AzureDevops: How to Connect Azure DevOps with GitHub

Connecting Azure DevOps with GitHub involves setting up a connection that allows you to integrate your GitHub repository with Azure DevOps services, such as Azure Pipelines for CI/CD.

Here’s a step-by-step guide on how to connect Azure DevOps with GitHub:

1. Create a New Project in Azure DevOps

We sign in to your Azure DevOps account

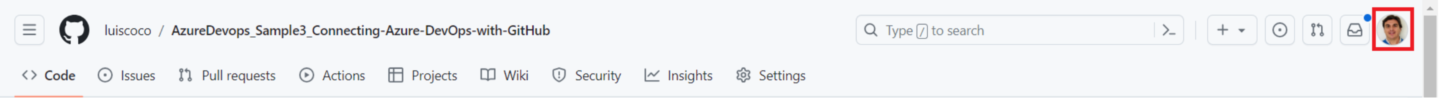
We navigate to the Azure DevOps portal and we create a new project if you haven’t done so already

We provide a name and description for your project

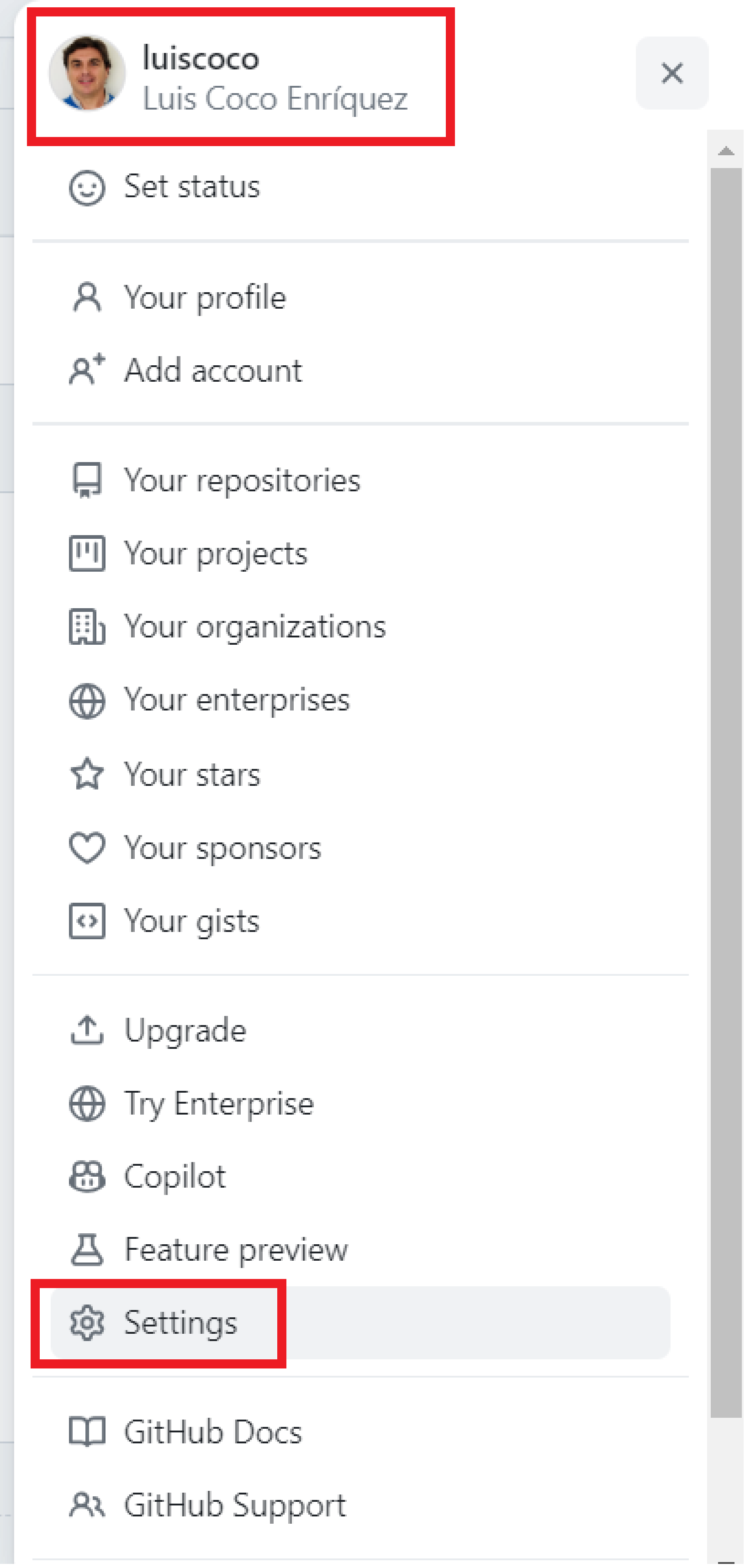
2. Generate GitHub Personal Access Token (PAT)

We go to GitHub and **sign in**

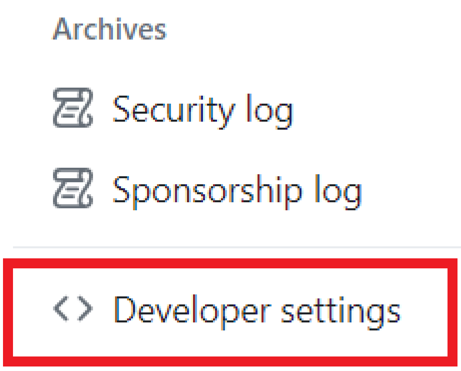
We click on your **profile icon** in the top right corner



