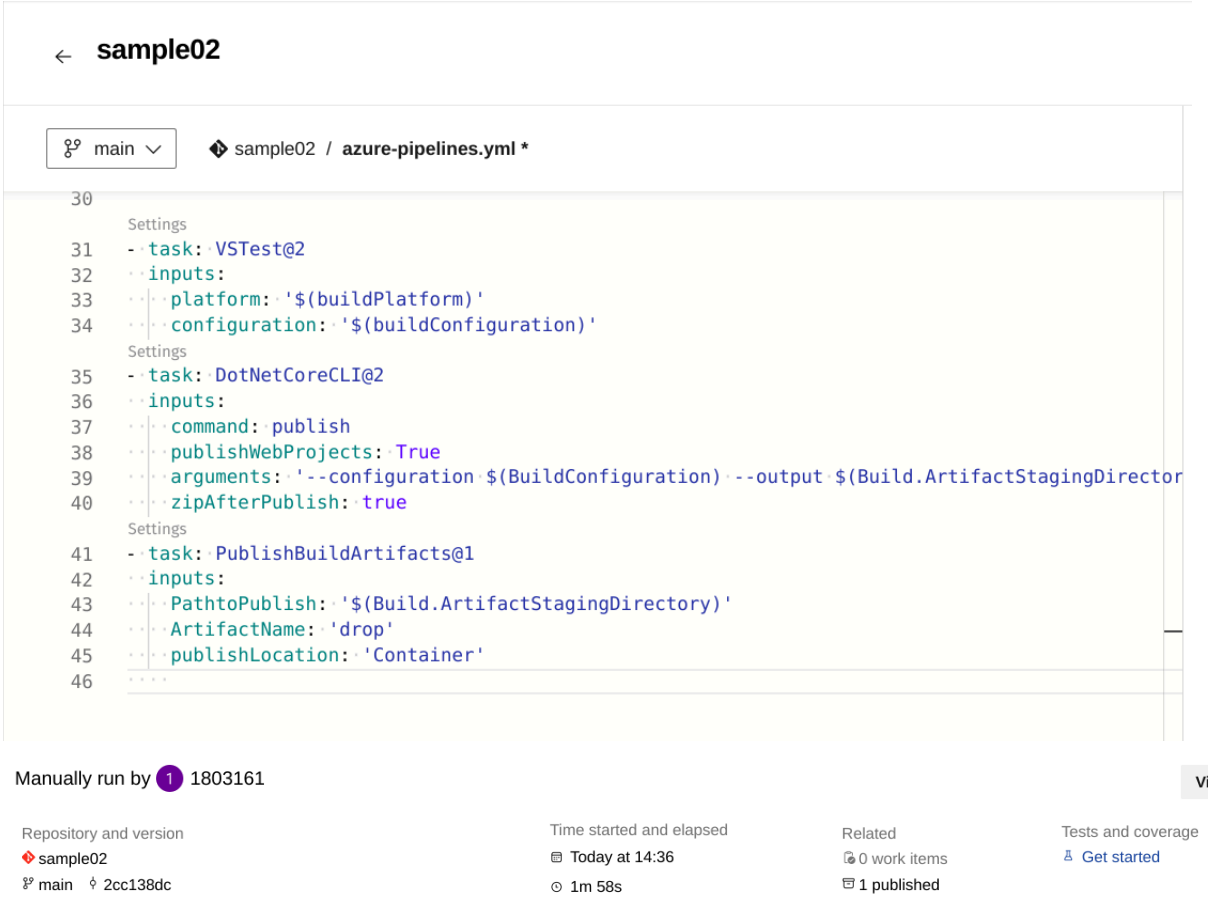
TP 4 - Pipelines

4.1 Verificar acceso a Pipelines concedido

Request Details:

Name	Carolina
Email	1803161@ucc.edu.ar
Organization Name	1803161
Parallelism Type	Private
Public Repositories	https://dev.azure.com/1803161/sample02
Explanation	Educational purposes
Note	According to our policy, we could grant you only private parallelism. We would encourage you to change the project visibility to Private on Project Settings page.

4.2 Agregar en pipeline YAML una tarea de Publish.



4.3 Explicar por qué es necesario contar con una tarea de Publish en un pipeline que corre en un agente de Microsoft en la nube.

Cuando un pipeline se ejecuta en un agente en la nube, ese agente es efímero. Una vez que termina la ejecución del pipeline, todo el entorno, incluidos los archivos generados durante la ejecución (builds, binarios, logs, etc.), se destruye. La tarea de Publish permite guardar esos artefactos de manera persistente, almacenándolos en un lugar donde puedan ser usados en etapas posteriores del pipeline o por otros pipelines.

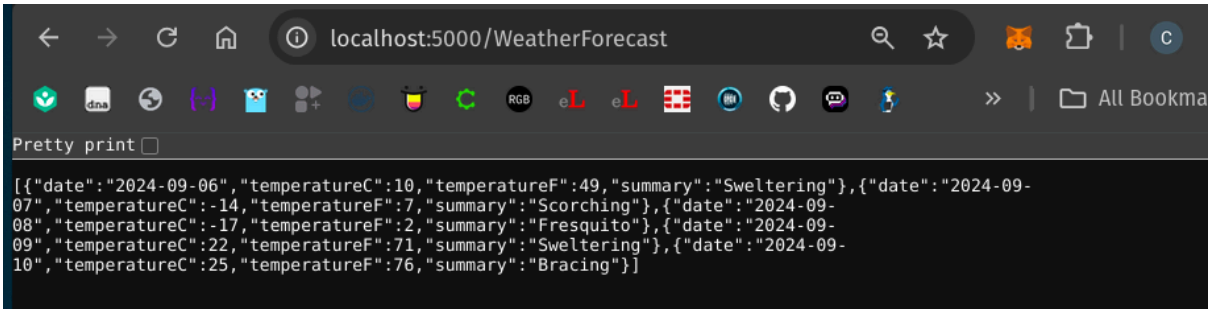
4.4 Descargar el resultado del pipeline y correr localmente el software compilado.

