AzureDevops: How to configure your laptop as self-hosted-agent to run Azure DevOps jobs

We are going to explain how to set up **Jobs** that **run on machines** that you manage (your **latptop** or a **Cloud Virtual Machine**), where you **install agent software supplied by Microsoft**

For this purpose **Azure DevOps** provides a feature called "**self-hosted agents**" for exactly this purpose

Self-hosted agents allow you to **run build and deployment jobs** directly **on machines you manage**, rather than on **Microsoft-hosted agents**

This approach gives you **more control over the environment** in which your jobs run, including the ability to **customize** the **operating system**, **installed software**, and **hardware specifications**

Here's a step-by-step guide to setting up **self-hosted agents** in **Azure DevOps**:

1. Prepare the Machine

Ensure your target machine meets the requirements:

Supported operating system (Windows, macOS, Linux)

Sufficient hardware resources (CPU, memory, disk space)

Network connectivity to Azure DevOps services

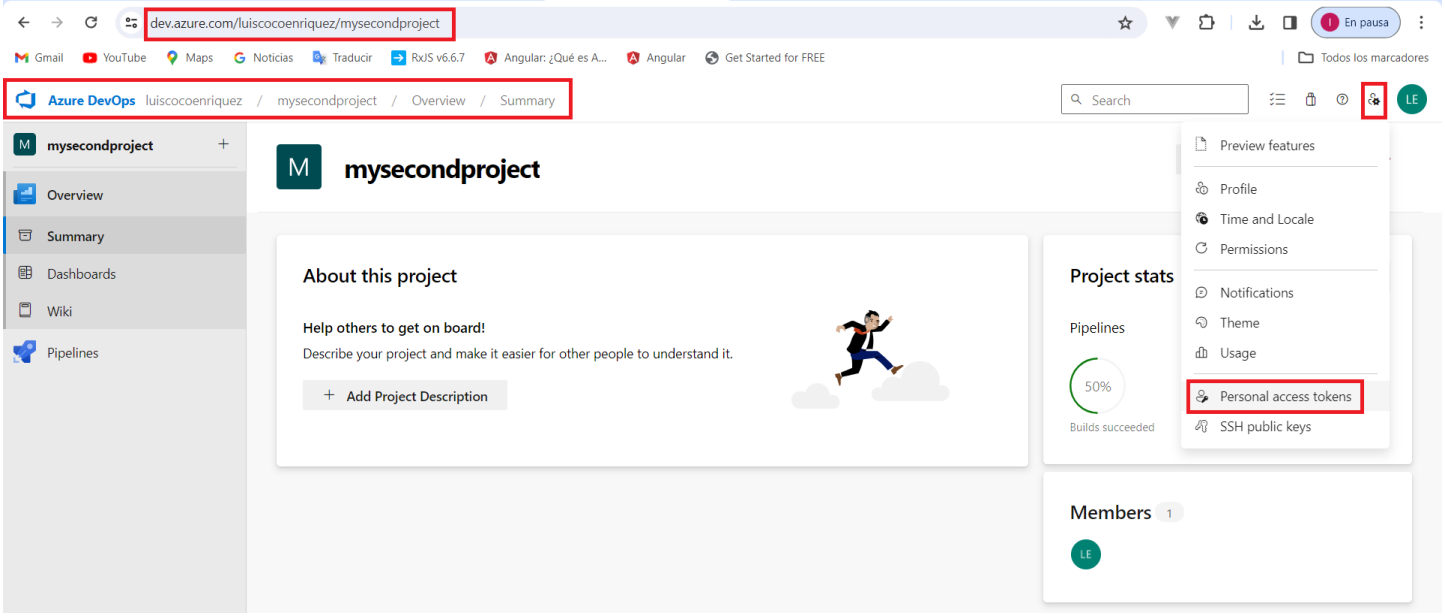
Required software installed (e.g., development tools, SDKs)

2. Create a Personal Access Token (PAT) in Azure DevOps

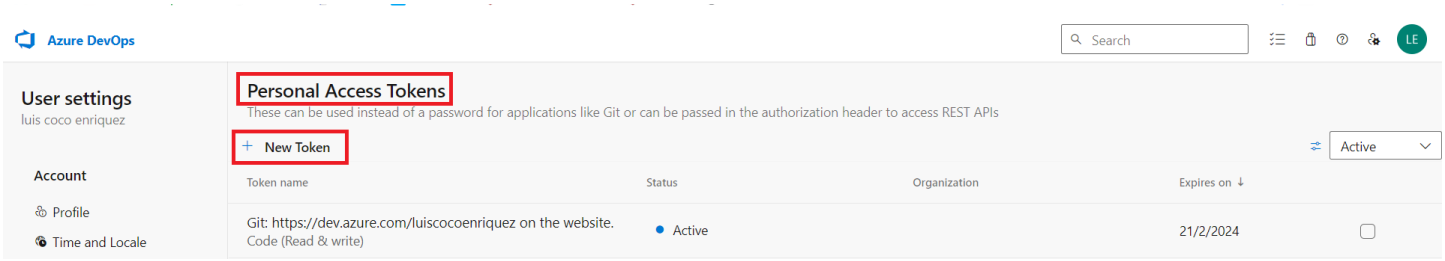
You'll need a **PAT** to authenticate the agent with Azure DevOps:

We **Sign in** to your Azure DevOps organization

We go to **User settings** -> **Personal access tokens**

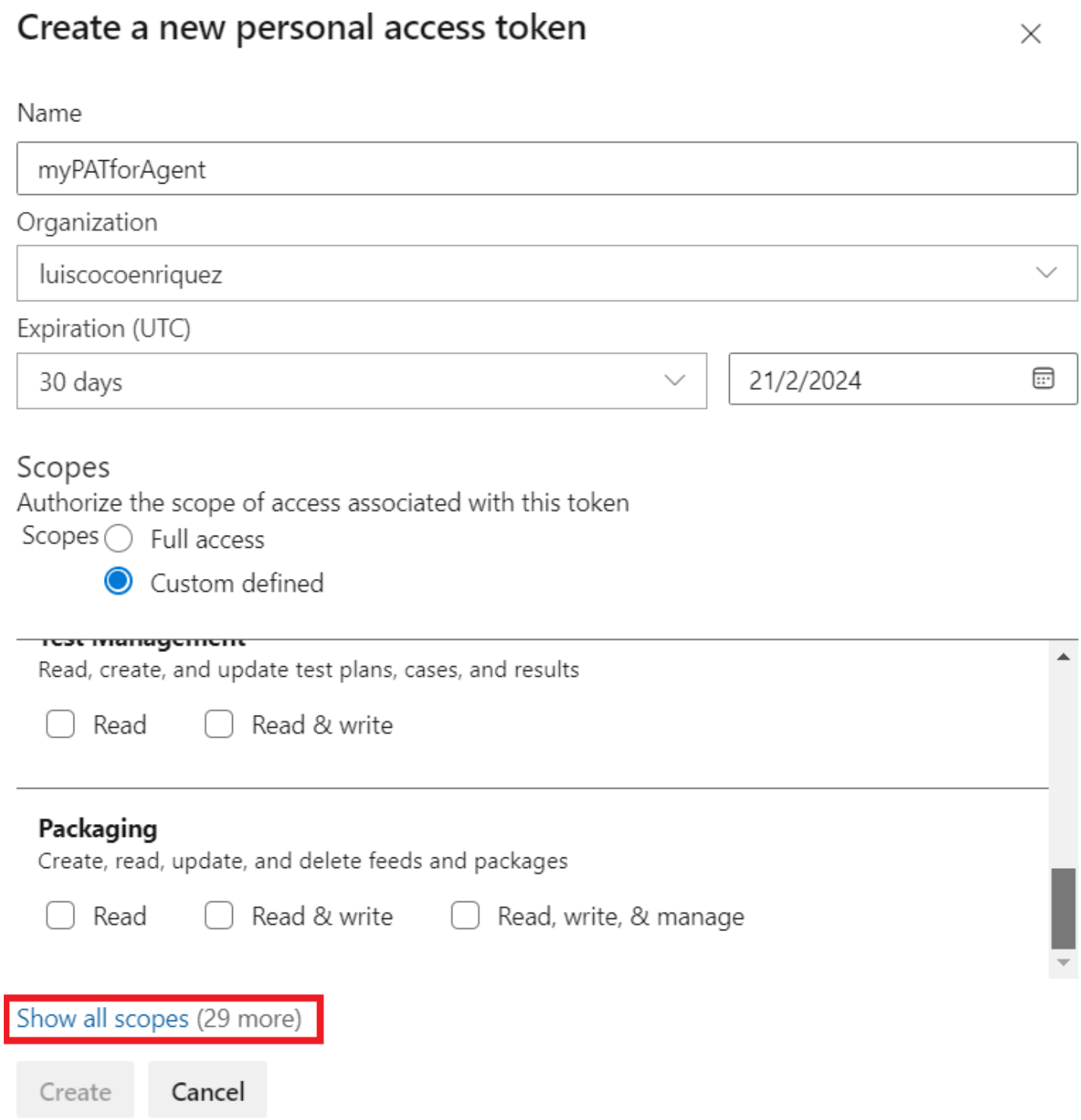


We click **New Token**

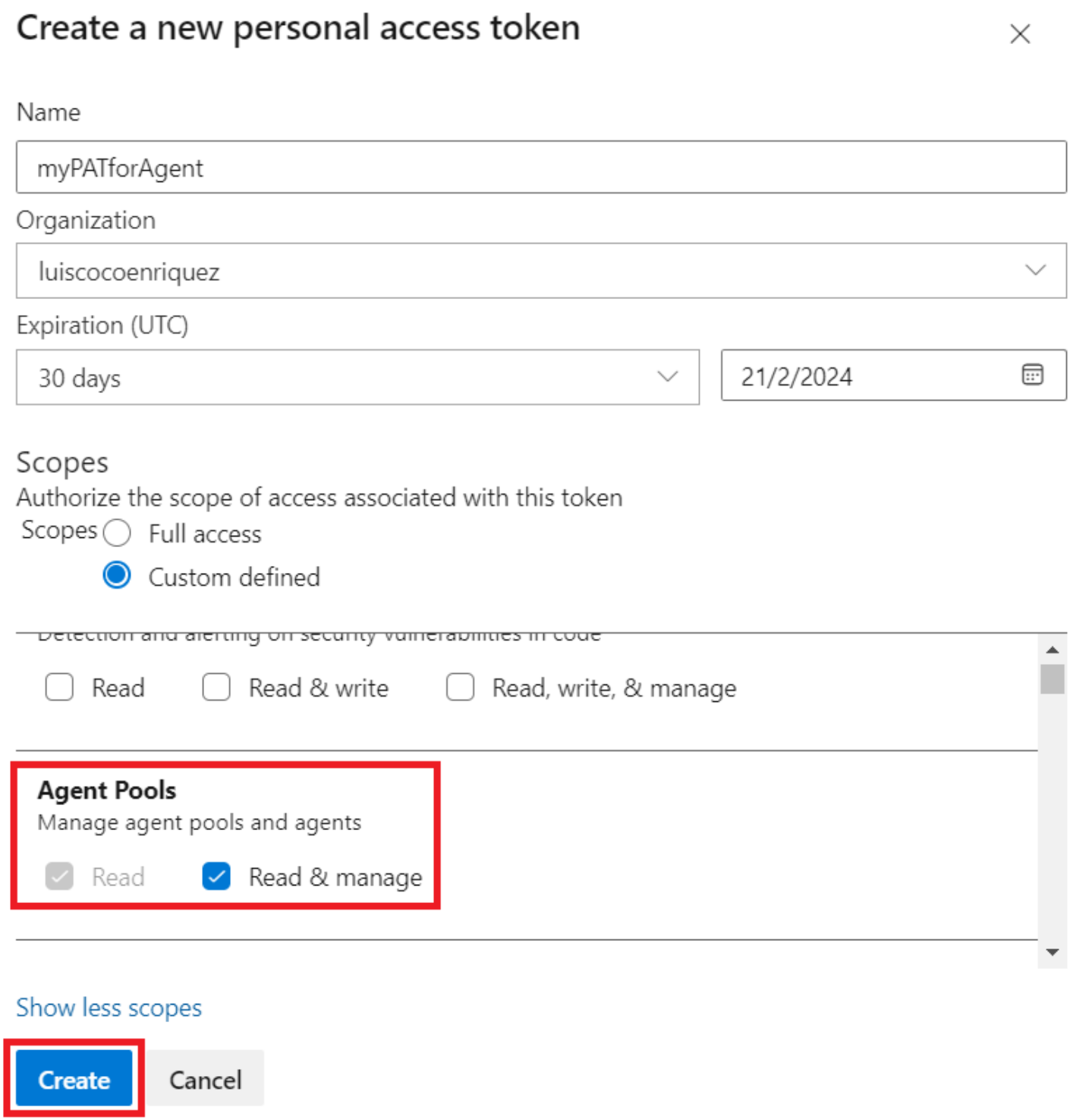


We give it a descriptive **name**, choose an **expiration**, and select the **Agent Pools (read, manage)** scope under Agent Pools

