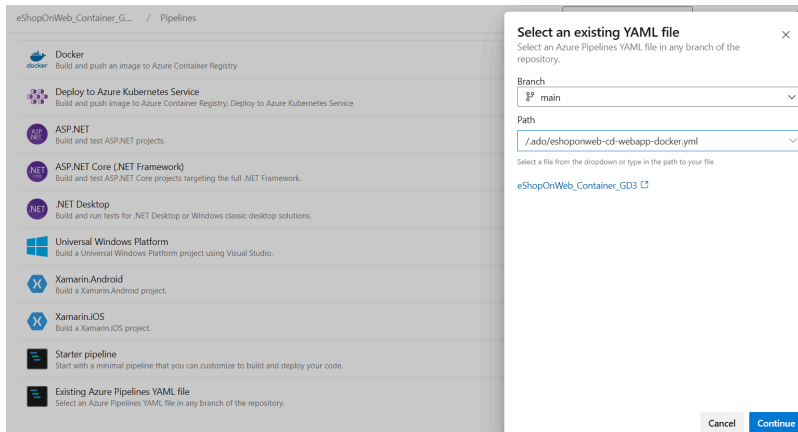
```
4  repositories:
5  - repository: self
6    trigger: none
7
8  variables:
9    azureServiceConnection: 'azure subs'
10   subscriptionId: 'YOUR-SUBSCRIPTION-ID'
11   resourceGroup: 'rg-az400-container-NAME'
12   location: 'centralus'
13
14  stages:
15  - stage: Build
16    jobs:
17    - job: Build
18      pool:
19        vmImage: 'ubuntu-latest'
20      steps:
21        Settings
22        - task: AzureResourceManagerTemplateDeployment@3
23          displayName: Deploy ACR using Bicep
24          inputs:
```

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`



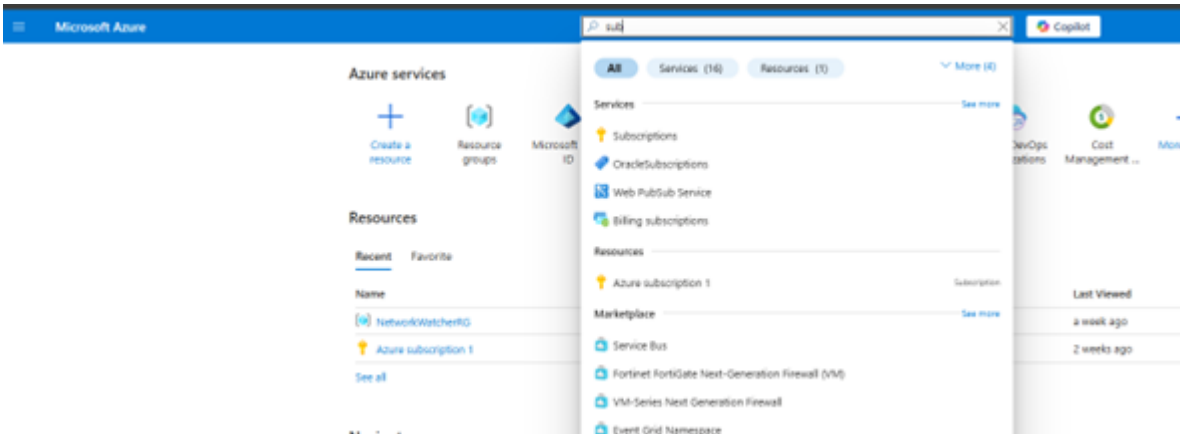
- presiona continuar

Editar Pipeline Con Valores de conexion

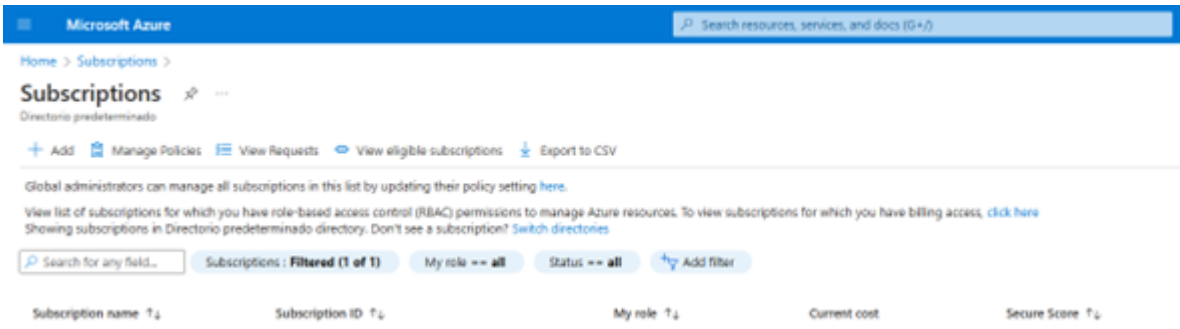
1.