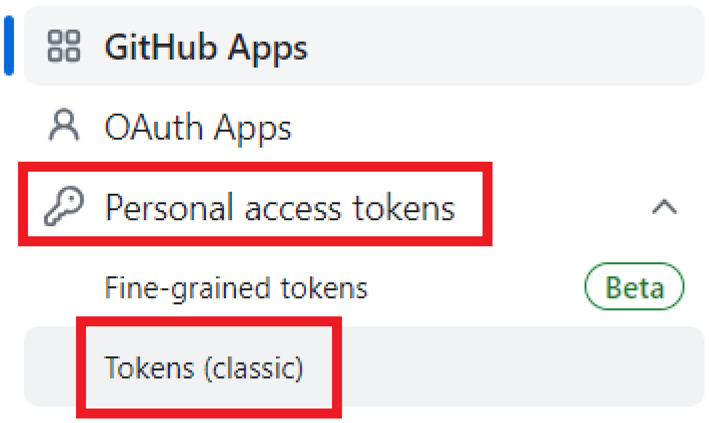
We select the **Settings** option



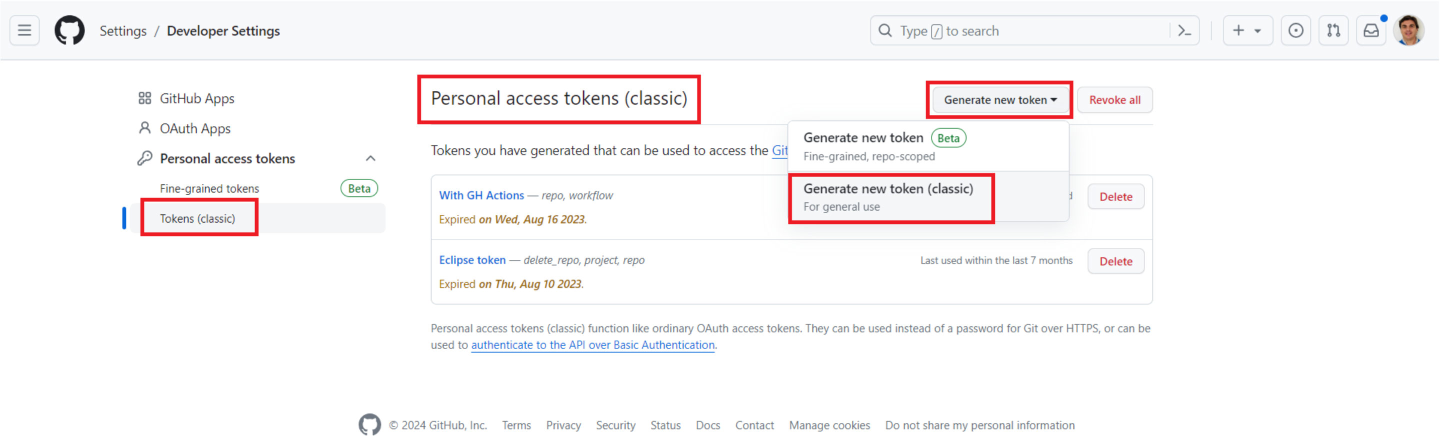
From the sidebar, we select **Developer settings** -> **Personal access tokens**



We click on Generate new token.



We give your token a descriptive name, select the scopes or permissions you want to grant this token (for Azure DevOps, select repo, admin:repo_hook), and then click Generate token



Personal access tokens (classic)