Name	Size
drop	3 MB
SimpleWebAPI.deploy-readme.txt	4 KB
SimpleWebAPI.deploy.cmd	15 KB
SimpleWebAPI.Parameters.xml	320 B
SimpleWebAPI.SetParameters.xml	147 B
SimpleWebAPI.SourceManifest.xml	148 B
SimpleWebAPI.zip	2 MB
WebApp.zip	2 MB

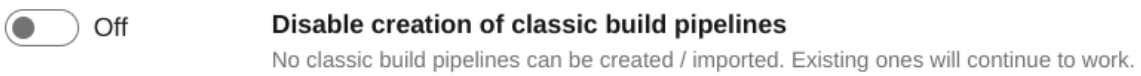
Descargar drop.
Descomprimir drop.zip y descomprimir SimpleWebApi.zip

```
caro@e14:~/Documents/tp_ingsw3/drop 4.4/SimpleWebAPI$ dotnet SimpleWebAPI.dll
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/caro/Documents/tp_ingsw3/drop 4.4/SimpleWebAPI
```

localhost:5000/WeatherForecast

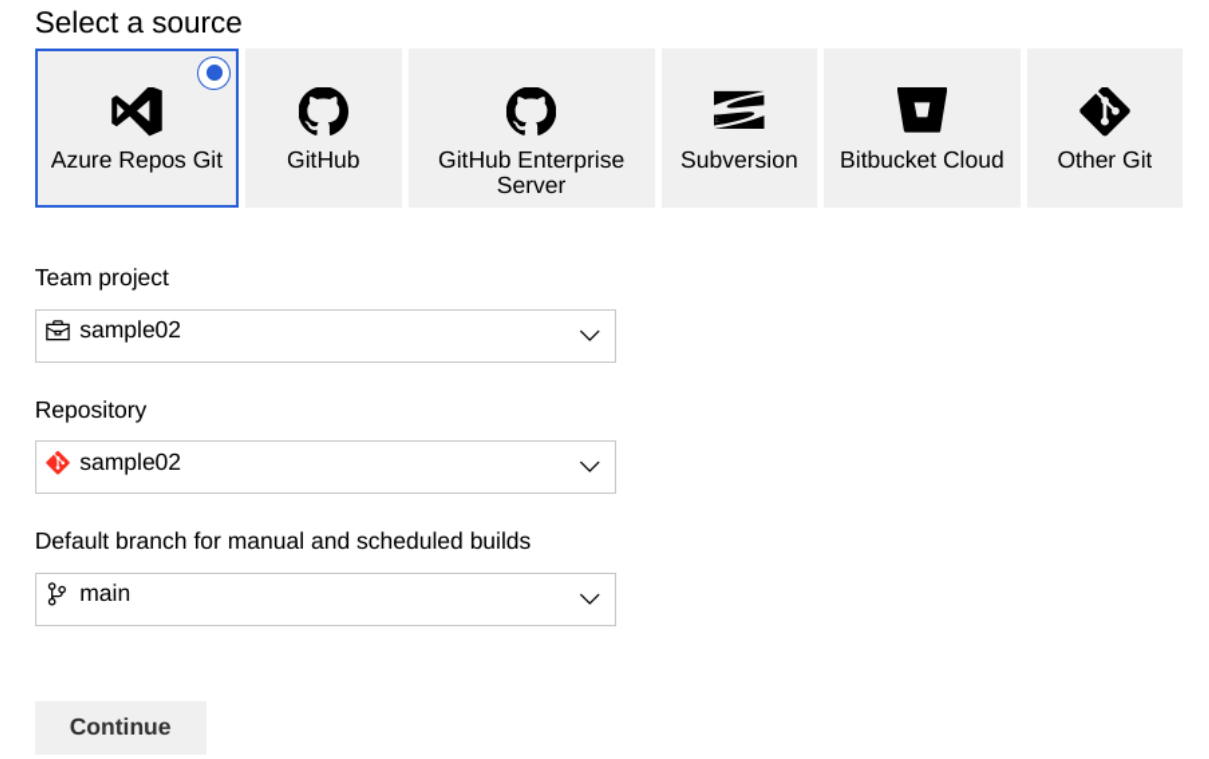


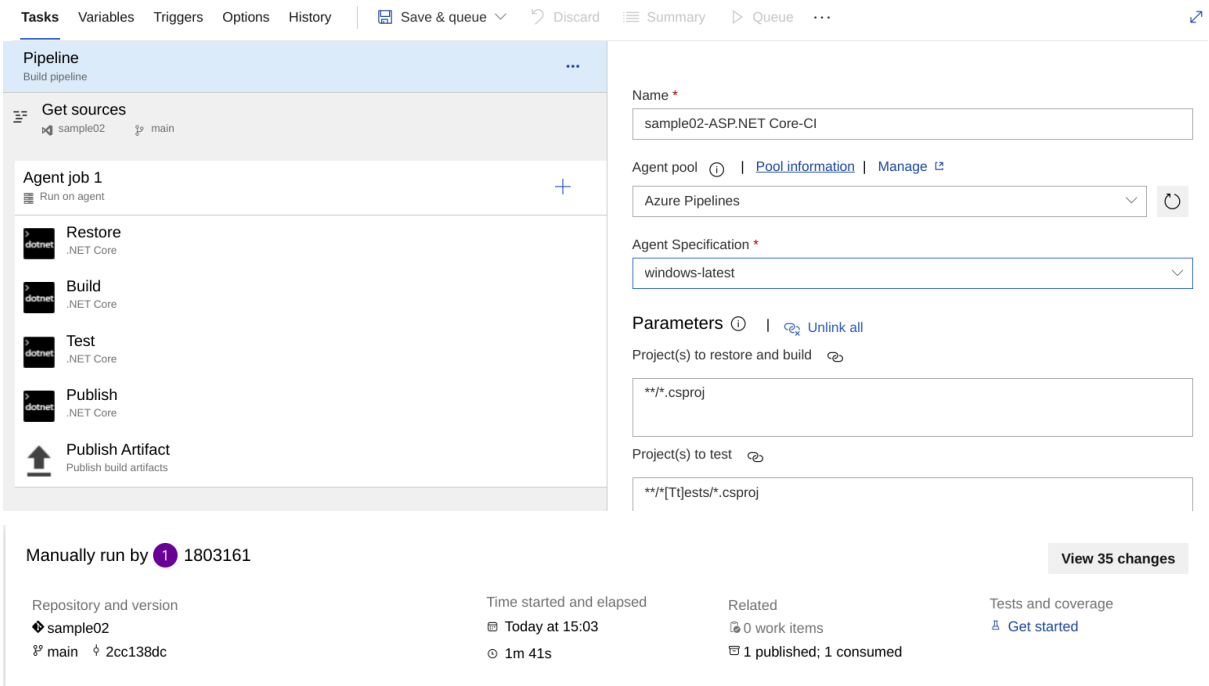
4.5 Habilitar el editor clásico de pipelines. Explicar las diferencias claves entre este tipo de editor y el editor YAML.