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

2. Ir a azure Subscriptions



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

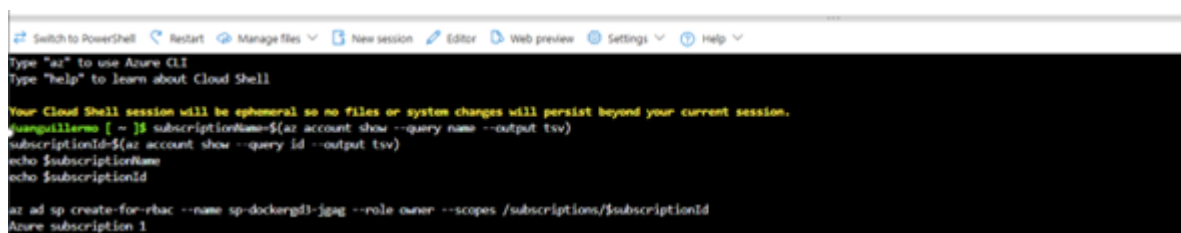
```
subscriptionId=$(az account show --query id --output tsv)
```

3. Chequear que los valores cargaron en las variables

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

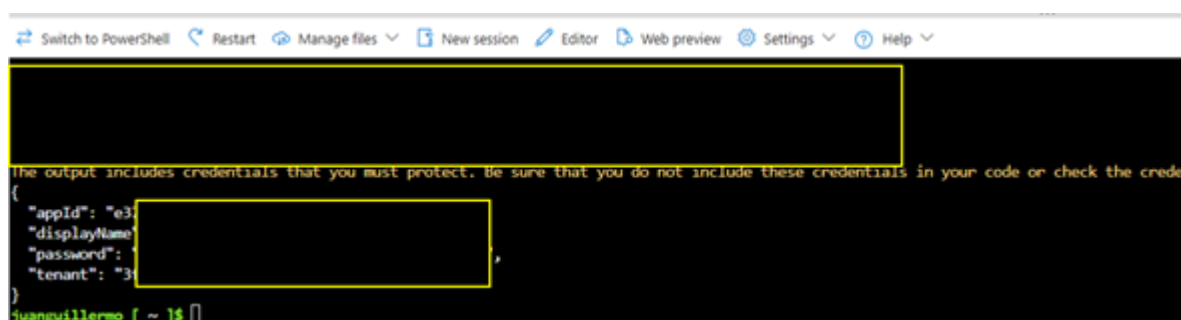
4. Crear un service principal

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



```
Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help  
Type "az" to use Azure CLI  
Type "help" to learn about Cloud Shell  
Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.  
juanguillermo [ ~ ]$ subscriptionName=$(az account show --query name --output tsv)  
subscriptionId=$(az account show --query id --output tsv)  
echo $subscriptionName  
echo $subscriptionId  
az ad sp create-for-rbac --name sp-dockergd3-jag --role owner --scopes /subscriptions/$subscriptionId  
Azure subscription 1
```