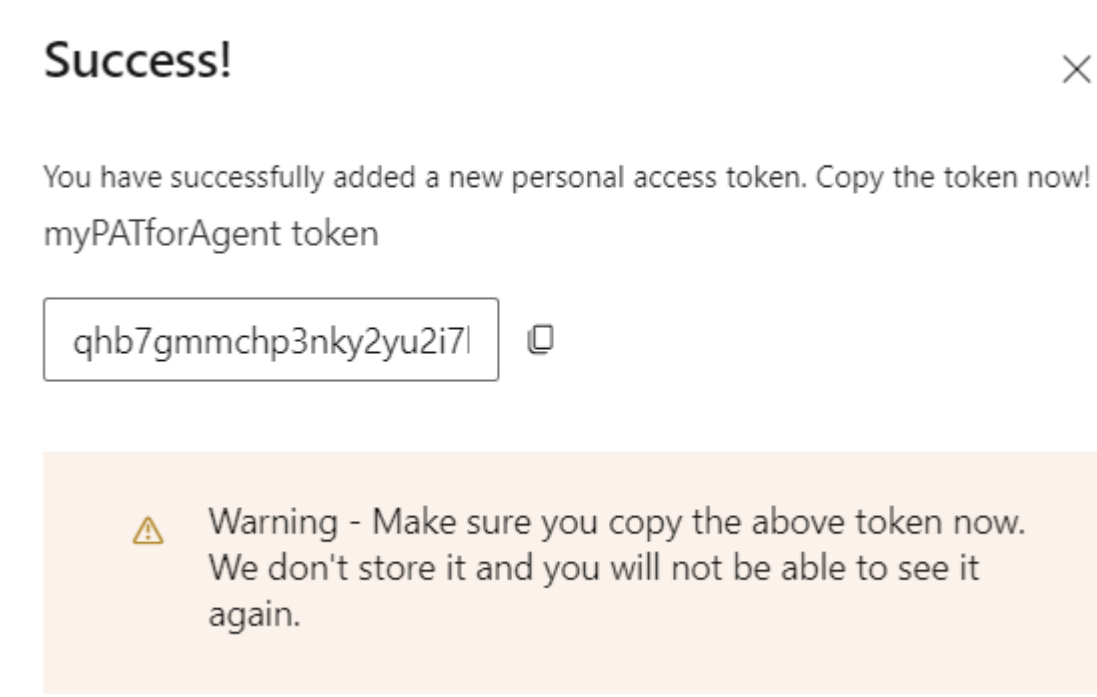
We press on **Show all scopes** link and we select **Agent Pools (read, manage)**