Editor clásico de pipelines: usa una GUI para configurar las opciones del pipeline, agregar tareas, etc. Es más fácil de usar para principiantes, y permite que no se requieran conocimientos de programación para crear pipelines. Menos portables porque están guardados dentro del proyecto de Azure DevOps.
Editor YAML: se basa en archivos de texto escritos en YAML. Es más difícil de utilizar, requiere conocer la sintaxis de YAML. Están almacenados como archivos en el repositorio por lo que son fáciles de reutilizar.

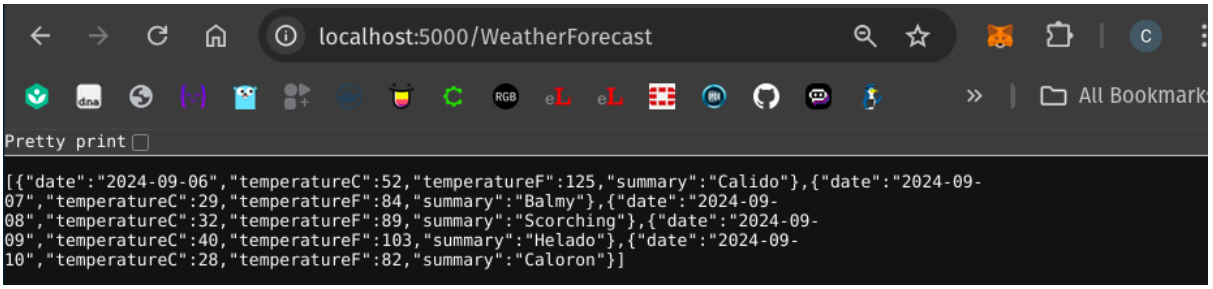
4.6 Crear un nuevo pipeline con el editor clásico. Descargar el resultado del pipeline y correr localmente el software compilado.





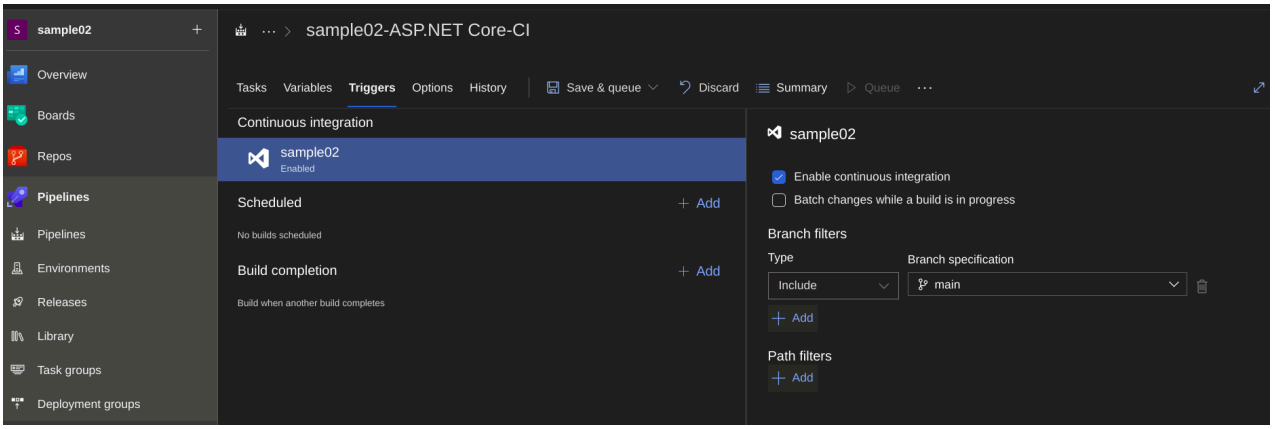
Descargar drop.
Descomprimir drop.zip y descomprimir SimpleWebApi.zip

```
caro@e14:~/Documents/tp_ingsw3/drop4.6/SimpleWebAPI$ dotnet SimpleWebAPI.dll
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/caro/Documents/tp_ingsw3/drop4.6/SimpleWebAPI
```

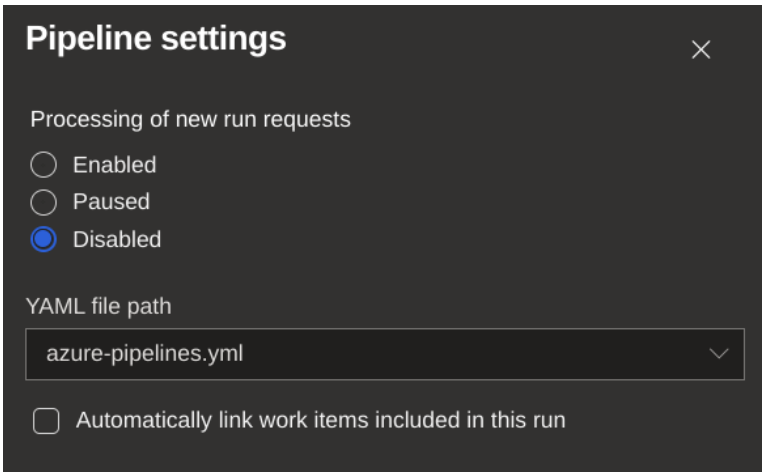


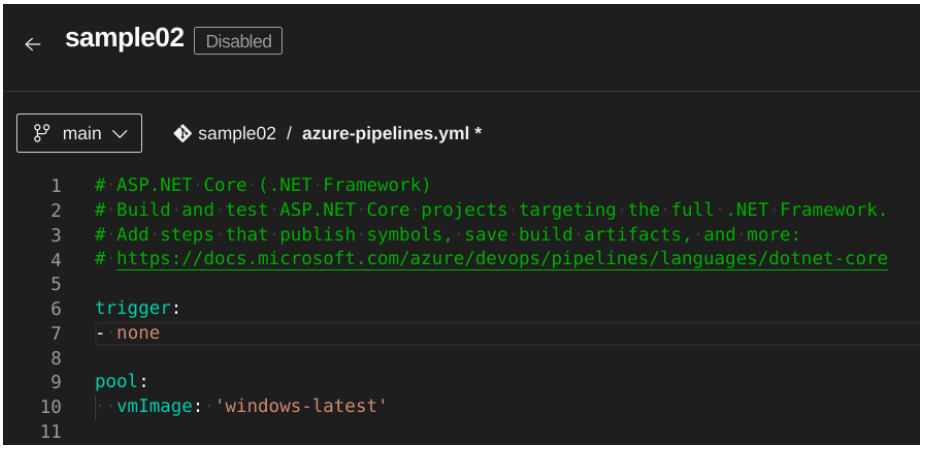
4.7 Configurar CI en ambos pipelines (YAML y Classic Editor). Mostrar resultados de la ejecución automática de ambos pipelines al hacer un commit en la rama main.