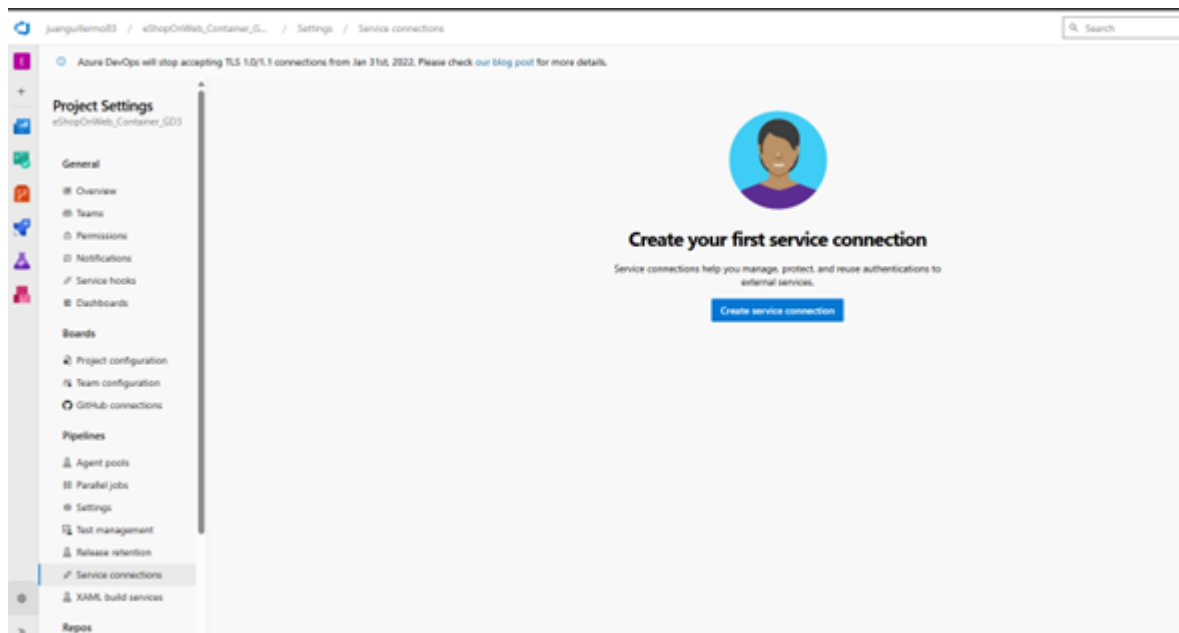
5. Guardar el resultado



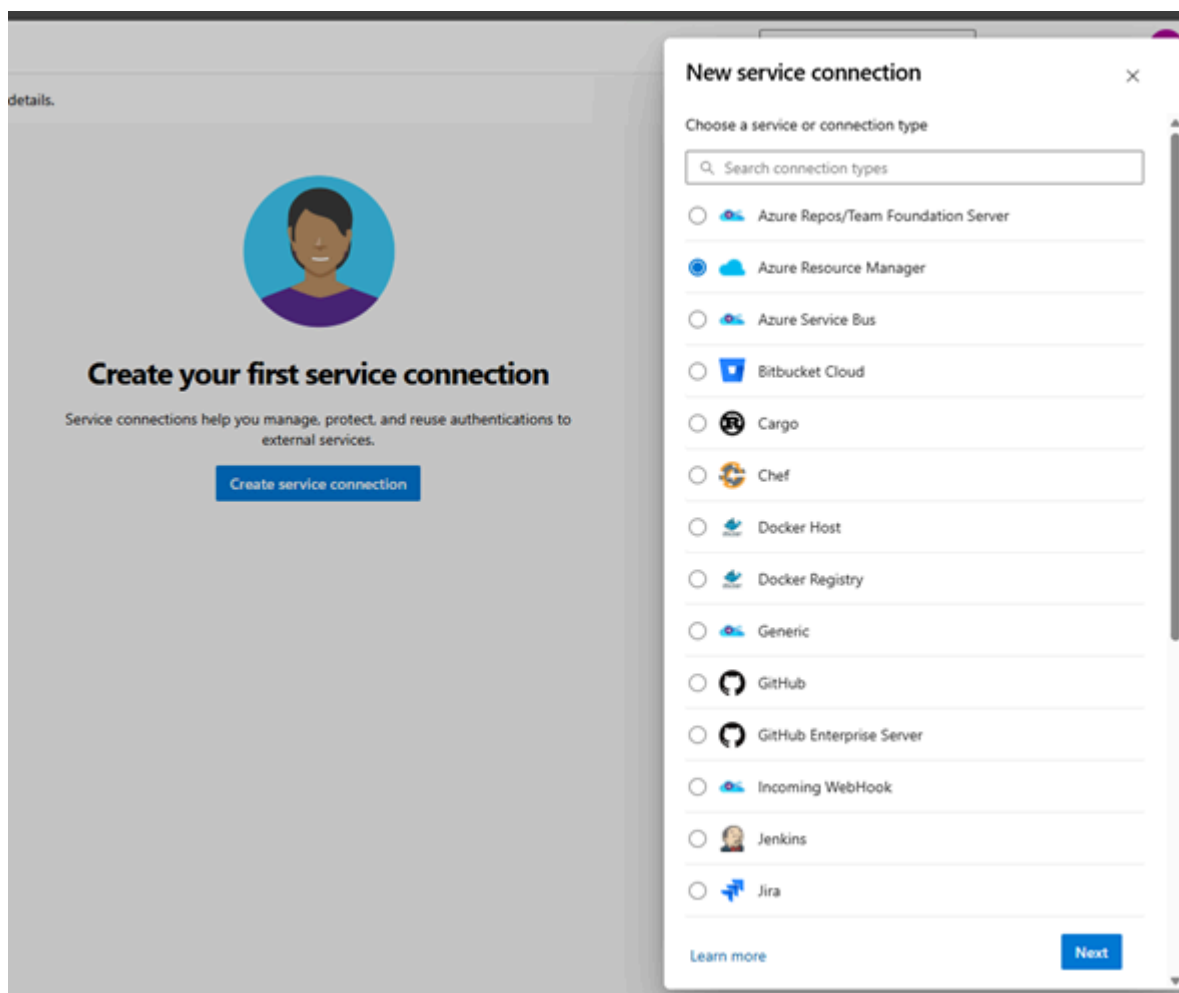
```
Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help  
The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or check the creden  
{  
  "appId": "e3  
  "displayName":  
  "password":  
  "tenant": "3  
}  
juanguillermo [ ~ ]$
```