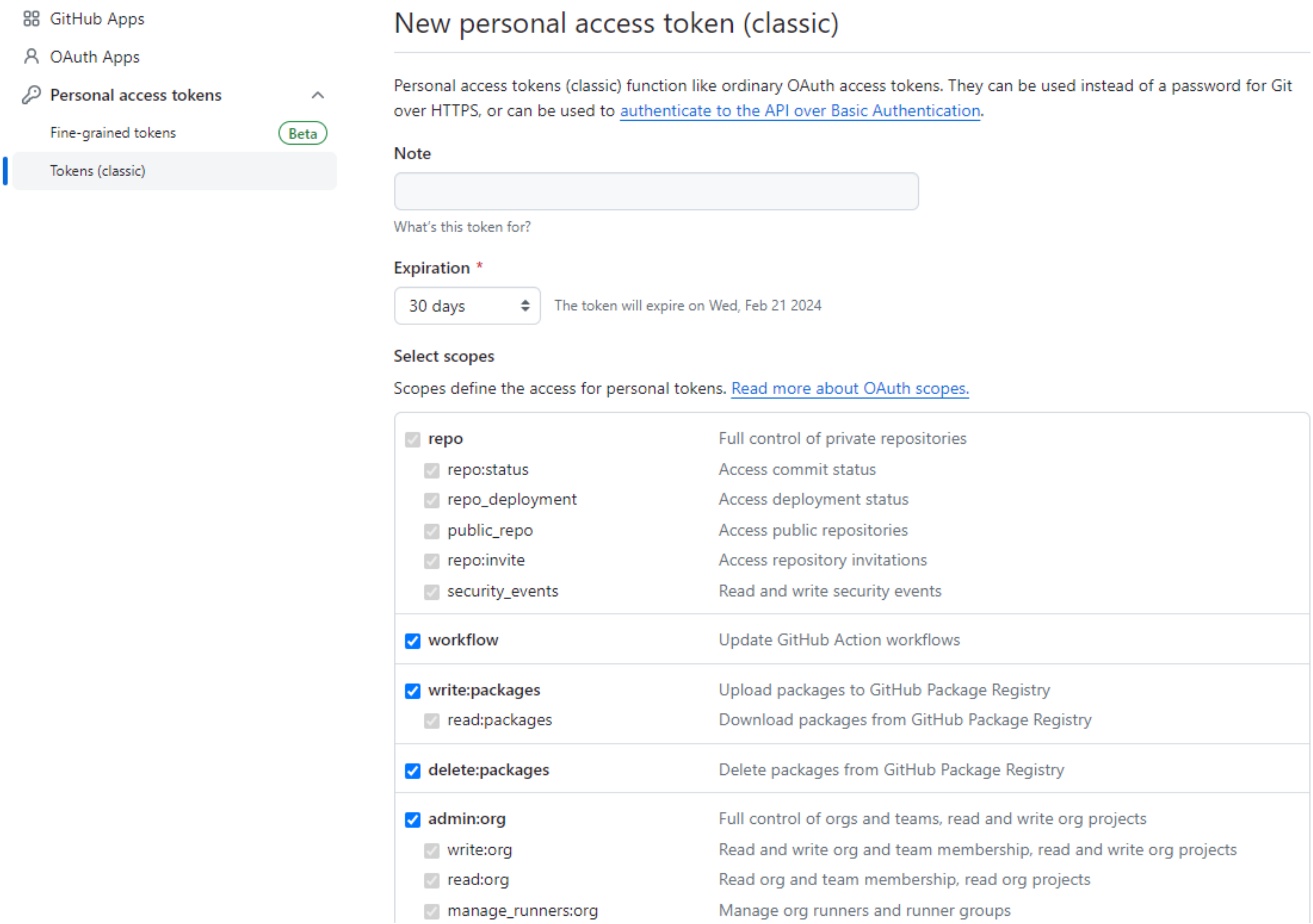
Tokens you have generated that can be used to access the Git

With GH Actions — *repo, workflow*
Expired *on Wed, Aug 16 2023.*

Eclipse token — *delete_repo, project, repo*
Last used within the last 7 months
Expired *on Thu, Aug 10 2023.*

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information



New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

What's this token for?

Expiration *

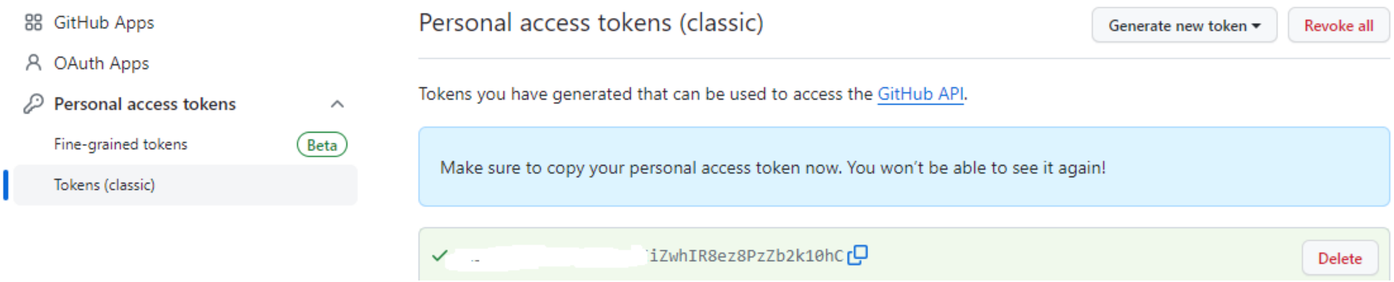
30 days — The token will expire on Wed, Feb 21 2024

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

☑ repo	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events
☑ workflow	Update GitHub Action workflows
☑ write:packages	Upload packages to GitHub Package Registry
☑ read:packages	Download packages from GitHub Package Registry
☑ delete:packages	Delete packages from GitHub Package Registry
☑ admin:org	Full control of orgs and teams, read and write org projects
☑ write:org	Read and write org and team membership, read and write org projects
☑ read:org	Read org and team membership, read org projects
☑ manage_runners:org	Manage org runners and runner groups

We copy the generated **token** and we save it somewhere secure; you won't be able to see it again



Personal access tokens (classic)