Continuous Integration en Classic Editor



Continuous Integration en YAML





4.8 Explicar la diferencia entre un agente MS y un agente Self-Hosted. ¿Qué ventajas y desventajas hay entre ambos? ¿Cuándo es conveniente y/o necesario usar un Self-Hosted Agent?

Agente Microsoft-Hosted (MS-Hosted Agents): son proporcionados y gestionados por Microsoft en su infraestructura en la nube. Se ejecutan en máquinas virtuales que Microsoft administra y actualiza.

Ventajas:
- Fácil de configurar.
- Actualización automática.
- Escalabilidad.

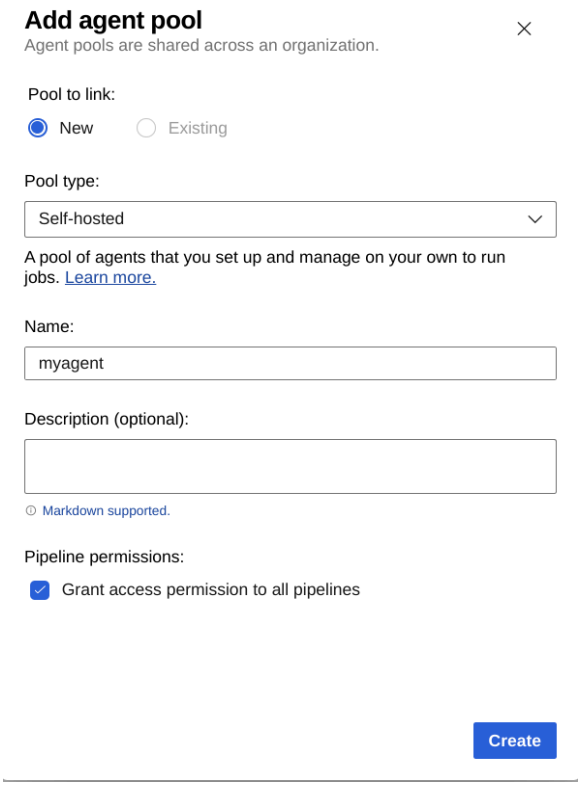
Desventajas:
- Configuración limitada: Los usuarios no tienen control sobre la configuración de estos agentes más allá de seleccionar la imagen del sistema operativo y las herramientas preinstaladas.
- Costo.
- Puede haber tiempos de espera para la disponibilidad del agente en horas pico.

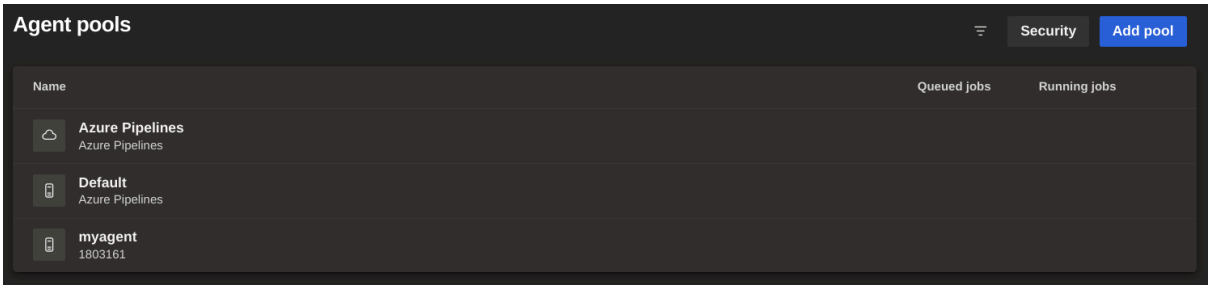
Agente Self-Hosted (Self-Hosted Agents): son instalados y gestionados por los usuarios en sus propias máquinas físicas o virtuales. Los usuarios tienen control total sobre la configuración, las herramientas instaladas y el entorno de ejecución.

Ventajas:
- Personalización.
- Control sobre el Hardware.
- Reducción de costos.
- Privacidad y seguridad.

Desventajas:
- Mantenimiento.
- Debe gestionar la escalabilidad manualmente.
- Configuración y gestión más complejas.

4.8 Crear un Pool de Agentes y un Agente Self-Hosted





4.9 Instalar y correr un agente en nuestra máquina local.
PAT personal access token: n4bhuajwmrflfcuokdc6iac5nc6iao27hvj4e6iagt6crevf7t3a

Get the agent

Windows | macOS | Linux

x64 | arm | arm64 | RHEL6

System prerequisites

Configure your account

Configure your account by following the steps outlined here.

Download the agent

Download

Create the agent

Create the agent

```
~/$ mkdir myagent && cd myagent
~/myagent$ tar zxvf ~/Downloads/vsts-agent-linux-x64-
3.243.1.tar.gz
```

Configure the agent Detailed instructions

```
~/myagent$ ./config.sh
```

Optionally run the agent interactively

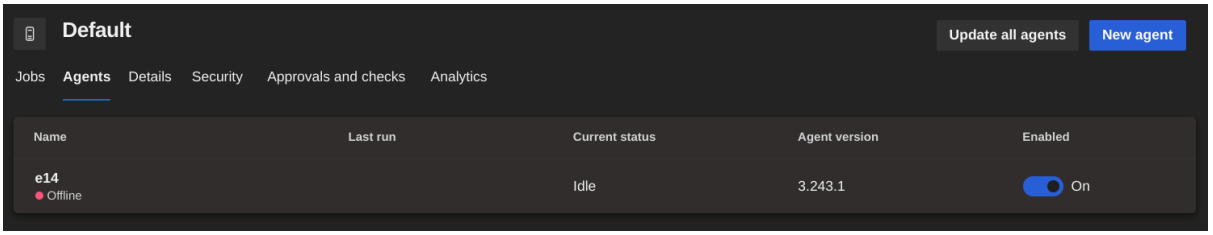
If you didn't run as a service above:

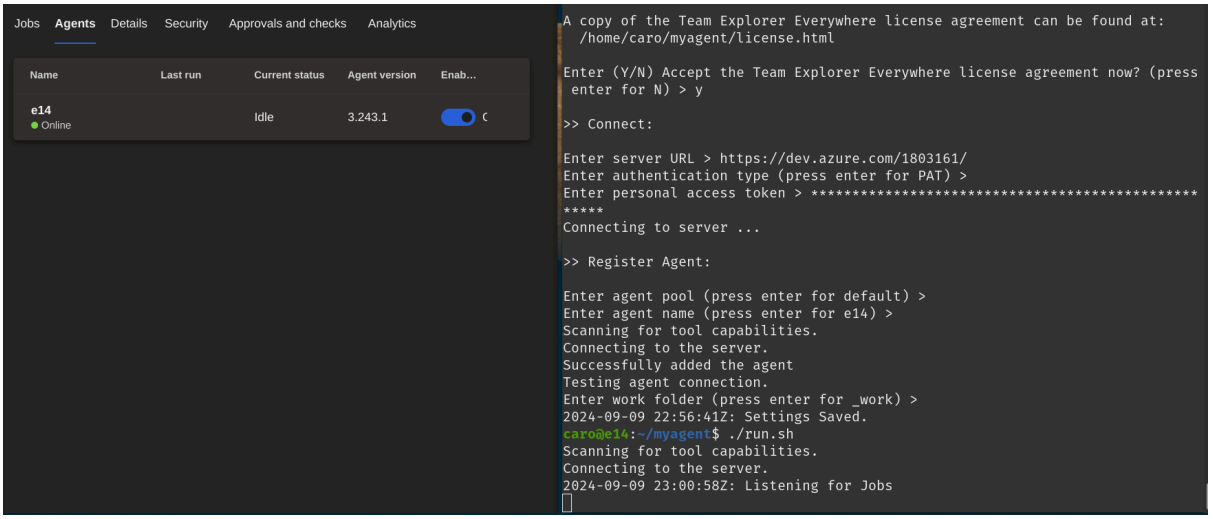
```
~/myagent$ ./run.sh
```

That's it!

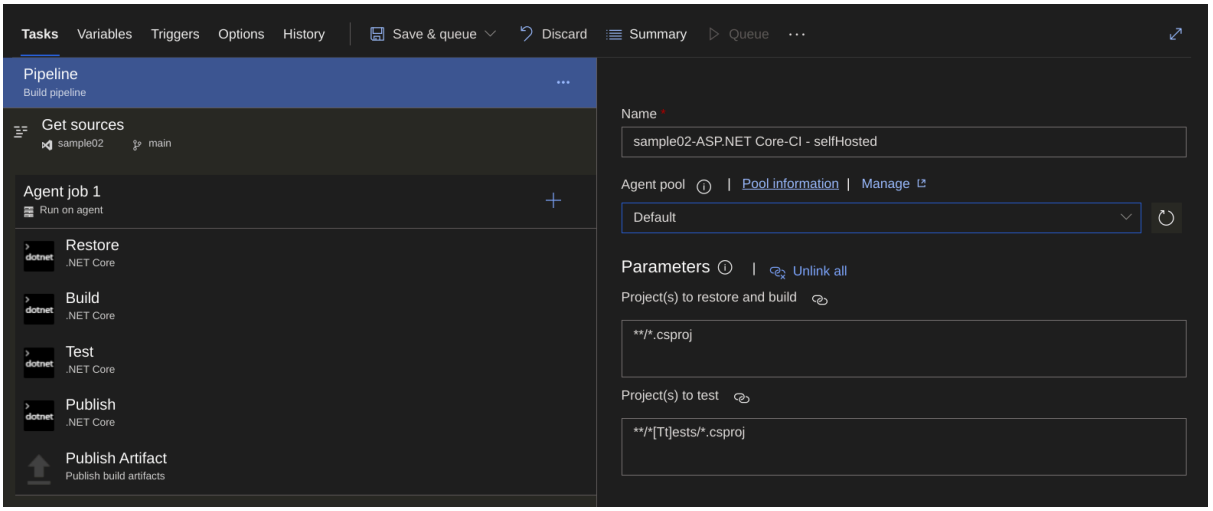
More Information

```
caro@e14:~/myagent$ ./config.sh

  ___                      ______ _            _ _
 / _ \                     | ___ (_)          | (_)
/ /_\ \_____   _ _ __ ___  | |_/ /_ _ __   ___| |_ _ __   ___ ___
|  _  |_  / | | | '__/ _ \ |  __/| | '_ \ / _ \ | | '_ \ / _ \/ __|
| | | |/ /| |_| | | |  __/ | |   | | |_) |  __/ | | | | |  __/\__ \
\_| |_/___|\__,_|_|  \___| \_|   |_| .__/ \___|_|_|_| |_|\___||___/
                                   | |
       agent v3.243.1              |_|         (commit 3bb22cd)


>> End User License Agreements:

Building sources from a TFVC repository requires accepting the Team Explorer Everywhere End User License Agreement. This step is not required for building so
urces from Git repositories.

A copy of the Team Explorer Everywhere license agreement can be found at:
  /home/caro/myagent/license.html

Enter (Y/N) Accept the Team Explorer Everywhere license agreement now? (press enter for N) > y

>> Connect:

Enter server URL > https://dev.azure.com/1803161/
Enter authentication type (press enter for PAT) >
Enter personal access token > **************************************************
Connecting to server ...

>> Register Agent:

Enter agent pool (press enter for default) >
Enter agent name (press enter for e14) >
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
Enter work folder (press enter for _work) >
2024-09-09 22:56:41Z: Settings Saved.
```