6. Ir a Azure devops para asociar el Service Principal con el Service Connection

- project settings -> service connections



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



8. Presiona "NEXT", continua llenando el formulario

New Azure service connection
Azure Resource Manager

Identity type
App registration (automatic) Recommended

Credential
Workload identity federation Recommended

Scope level
☒ Subscription
☐ Management Group
☐ Machine Learning Workspace

Subscription
Azure subscription 1 (55cf7cc1-88cf-41db-8033-dec962b45...)

Resource group
Specify to limit access to the chosen resource group only.

Service Connection Name

Service Management Reference (optional)
Provide a Service Management Reference if your organization requires one to be configured on Entra ID App registrations.

Description (optional)

Security

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

New Azure service connection
Azure Resource Manager

App registration or managed identity (manual)

Credential
Secret

Environment
Azure Cloud

Scope Level
☒ Subscription
☐ Management Group
☐ Machine Learning Workspace

Subscription ID

Subscription name

Authentication
Warning: using a secret or certificate will require manual rotation and management. We recommend using workload identity federation. [Learn about credential types](#)

Application (client) ID

Directory (tenant) ID

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

Credential

☒ Service principal key ☐ Certificate

Client secret

.....

Verify ✔ Verification Succeeded

Service Connection Name

Description (optional)

Security

☐ Grant access permission to all pipelines

[Learn more](#)
[Troubleshoot](#)

Back **Verify and save** ▾

e. Service Connection Name

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

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

Client secret

.....

Verify

Service Connection Name

azure subs

Description (optional)

Security

☐ Grant access permission to all pipelines

[Learn more](#)
[Troubleshoot](#)

Back **Verify and save** ▾

f. Marcar Grant access permission to all pipelines

Credential

☒ Service principal key ☐ Certificate

Client secret

.....

Verify

Service Connection Name

azure subs

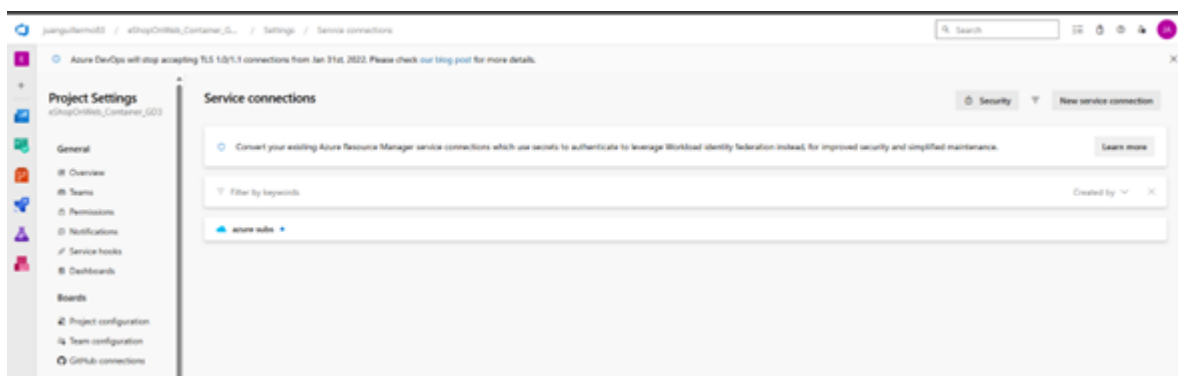
Description (optional)