Then we click **Create** and copy the token



Store it securely; you won't be able to see it again

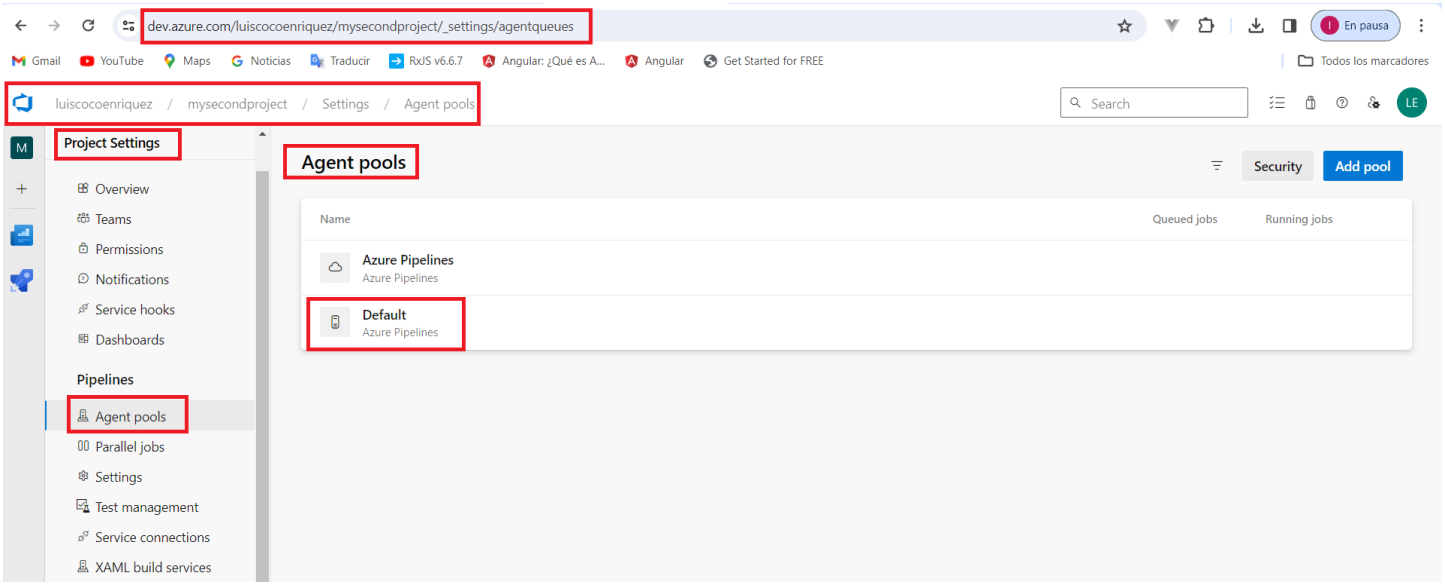


3. Download, Configure and Run the Agent

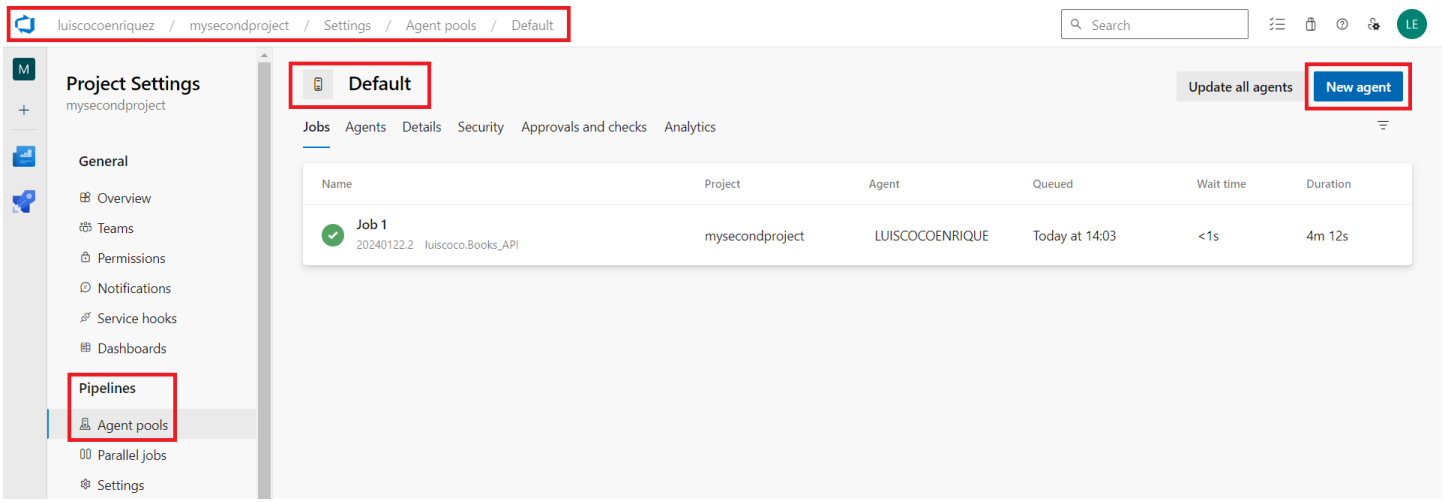
3.1. We create a Default Agent pool

In Azure DevOps, we go to **Project settings** > **Agent pools**

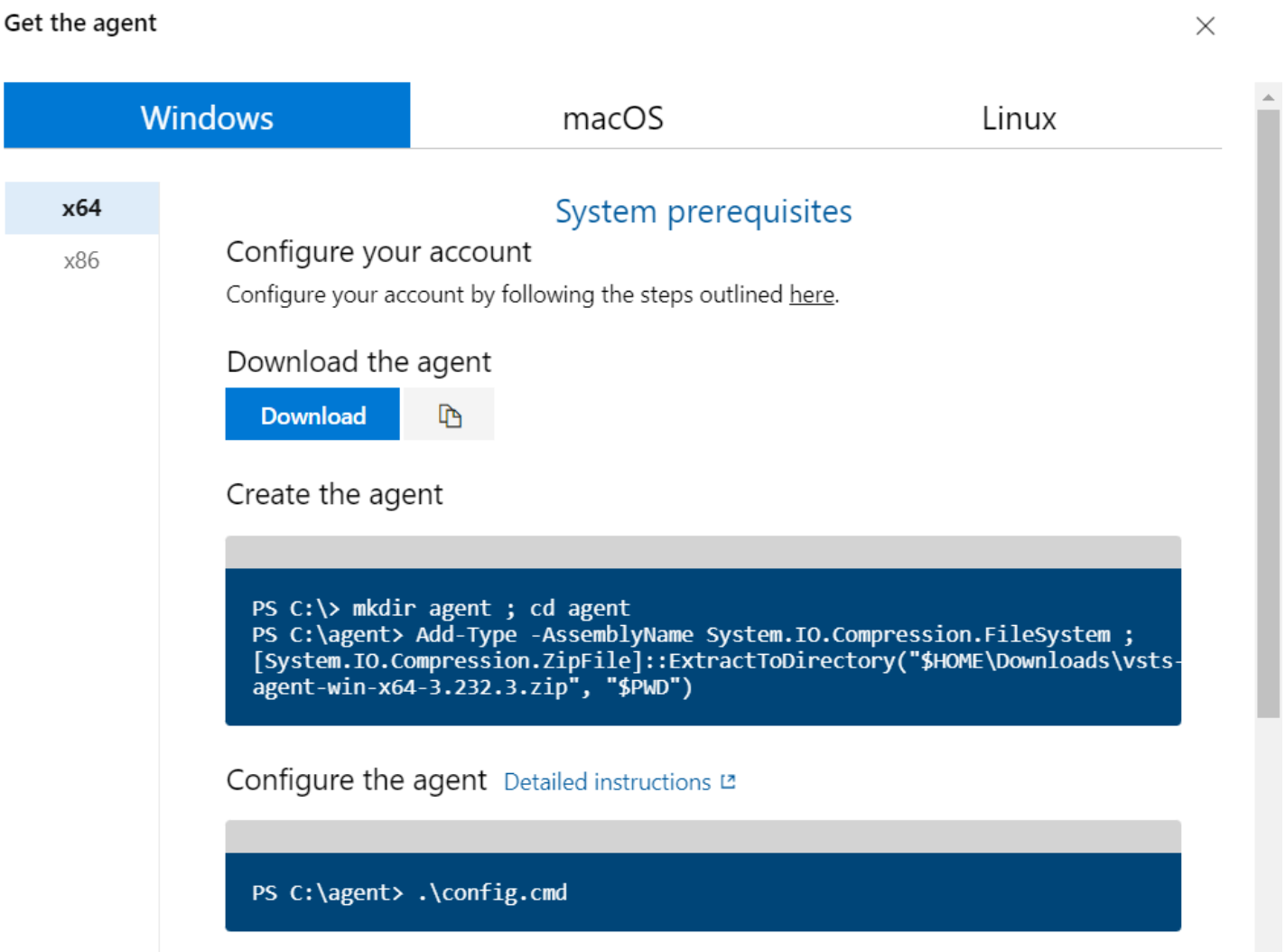
We choose the **default pool** or we create a new one