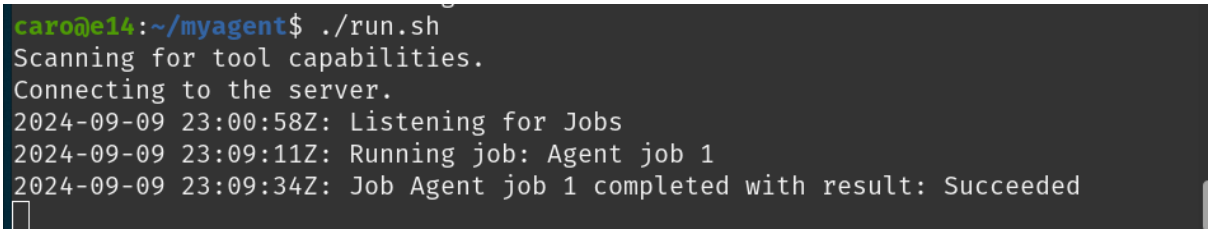


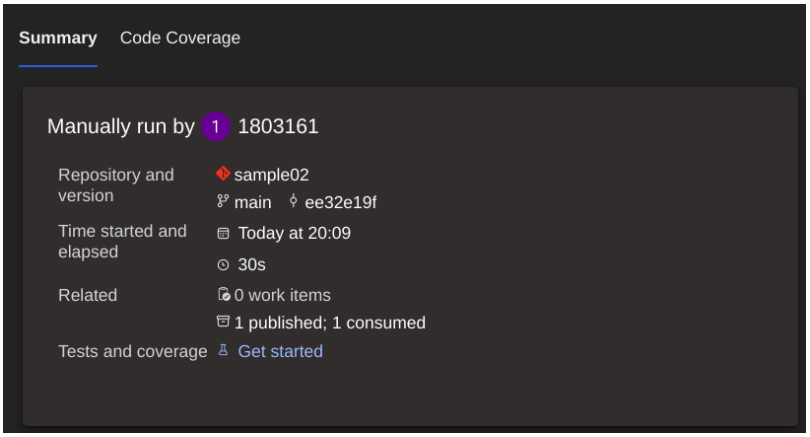


4.10 Crear un pipeline que use el agente Self-Hosted alojado en nuestra máquina local.

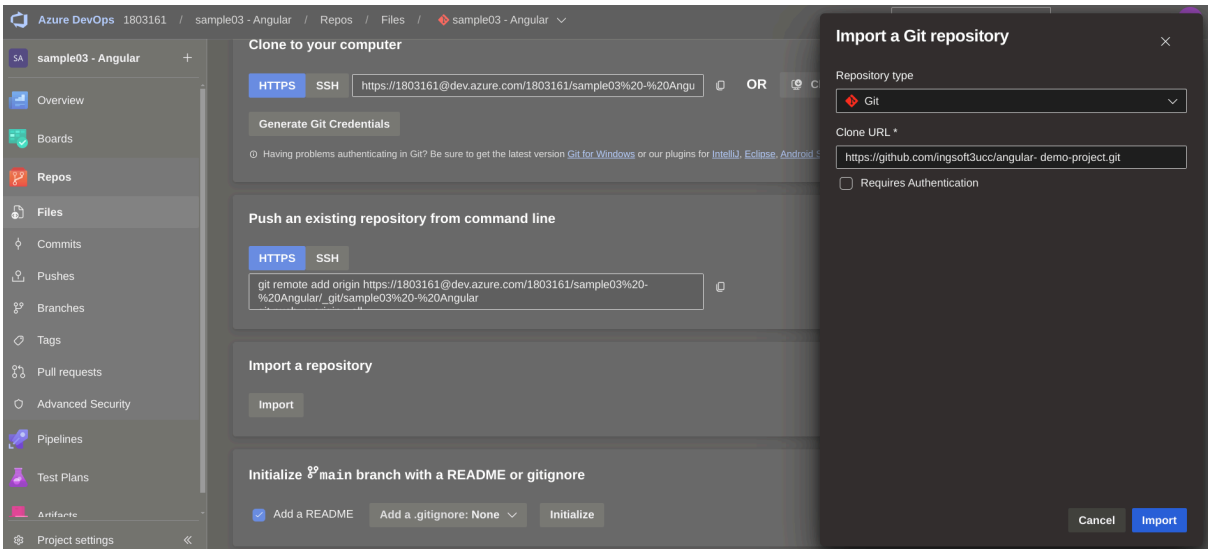


4.11 Buscar el resultado del pipeline y correr localmente el software compilado.