Security

☒ Grant access permission to all pipelines

[Learn more](#) [Troubleshoot](#) [Back](#) [Verify and save](#)

g. Service Connection ha sido creado



Actualizar y Ejecutar el Pipeline

1. Actualizamos

← eShopOnWeb_Container_GD3

Variables Validate and save


main

eShopOnWeb_Container_GD3 / .ado/eshoponweb-cd-webapp-docker.yml *


```
2 ## Pipeline actualizado con el service connection
3
4 resources:
5   - repositories:
6     - repository: self
7       trigger: none
8
9   variables:
10    - azureServiceConnection: 'azure subs'
11    - resourceGroup: 'rg-az400-RG1'
12    - location: 'centralus'
```

Tasks


Search tasks

 .NET Core

Build, test, package, or publish a dotnet applicatio...

 Android signing

Sign and align Android APK files

 Ant

Build with Apache Ant

2. Save and validate

Validate and save

×

Validate and commit .ado/eshoponweb-cd-webapp-docker.yml to the repository.

Validation

✓

 Pipeline is valid.

Commit message

Update eshoponweb-cd-webapp-docker.yml for Azure Pipelines

Optional extended description

Add an optional description...

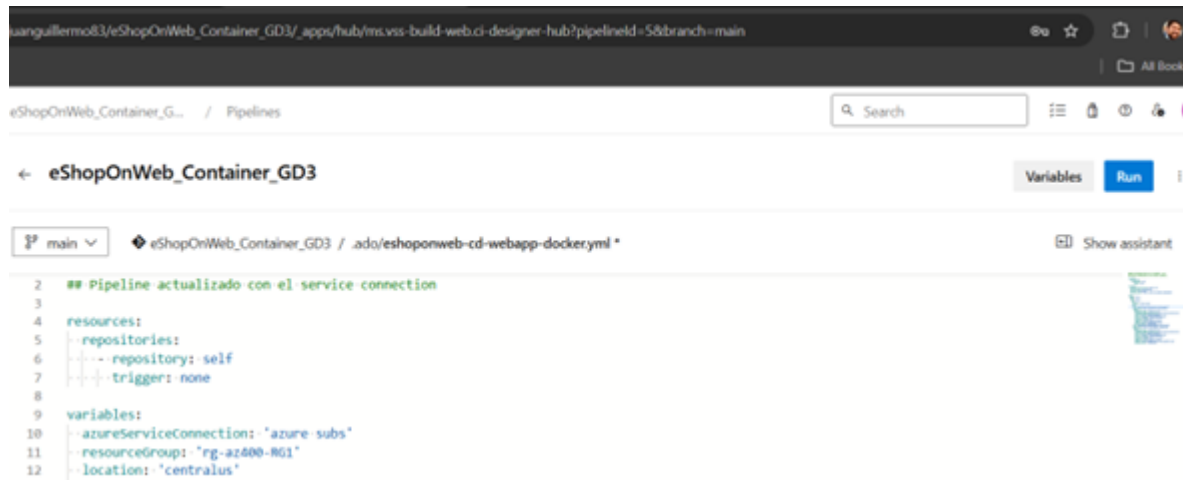
☒ Commit directly to the main branch

☐ Create a new branch for this commit

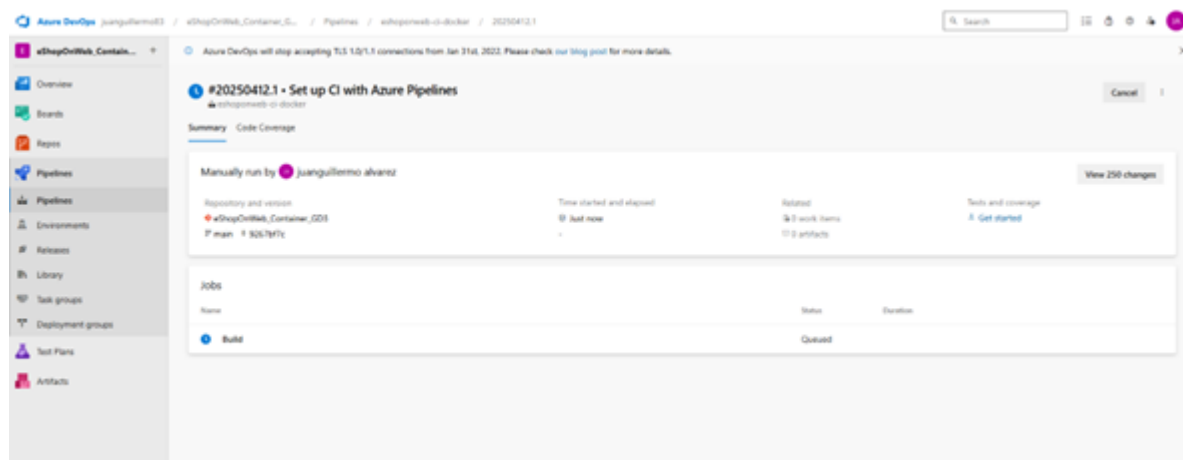
Cancel

Save

3. Ejecutar el pipeline en RUN



4. ejecucion



5. Build CI

The screenshot shows the 'Build the docker image' job in the 'eshoponweb-ci-docker' pipeline. The job is currently running. The left sidebar shows the job steps: Initialize job (7s), Checkout eShopOnWe... (2s), Deploy ACR using Bicep (30s), Parse Bicep Output (1s), Build the docker im... (1m 7s), Push the docker images, and Post-job: Checkout eShop... The main panel displays the build log, which includes commands like 'dotnet publish' and 'docker build'.

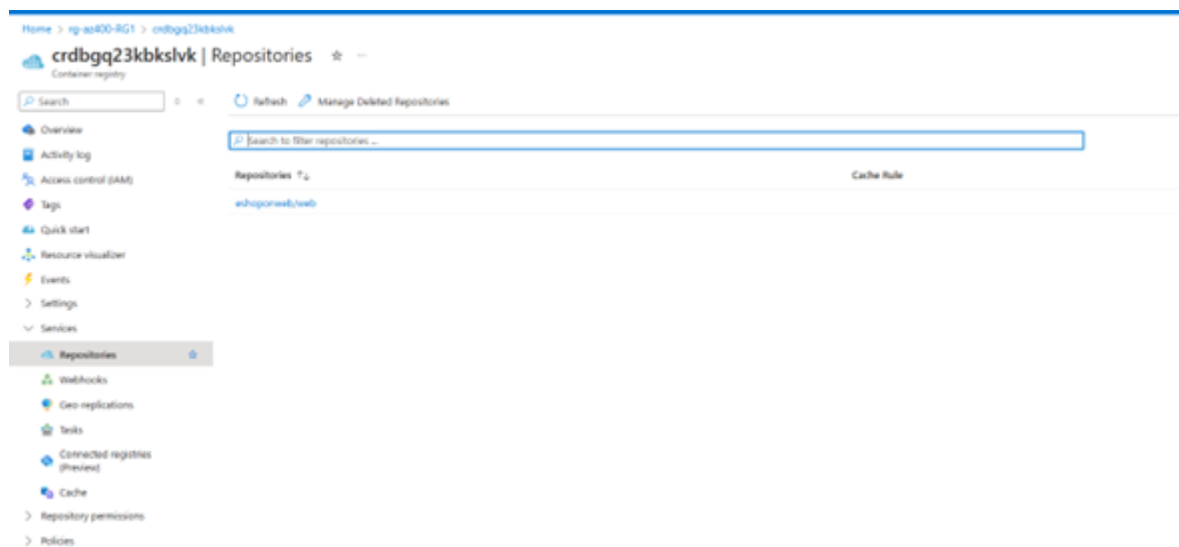
```
153 #13 9.538 Restored /app/src/ApplicationCore/ApplicationCore.csproj (in 1.84 sec).