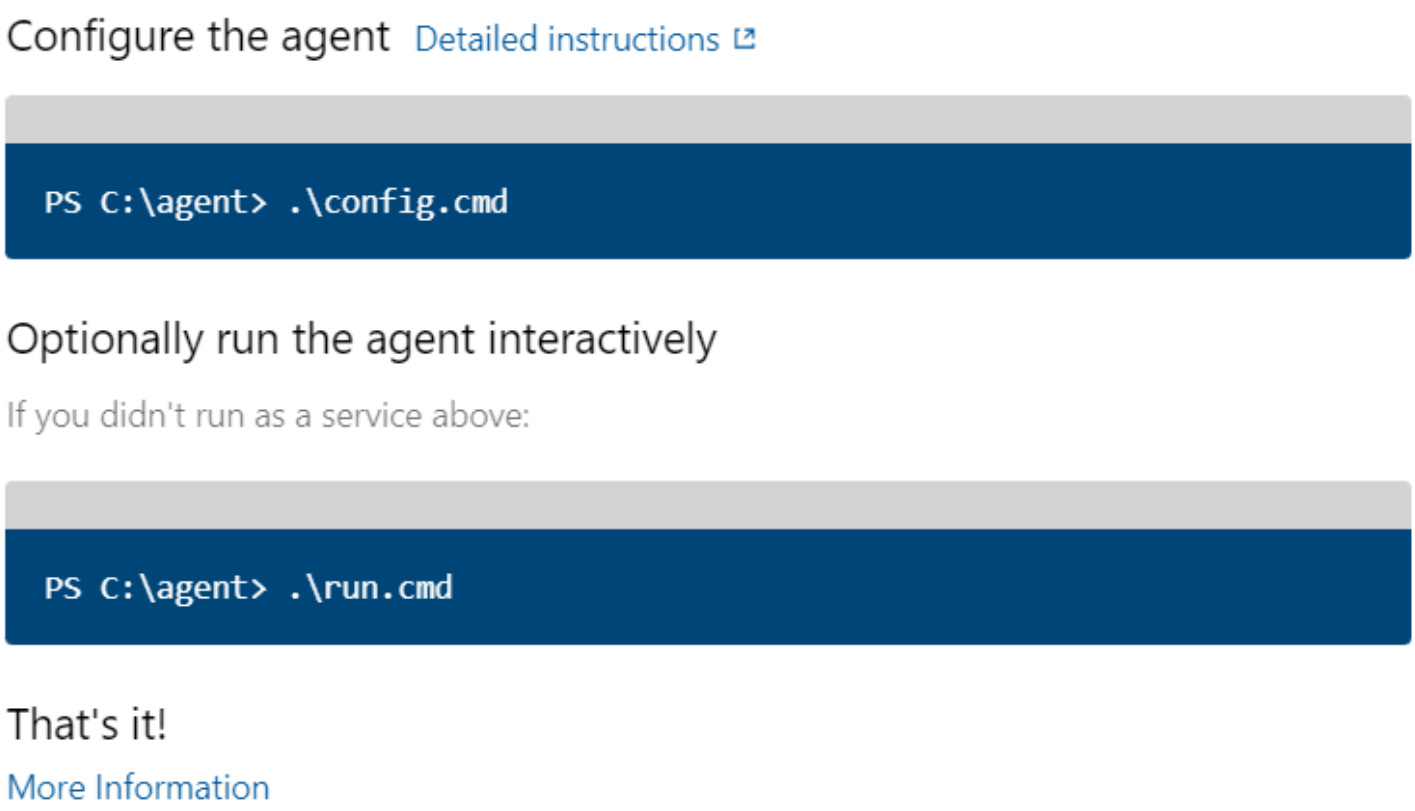


We click **New agent** and we select the operating system of your target machine



3.2. We download the agent package



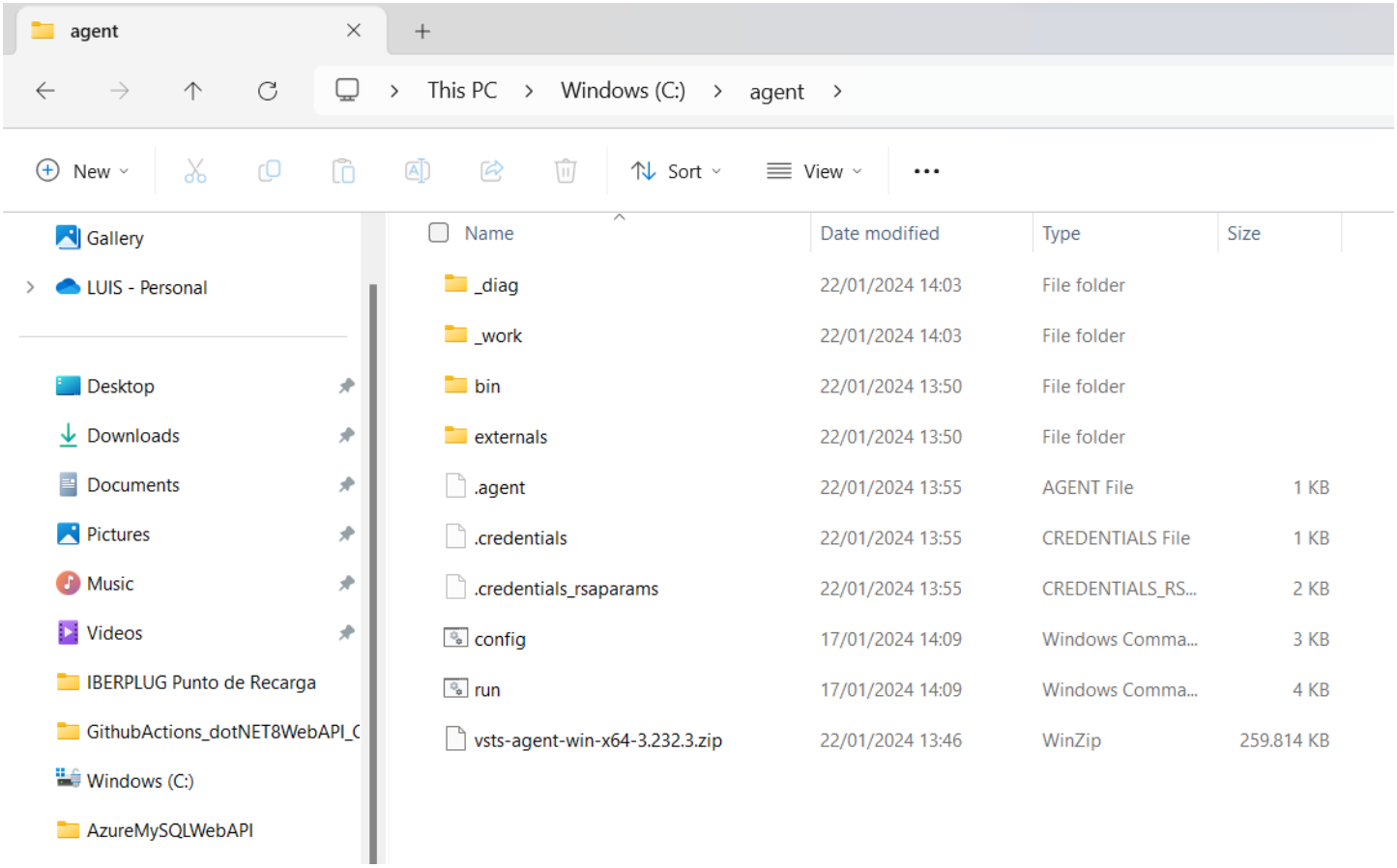


We run PowerShell and we execute the following commands:

3.3. We create the agent

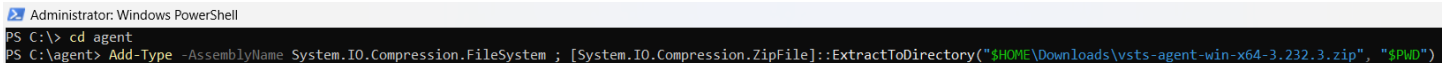
```
PS C:\> mkdir agent ; cd agent
```

We move the downloaded agent ZIP file from the **Downloads** folder to the in the **agent** directory



We create the agent from the ZIP file

```
PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ; [System.IO.Compression.
```



3.4. We configure the agent

```
PS C:\agent> .\config.cmd
```

```
PS C:\agent> .\config.cmd


  ___                      ______ _            _ _
 / _ \                     | ___ (_)          | (_)
/ /_\ \_____   _ _ __ ___  | |_/ /_ _ __   ___| |_ _ __   ___  ___
|  _  |_  / | | | '__/ _ \ |  __/| | '_ \ / _ \ | | '_ \ / _ \/ __|
| | | |/ /| |_| | | |  __/ | |   | | |_) |  __/ | | | | |  __/\__ \
\_| |_/___|\__,_|_|  \___| \_|   |_| .__/ \___|_|_|_| |_|\___||___/
                                    | |
       agent v3.232.3               |_|          (commit 948e48a)


>> Conectar:

Escribir dirección URL del servidor > https://dev.azure.com/luiscocoenriquez/
Entrar tipo de autenticación (presione Entrar para PAT) > 6znuwyv53z6bx6aelmaabi75utl4xg5qadcsrz56ndypjg33wpja
Escriba un valor válido para tipo de autenticación.
Entrar tipo de autenticación (presione Entrar para PAT) > 6znuwyv53z6bx6aelmaabi75utl4xg5qadcsrz56ndypjg33wpja