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

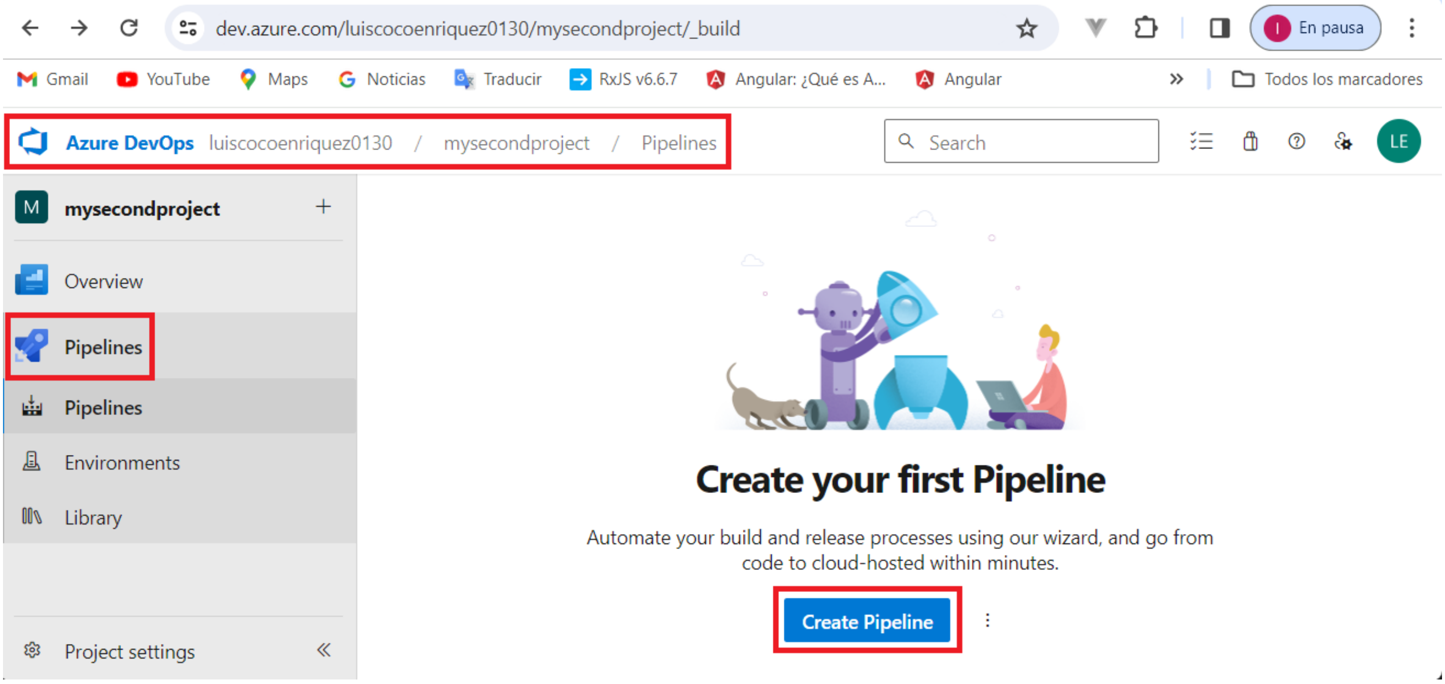
✓ [redacted]iZwhIR8ez8PzZb2k10hC

3. Connect Azure DevOps to GitHub

3.1. Using Azure Pipelines

In Azure DevOps, we go to your project and we select **Pipelines** from the left navigation panel

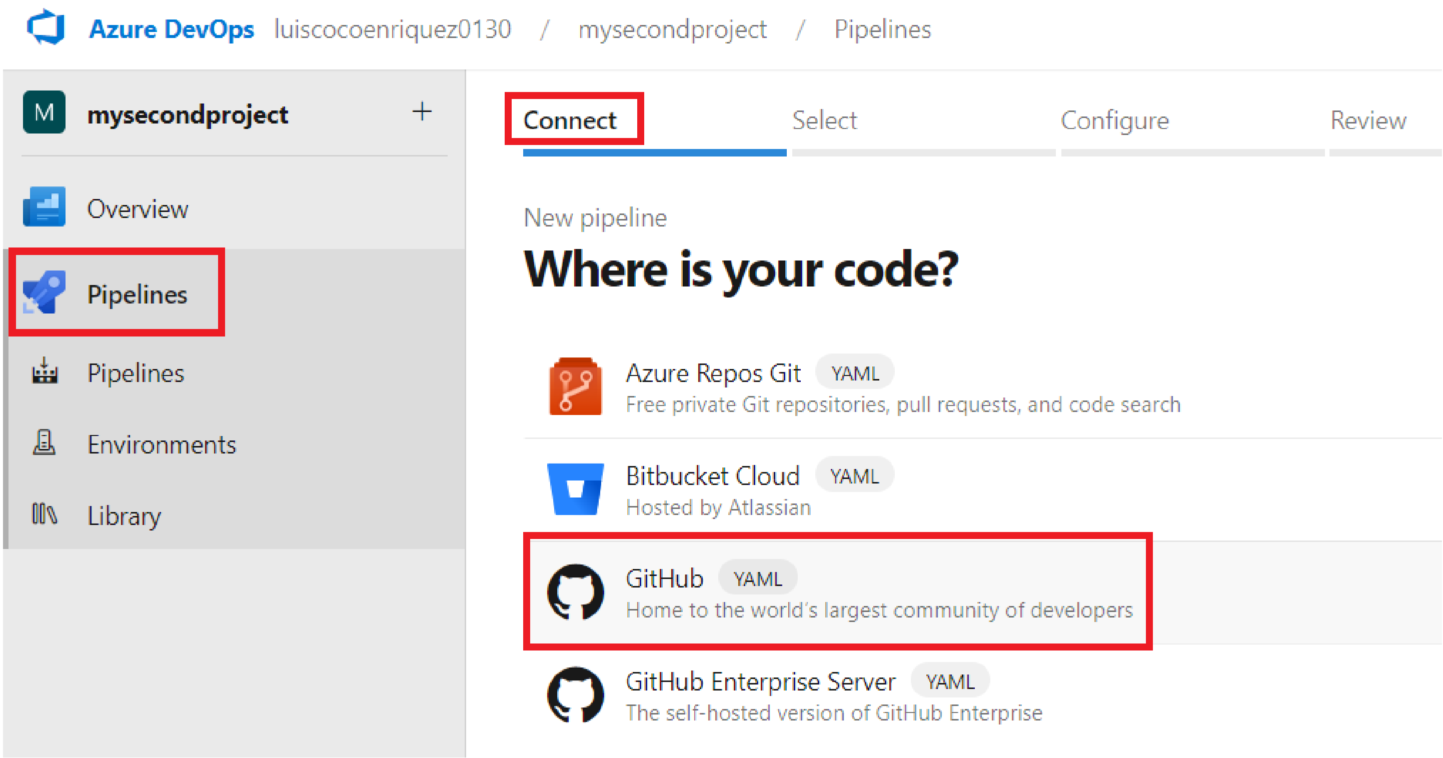
We click on **Create Pipeline**



Create your first Pipeline

Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.