154 #13 12.83 Restored /app/src/Infrastructure/Infrastructure.csproj (in 3.28 sec).
155 #13 12.83 Restored /app/src/BlazorShared/BlazorShared.csproj (in 5 ms).
156 #13 20.15 Restored /app/src/Web/Web.csproj (in 18.12 sec).
157 #13 DONE 22.5s
158
159 #14 [build 7/7] RM dotnet publish -c Release -o out
160 #14 0.960 Determining projects to restore...
161 #14 3.182 Restored /app/src/Web/Web.csproj (in 1000 ms).
162 #14 3.189 4 of 5 projects are up-to-date for restore.
163 #14 3.369 Restore operation started
164 #14 3.369 Restore operation started
165 #14 5.151 6 libraries restored in 1.78 seconds
166 #14 5.151
167 #14 9.225 BlazorShared -> /app/src/BlazorShared/bin/Release/net8.0/BlazorShared.dll
168 #14 10.51 ApplicationCore -> /app/src/ApplicationCore/bin/Release/net8.0/ApplicationCore.dll
169 #14 12.70 Infrastructure -> /app/src/Infrastructure/bin/Release/net8.0/Infrastructure.dll
170 #14 15.98 BlazorAdmin -> /app/src/BlazorAdmin/bin/Release/net8.0/BlazorAdmin.dll
171 #14 15.98 BlazorAdmin (Blazor output) -> /app/src/BlazorAdmin/bin/Release/net8.0/wwwroot
172 #14 16.22
173 #14 16.22 Bundler: Begin processing bundlerconfig.json
174 #14 16.39 Bundler: Done processing bundlerconfig.json
175 #14 22.79 /app/src/Web/Program.cs(126,13): warning CS6018: 'AzureAppConfigurationRefreshOptions.SetCacheExpiration(TimeSpan)' is obsolete: 'The CacheExpirationInterval property is obsolete. Use the CacheExpirationInterval property instead.'
176 #14 22.80 /app/src/Web/Program.cs(131,13): warning CS6018: 'FeatureFlagOptions.CacheExpirationInterval' is obsolete: 'The CacheExpirationInterval property is obsolete. Use the CacheExpirationInterval property instead.'
177 #14 23.25 Web -> /app/src/Web/bin/Release/net8.0/Web.dll
178 #14 24.00 Optimizing assemblies for size may change the behavior of the app. Be sure to test after publishing. See: https://aka.ms/
```