Escriba un valor válido para tipo de autenticación.
Entrar tipo de autenticación (presione Entrar para PAT) >
Escribir token de acceso personal > **************************************************
Conectando con el servidor...

>> Registrar agente:

Entrar grupo de agentes (presione Entrar para default) >
Entrar nombre del agente (presione Entrar para LUISCOCOENRIQUE) >
Examinando las capacidades de la herramienta.
Conectando al servidor.
El agente se agregó correctamente
Probando la conexión del agente.
Entrar carpeta de trabajo (presione Entrar para _work) >
2024-01-22 12:55:35Z: Configuración guardada.
Entrar ¿Ejecutar el agente como servicio? (S/N) (presione Entrar para N) >
Entrar ¿Configurar el inicio de sesión automático y ejecutar el agente al iniciar? (S/N) (presione Entrar para N) >
```

3.5. We run the agent

```
PS C:\agent> .\run.cmd
```

```
PS C:\agent> .\run.cmd
Examinando las capacidades de la herramienta.
Conectando al servidor.
2024-01-22 12:59:44Z: Escuchando trabajos
2024-01-22 13:03:48Z: Ejecutando el trabajo: Job
2024-01-22 13:08:02Z: el trabajo Job se completó con el resultado: Succeeded
```

4. Use the Self-hosted Agent in Your Pipelines

When defining a pipeline (YAML or through the UI), specify the pool where your self-hosted agent resides:

```
pool:
  name: MyPool # Replace with your agent pool name
```

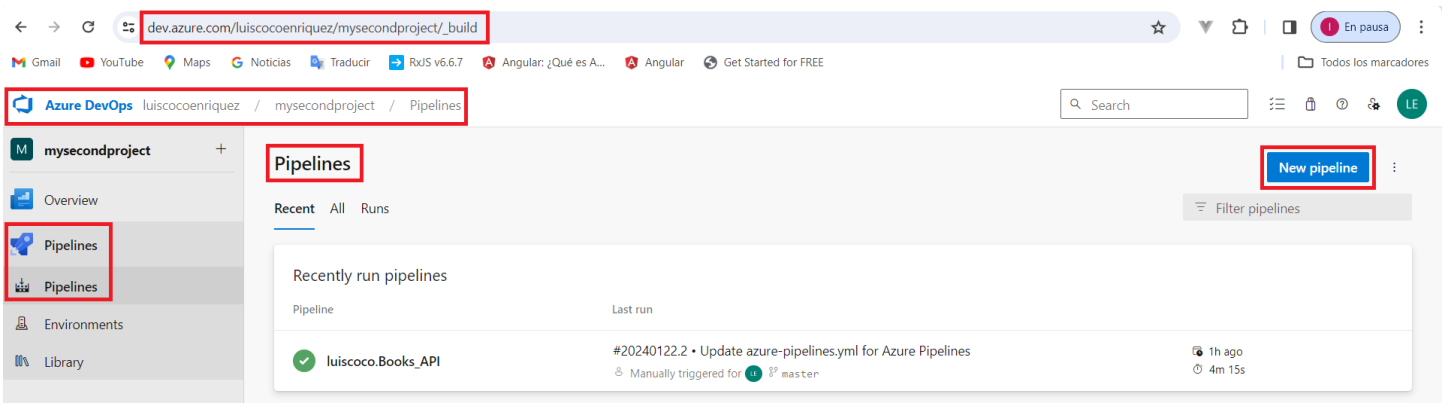
For example in our case we created a Default agent:

```
pool:
  name: 'Default'
```