We choose **GitHub** as the **code source**



New pipeline

Where is your code?

Azure Repos Git YAML
Free private Git repositories, pull requests, and code search

Bitbucket Cloud YAML
Hosted by Atlassian

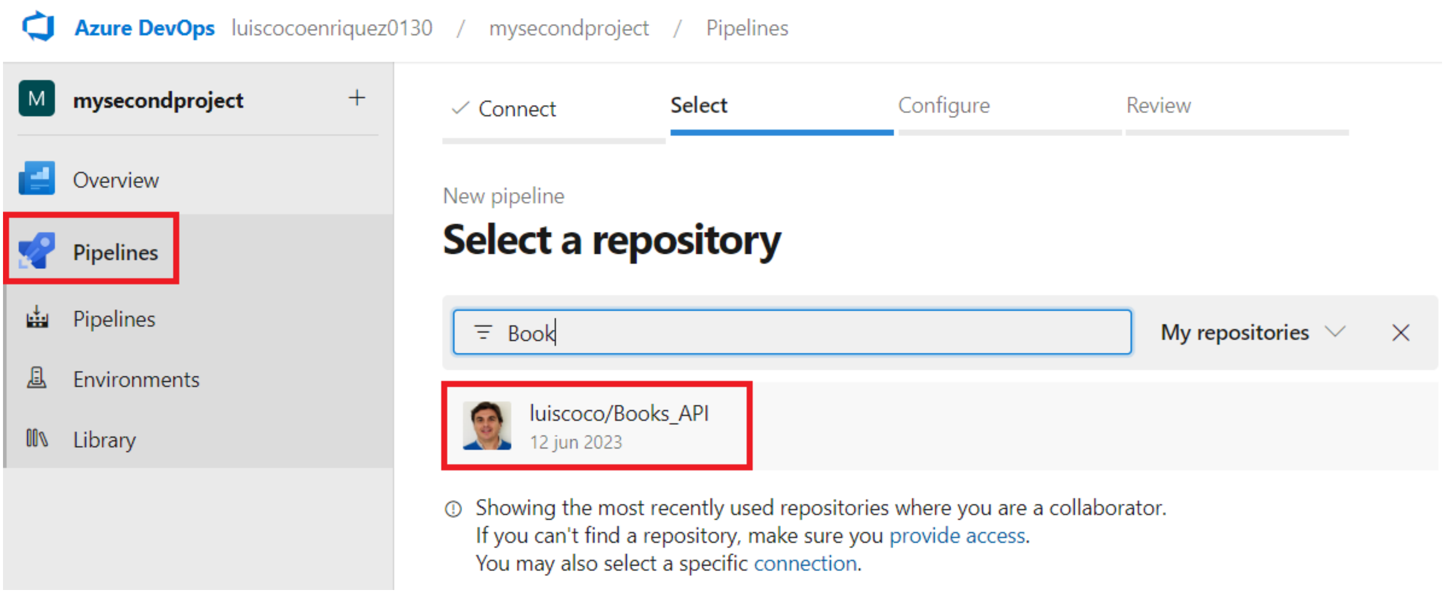
GitHub YAML
Home to the world's largest community of developers

GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise

You may be prompted to sign into GitHub and authorize Azure Pipelines if you haven't already connected your GitHub account to Azure DevOps.

After authorization, select the repository you want to connect to Azure DevOps.