6. Deploy

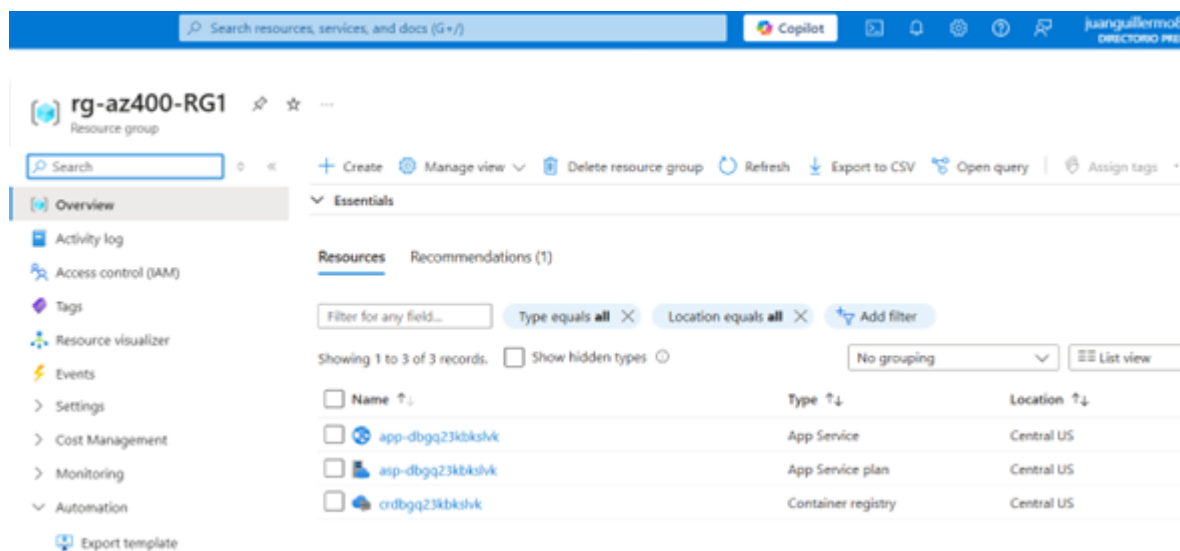
The screenshot shows the 'Deploy' job in the 'eshoponweb-ci-docker' pipeline. The job is currently running. The left sidebar shows the job steps: Initialize job (2s), Checkout eShopOnWe... (2s), Deploy App Service... (1m 2s), Add Role Assignment ... (26s), Post-job: Checkout eS... (<1s), Finalize Job (<1s), and Report build status (<1s). The main panel displays the deployment log, which includes commands like 'az login' and 'az aks deploy'.

```
1 ##[warning]See https://aka.ms/azdo-ubuntu-24.04 for changes to the ubuntu-24.04 image. Some tools (e.g. Pulumi, NuGet, Terraform) are no
2
3 Pool: Azure Pipelines
4 Image: ubuntu-latest
5 Agent: Hosted Agent
6 Started: Yesterday at 10:57 AM
7 Duration: 1m 34s
8
9 Job preparation parameters
```