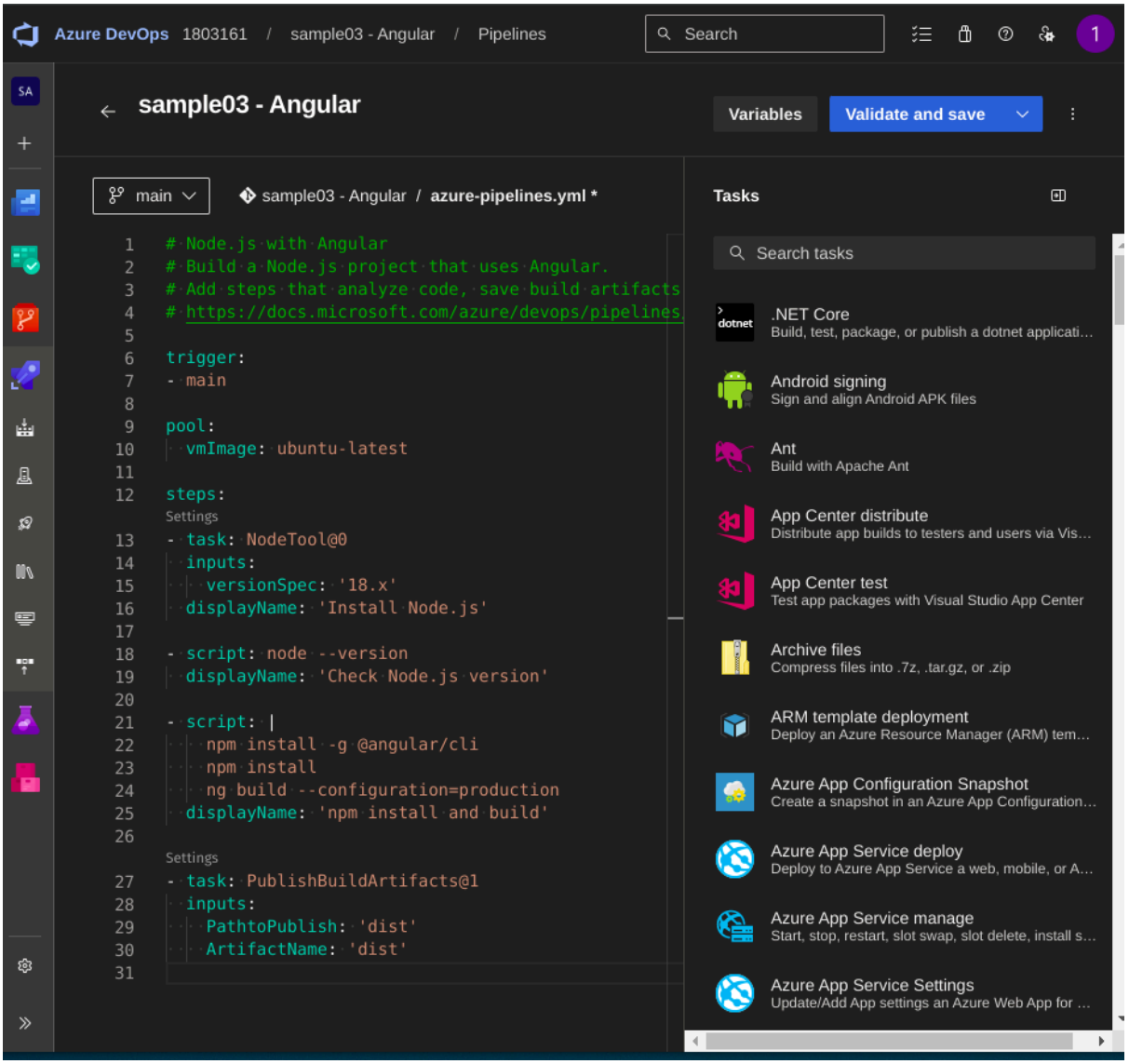


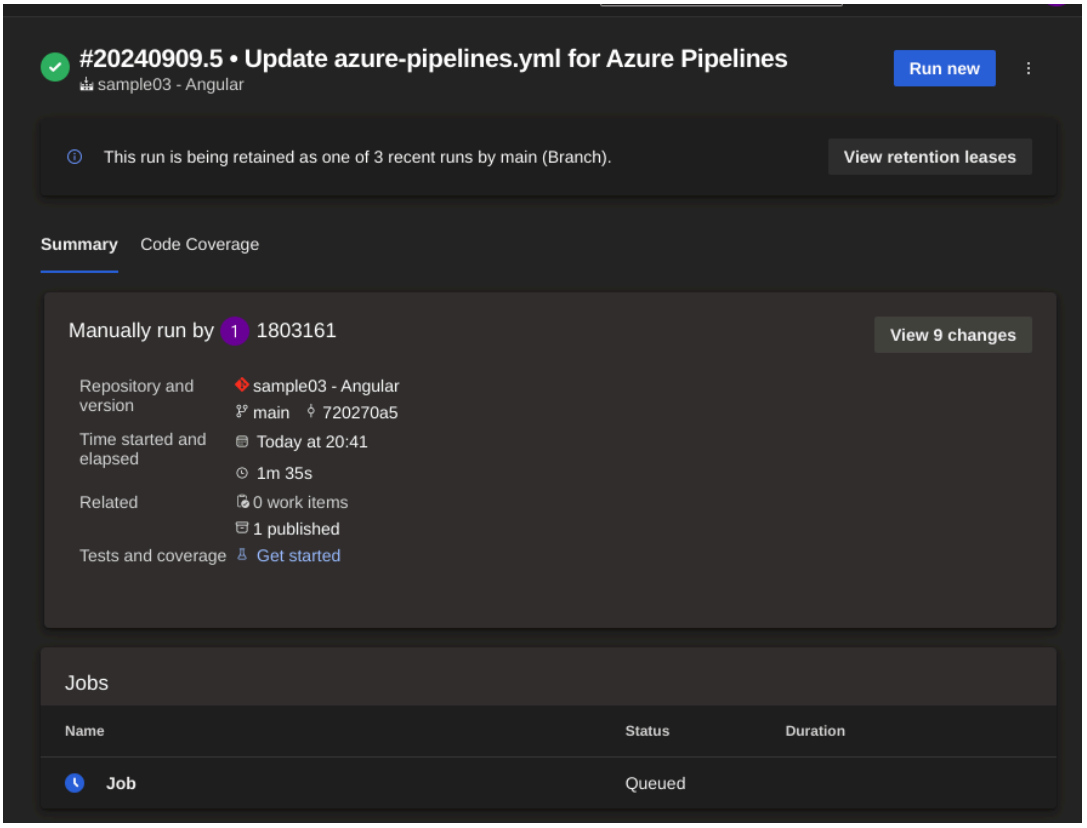


4.12 Crear un nuevo proyecto en ADO clonado desde un repo que contenga una aplicación en Angular como por ejemplo https://github.com/ingsoft3ucc/angular-demo-project.git