We select the Github repo to connect to



New pipeline

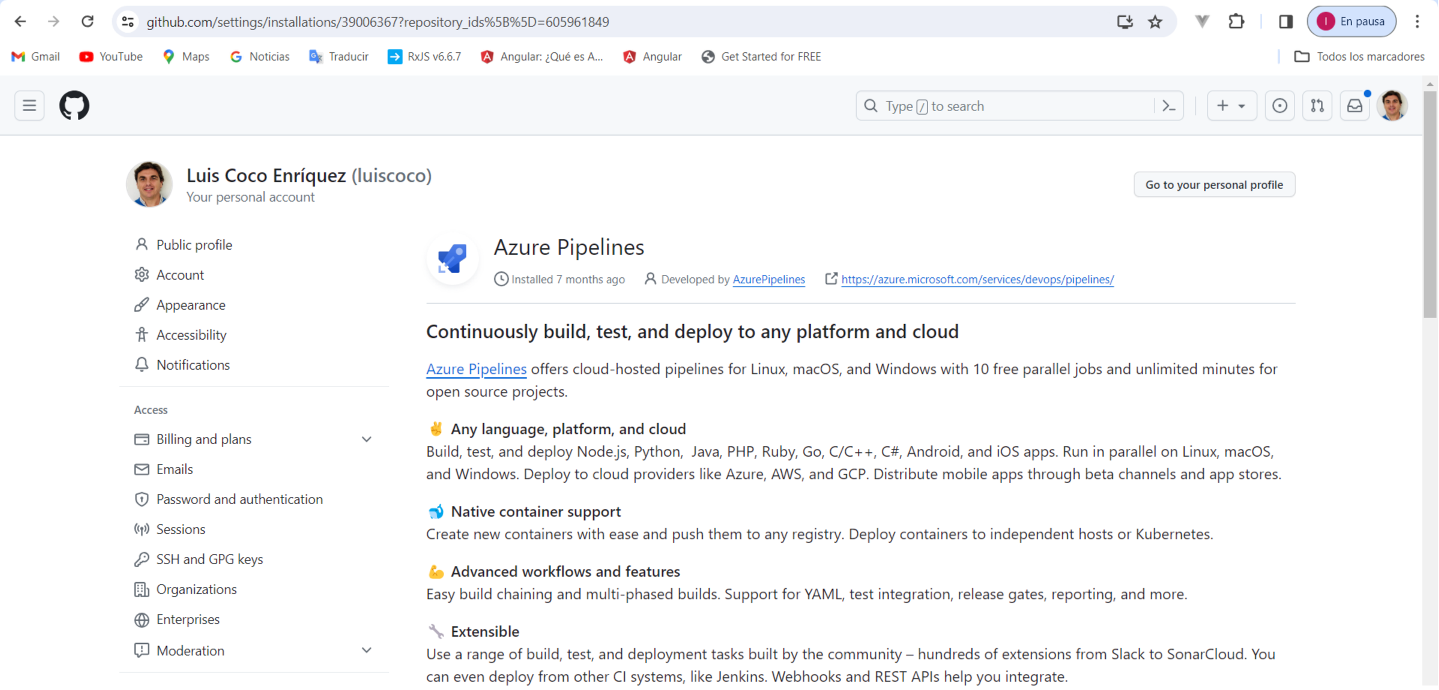
Select a repository

Book

luiscoco/Books_API
12 jun 2023

Showing the most recently used repositories where you are a collaborator.
If you can't find a repository, make sure you provide access.
You may also select a specific connection.

We automatically navigate to the following page



Luis Coco Enríquez (luiscoco)
Your personal account

Azure Pipelines

Installed 7 months ago · Developed by AzurePipelines · https://azure.microsoft.com/services/devops/pipelines/

Continuously build, test, and deploy to any platform and cloud

Azure Pipelines offers cloud-hosted pipelines for Linux, macOS, and Windows with 10 free parallel jobs and unlimited minutes for open source projects.

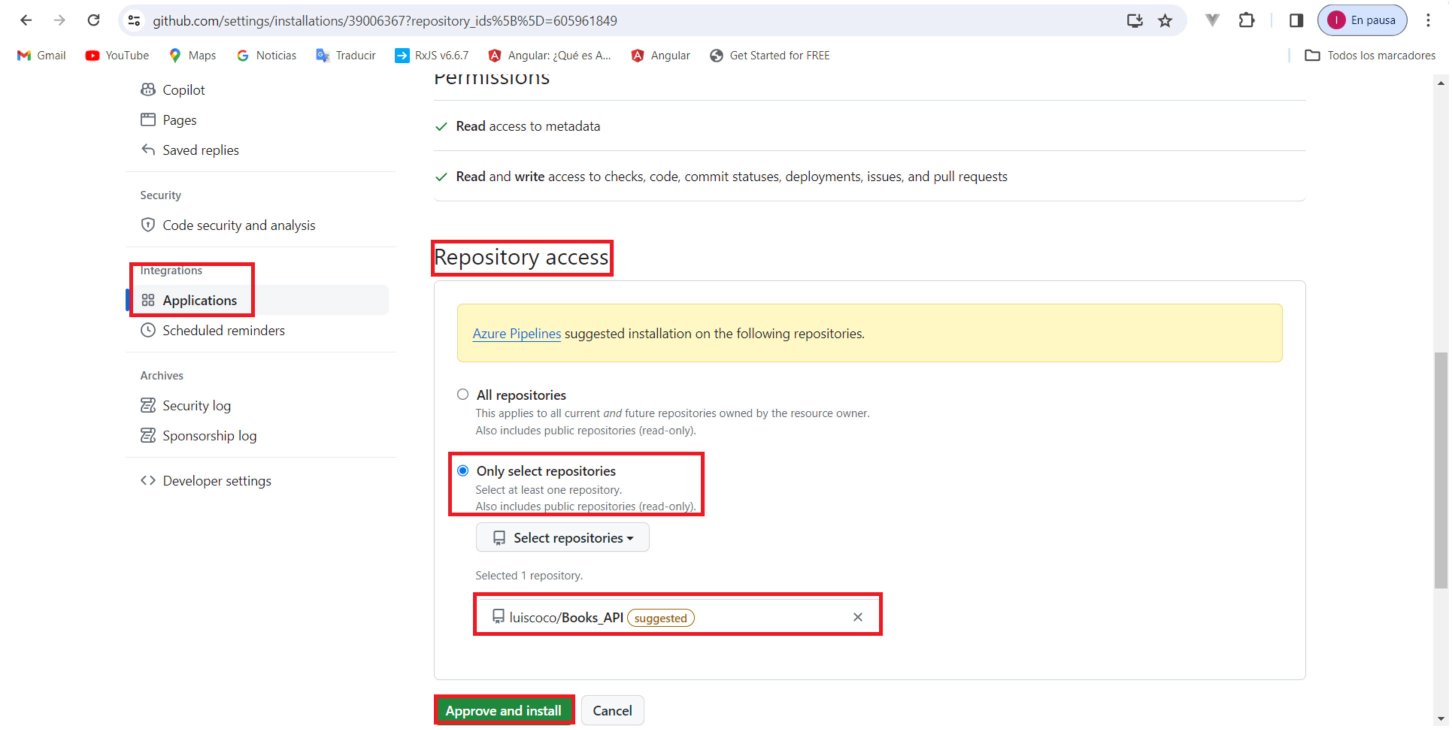
✌ **Any language, platform, and cloud**
Build, test, and deploy Node.js, Python, Java, PHP, Ruby, Go, C/C++, C#, Android, and iOS apps. Run in parallel on Linux, macOS, and Windows. Deploy to cloud providers like Azure, AWS, and GCP. Distribute mobile apps through beta channels and app stores.