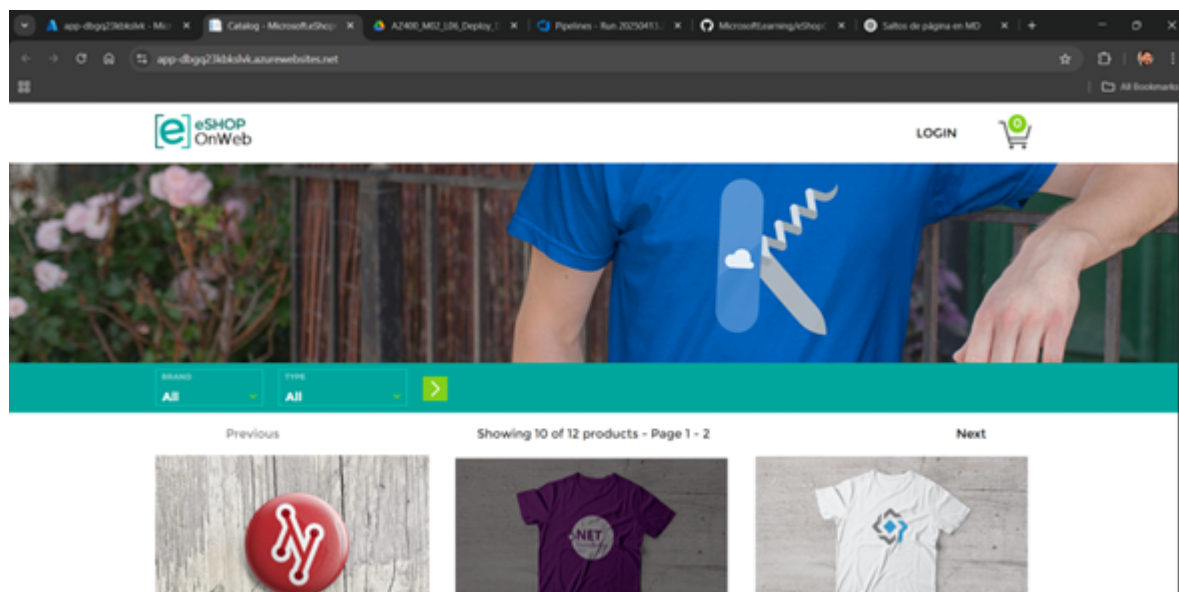
7. Verificar los recursos



8. ingresar al app service



9. ir al browser



Reflexión sobre el proceso de ejecución

Durante la ejecución del laboratorio, tuve la oportunidad de profundizar en el uso de herramientas como Azure DevOps y Azure, lo que me permitió no solo completar las tareas asignadas, sino también comprender de manera más integral cómo se gestiona un flujo de trabajo en la nube.

Una de las principales dificultades que enfrenté fue la configuración inicial de los pipelines en Azure DevOps. Al principio, me costó entender cómo se definían los pasos del pipeline y cómo vincularlos correctamente al repositorio del proyecto. Sin embargo, investigando la documentación oficial y haciendo pruebas iterativas, logré superar esta barrera y automatizar correctamente los procesos de integración y despliegue continuo (CI/CD).

Otra dificultad fue la configuración de los recursos en Azure, particularmente el aprovisionamiento de máquinas virtuales y servicios asociados, donde fue necesario entender correctamente los grupos de recursos, políticas de seguridad y costos asociados. Esto me llevó a ser más consciente de la planificación de recursos y el control de versiones, elementos clave en cualquier entorno profesional.

Las herramientas utilizadas fueron esenciales para lograr un resultado exitoso. Azure DevOps permitió gestionar el trabajo de manera estructurada mediante tableros Kanban y tareas bien definidas, mientras que Azure facilitó el despliegue ágil y seguro de los componentes del sistema, promoviendo un enfoque moderno en la entrega de soluciones.

De este proceso aprendí la importancia de la automatización, la planificación y el monitoreo constante. También comprendí que seguir los pasos sin entender su propósito puede llevar a errores difíciles de detectar. Reflexionar sobre cada acción, validar resultados y aprovechar las herramientas disponibles me permitió mejorar mis competencias técnicas y fortalecer mi capacidad para resolver problemas en entornos reales de desarrollo y operaciones.

Juan Guillermo Álvarez

Ingeniero DevSecOps

Valparaíso, Chile – Abril 2025

[linkedin.com/in/juan-guillermo-alvarez-gallardo](https://www.linkedin.com/in/juan-guillermo-alvarez-gallardo)