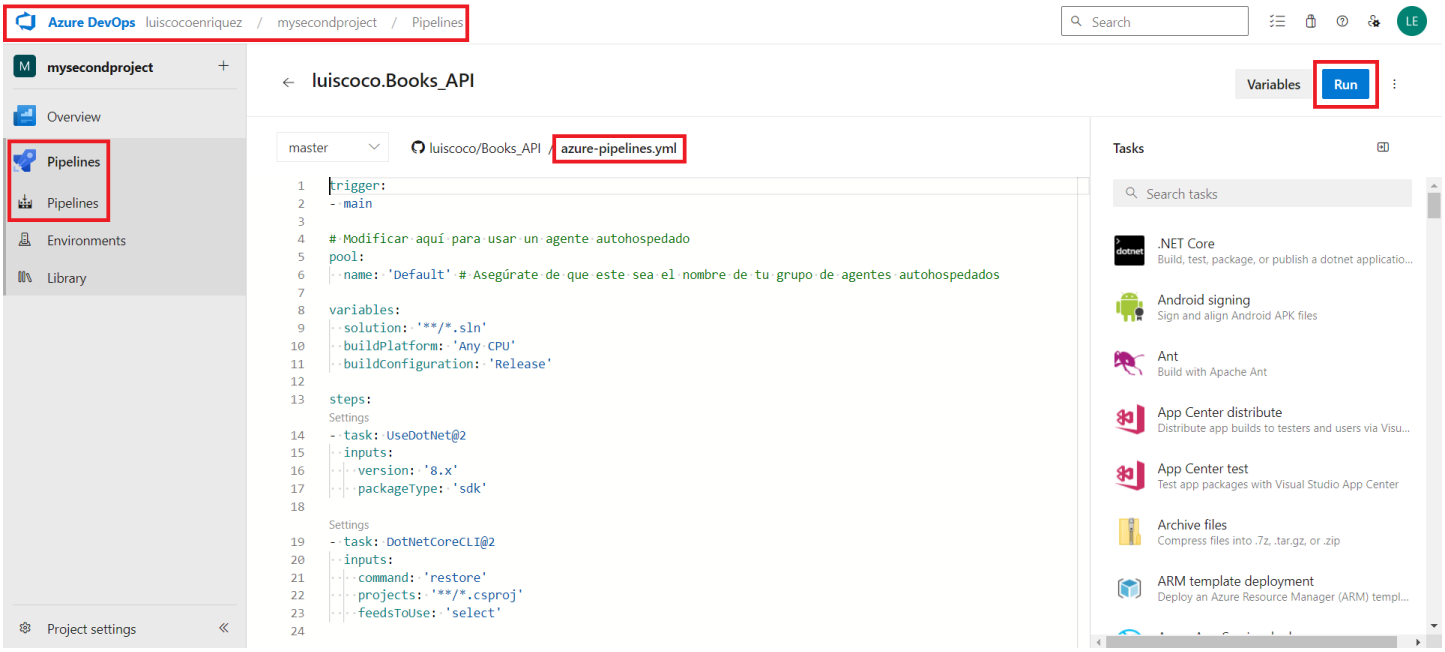
In our case see the Azure DevOps Pipeline configuration yaml file:

```
trigger:
- main

# Modificar aquí para usar un agente autohospedado
pool:
  name: 'Default' # Asegúrate de que este sea el nombre de tu grupo de agentes autohospedados

variables:
  solution: '**/*.sln'
  buildPlatform: 'Any CPU'
  buildConfiguration: 'Release'

steps:
- task: UseDotNet@2
  inputs:
    version: '8.x'
    packageType: 'sdk'

- task: DotNetCoreCLI@2
  inputs:
    command: 'restore'
    projects: '**/*.csproj'
    feedsToUse: 'select'

- task: DotNetCoreCLI@2
  inputs:
    command: 'build'
    projects: '**/*.csproj'
    arguments: '--configuration $(buildConfiguration)'

# Optional: Add steps for running tests here

- task: DotNetCoreCLI@2
  inputs:
    command: 'publish'
    publishWebProjects: true
    arguments: '--configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirector
    zipAfterPublish: true

- task: PublishBuildArtifacts@1
  inputs:
    PathtoPublish: '$(Build.ArtifactStagingDirectory)'
    ArtifactName: 'drop'
    publishLocation: 'Container'
```