🐳 **Native container support**
Create new containers with ease and push them to any registry. Deploy containers to independent hosts or Kubernetes.

💪 **Advanced workflows and features**
Easy build chaining and multi-phased builds. Support for YAML, test integration, release gates, reporting, and more.

🔧 **Extensible**
Use a range of build, test, and deployment tasks built by the community – hundreds of extensions from Slack to SonarCloud. You can even deploy from other CI systems, like Jenkins. Webhooks and REST APIs help you integrate.

We scroll down and select the repo and press the button **Approve and Install**



Permissions

✓ **Read** access to metadata

✓ **Read** and **write** access to checks, code, commit statuses, deployments, issues, and pull requests

Repository access

Azure Pipelines suggested installation on the following repositories.

○ **All repositories**
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

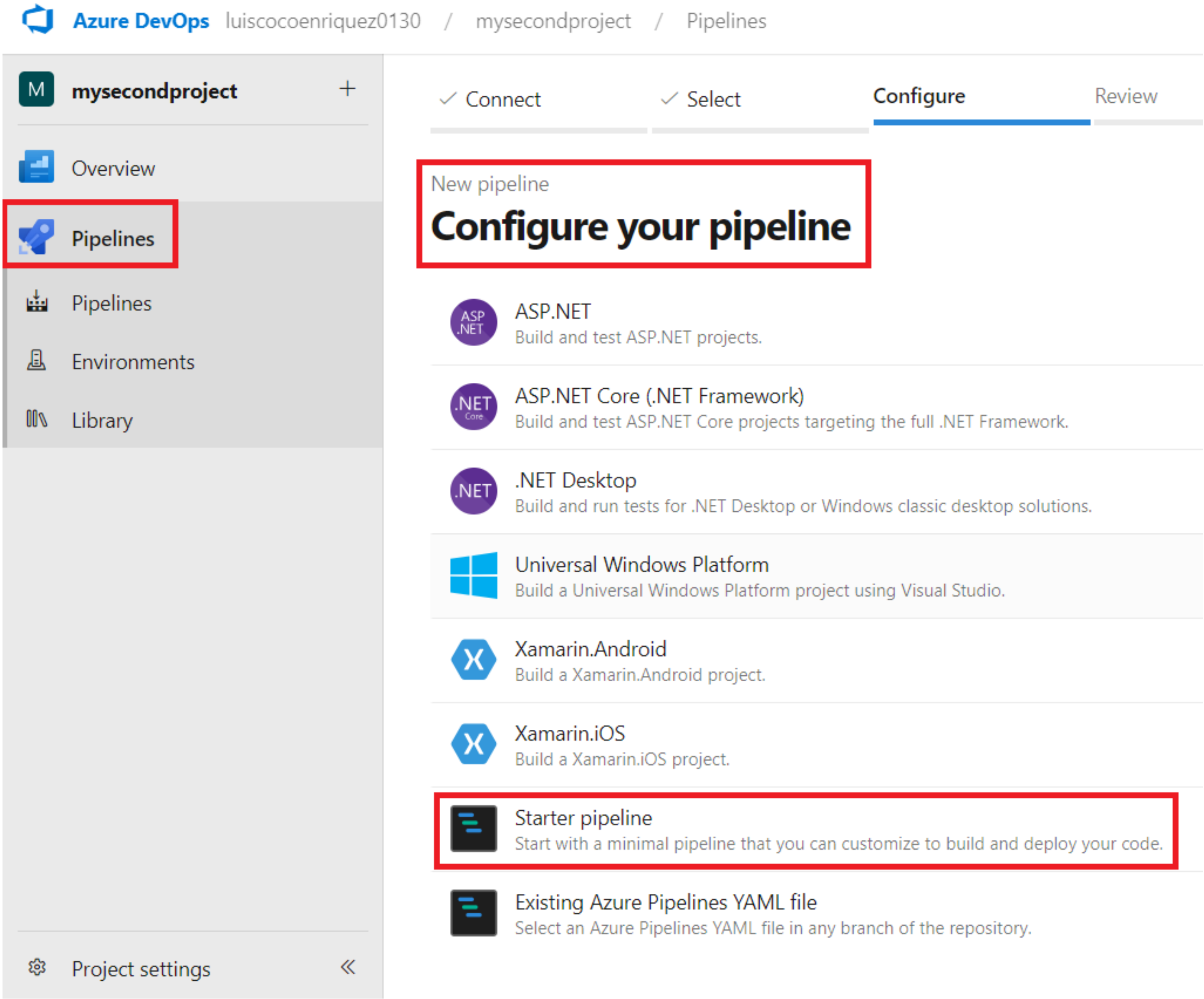
● **Only select repositories**
Select at least one repository.
Also includes public repositories (read-only).