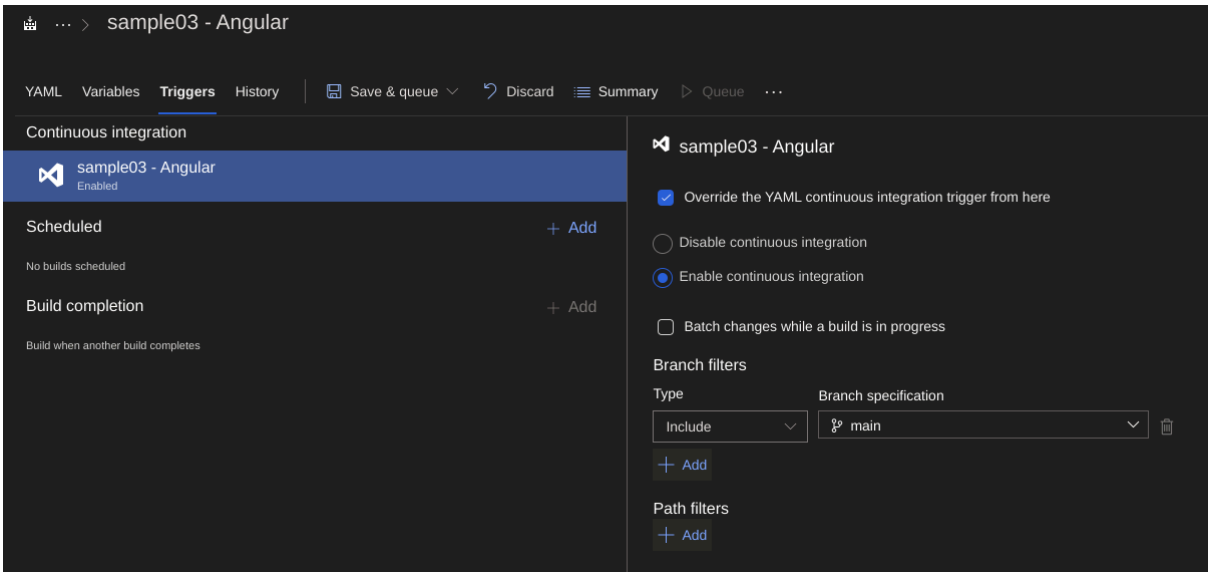


4.13 Configurar un pipeline de build para un proyecto de tipo Angular como el clonado.

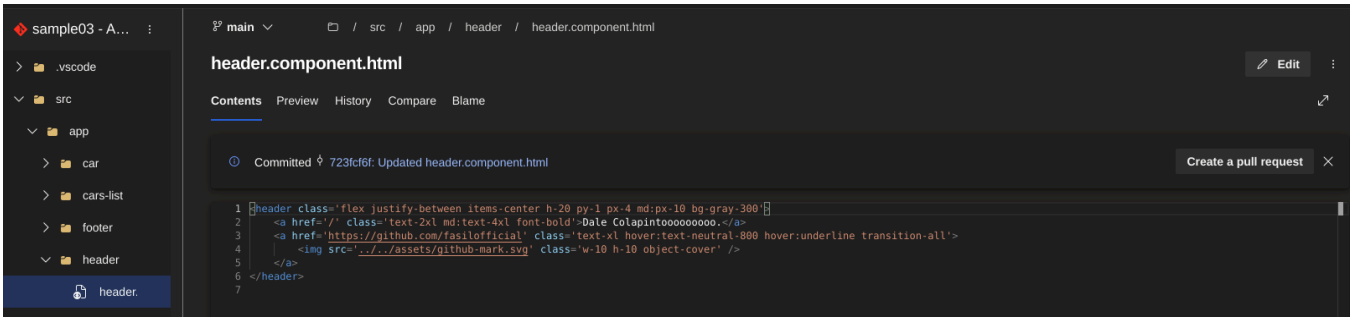

Azure DevOps 1803161 / sample03 - Angular / Pipelines
Search
sample03 - Angular
Variables
Validate and save
main
sample03 - Angular / azure-pipelines.yml *
Node.js with Angular
Build a Node.js project that uses Angular.
Add steps that analyze code, save build artifacts
https://docs.microsoft.com/azure/devops/pipelines
trigger:
- main
pool:
vmImage: ubuntu-latest
steps:
Settings
- task: NodeTool@0
inputs:
versionSpec: '18.x'
displayName: 'Install Node.js'
- script: node --version
displayName: 'Check Node.js version'
- script: |
npm install -g @angular/cli
npm install
ng build --configuration=production
displayName: 'npm install and build'
Settings
- task: PublishBuildArtifacts@1
inputs:
PathtoPublish: 'dist'
ArtifactName: 'dist'
Tasks
Search tasks
.NET Core
Build, test, package, or publish a dotnet applicati…
Android signing
Sign and align Android APK files
Ant
Build with Apache Ant
App Center distribute
Distribute app builds to testers and users via Vis…
App Center test
Test app packages with Visual Studio App Center
Archive files
Compress files into .7z, .tar.gz, or .zip
ARM template deployment
Deploy an Azure Resource Manager (ARM) tem…
Azure App Configuration Snapshot
Create a snapshot in an Azure App Configuration…
Azure App Service deploy
Deploy to Azure App Service a web, mobile, or A…
Azure App Service manage
Start, stop, restart, slot swap, slot delete, install s…
Azure App Service Settings
Update/Add App settings an Azure Web App for …

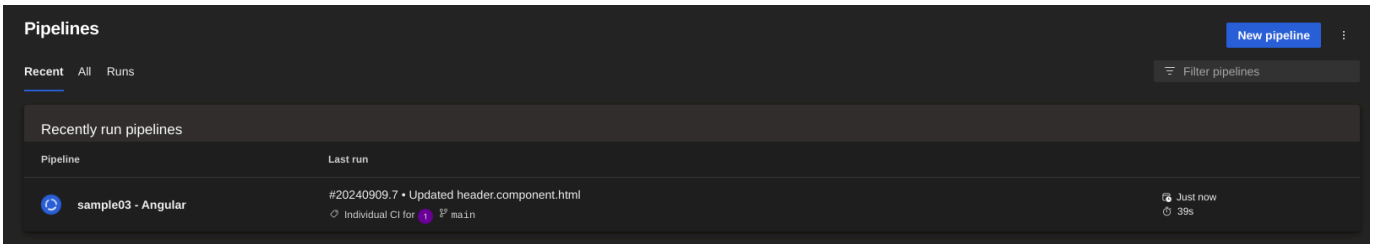


4.14 Habilitar CI para el pipeline.



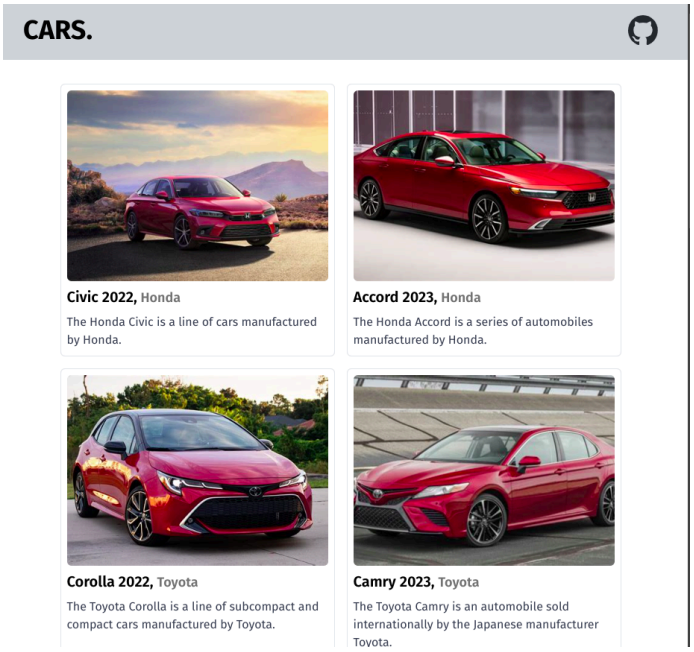
4.15 Hacer un cambio a un archivo del proyecto (algún cambio en el HTML que se renderiza por ejemplo) y verificar que se ejecute automáticamente el pipeline.





4.16 Descargar el resultado del pipeline y correr en un servidor web local el sitio construido.

4.17 Mostrar el antes y el después del cambio
Antes



Después

```
caro@e14:~/Documents/tp_ingsw3/tp4.ej4.16/first-app/server$ node server.mjs
Node Express server listening on http://localhost:4000
```

Dale Colapintooooooooo.

Civic 2022, Honda

The Honda Civic is a line of cars manufactured by Honda.

Accord 2023, Honda

The Honda Accord is a series of automobiles manufactured by Honda.

Corolla 2022, Toyota

The Toyota Corolla is a line of subcompact and compact cars manufactured by Toyota.

Camry 2023, Toyota

The Toyota Camry is an automobile sold internationally by the Japanese manufacturer Toyota.