We create a New Pipeline

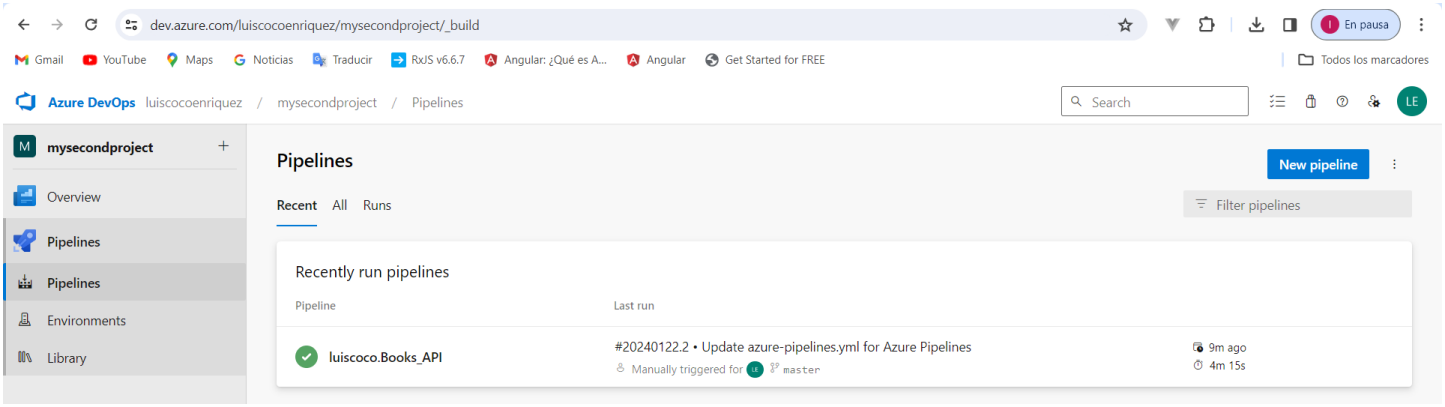


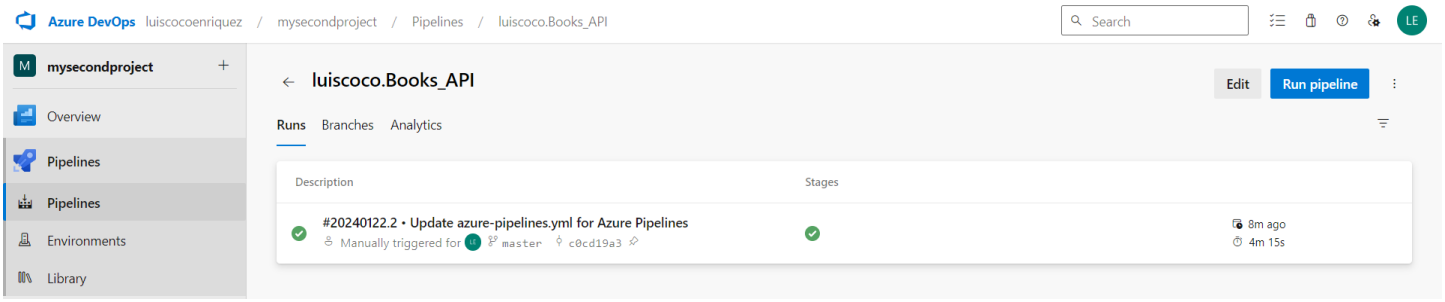
We input the above **yml file source code** to configure the pipeline and we **Run the pipeline**

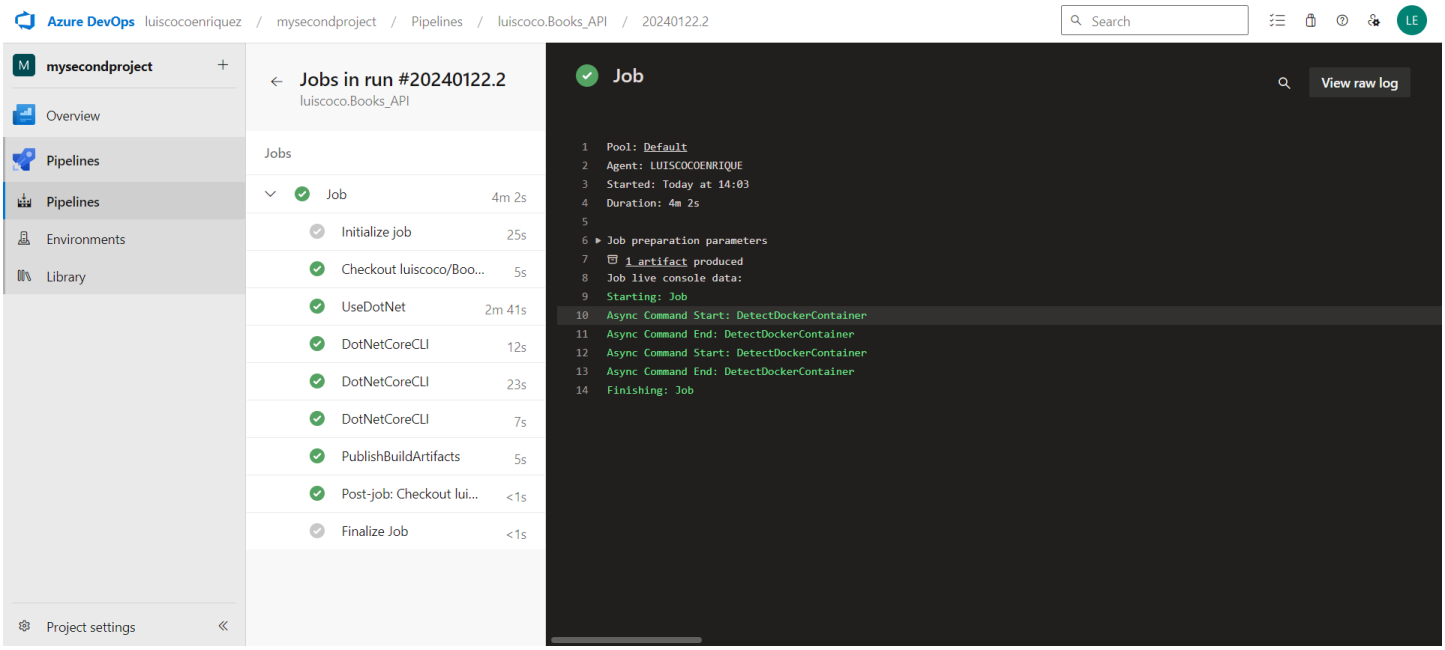


Our jobs will now run on my laptop, where we intalled and setup the **self-hosted agent**

5. Verify the Job successfully run




Azure DevOps luiscocoenriquez / mysecondproject / Pipelines / luiscoco.Books_API
Search
LE
mysecondproject
Overview
Pipelines
Pipelines
Environments
Library
luiscoco.Books_API
Edit
Run pipeline
Runs
Branches
Analytics
Description
Stages
#20240122.2 • Update azure-pipelines.yml for Azure Pipelines
Manually triggered for
master
c0cd19a3
8m ago
4m 15s


Azure DevOps luiscocoenriquez / mysecondproject / Pipelines / luiscoco.Books_API / 20240122.2
Search
LE
mysecondproject
Overview
Pipelines
Pipelines
Environments
Library
Project settings
Jobs in run #20240122.2
luiscoco.Books_API
Jobs
Job 4m 2s
Initialize job 25s
Checkout luiscoco/Boo... 5s
UseDotNet 2m 41s
DotNetCoreCLI 12s
DotNetCoreCLI 23s
DotNetCoreCLI 7s
PublishBuildArtifacts 5s
Post-job: Checkout lui... <1s
Finalize Job <1s
Job
View raw log
1 Pool: Default
2 Agent: LUISCOCOENRIQUE
3 Started: Today at 14:03
4 Duration: 4m 2s
5
6 ▶ Job preparation parameters
7 1 artifact produced
8 Job live console data:
9 Starting: Job
10 Async Command Start: DetectDockerContainer
11 Async Command End: DetectDockerContainer
12 Async Command Start: DetectDockerContainer
13 Async Command End: DetectDockerContainer
14 Finishing: Job