Selected 1 repository.

luiscoco/**Books_API** suggested ×

We are redirected to the Azure DevOps to continue with the PipeLine configuration

We select the **Starter pipeline** option to input the **yaml** file source code



Azure DevOps luiscocoenriquez0130 / mysecondproject / Pipelines

M mysecondproject +

Overview

Pipelines

Pipelines

Environments

Library

Project settings «

✓ Connect ✓ Select Configure Review

New pipeline

Configure your pipeline

ASP.NET
Build and test ASP.NET projects.

ASP.NET Core (.NET Framework)
Build and test ASP.NET Core projects targeting the full .NET Framework.

.NET Desktop
Build and run tests for .NET Desktop or Windows classic desktop solutions.

Universal Windows Platform
Build a Universal Windows Platform project using Visual Studio.

Xamarin.Android
Build a Xamarin.Android project.

Xamarin.iOS
Build a Xamarin.iOS project.

Starter pipeline
Start with a minimal pipeline that you can customize to build and deploy your code.

Existing Azure Pipelines YAML file
Select an Azure Pipelines YAML file in any branch of the repository.

This is the **yaml file** for **Azure Pipeline**

```yaml
trigger:
- main

pool:
  vmImage: 'windows-latest'

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'
```

This is the **main.yml** file for **Github actions**

See the gihub repo: https://github.com/luiscoco/Books_API

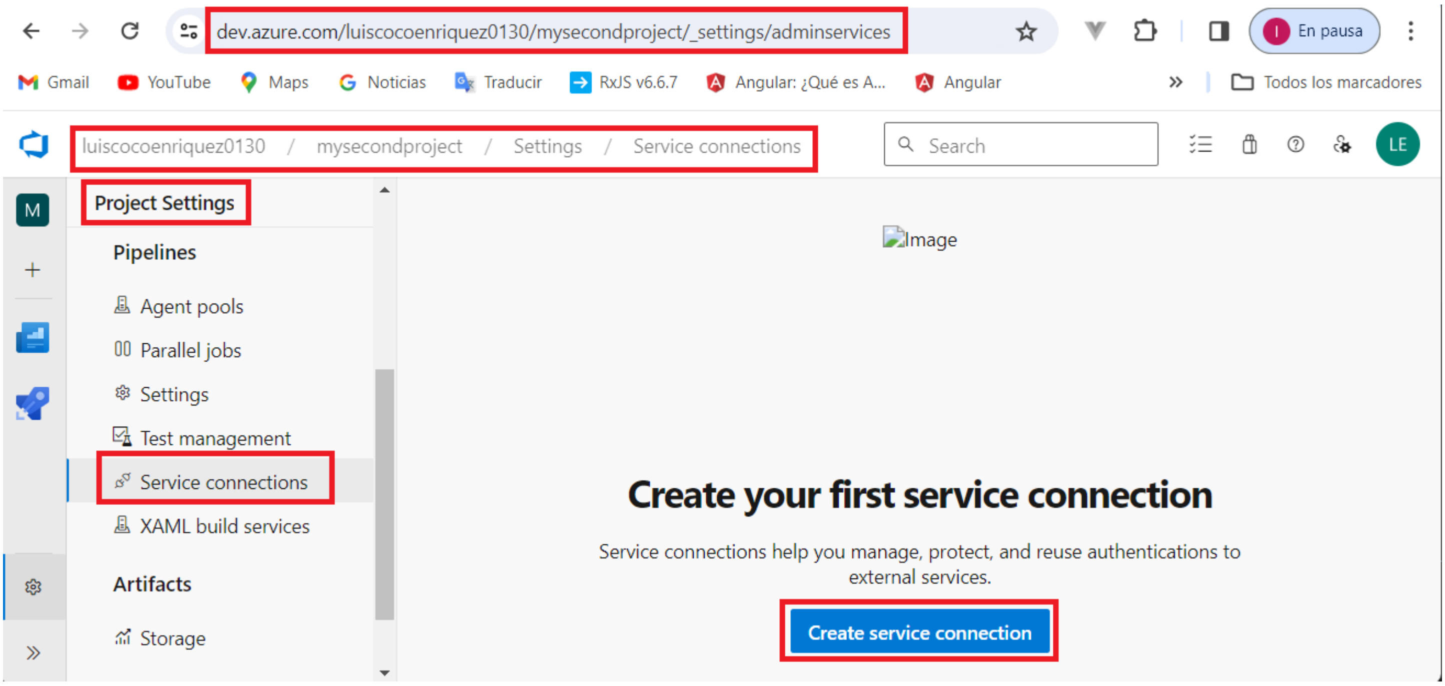
```
name: .NET 8 CI Build

on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

jobs:
  build:

    runs-on: windows-latest

    steps:
    - uses: actions/checkout@v3
      with:
        fetch-depth: 0

    - name: Setup .NET 8
      uses: actions/setup-dotnet@v2
      with:
        dotnet-version: '8.0.x'

    - name: Restore dependencies
      run: dotnet restore

    - name: Build
      run: dotnet build --no-restore -c Release

    # Uncomment the following lines if you have tests
    #- name: Test
    #  run: dotnet test --no-build -c Release --verbosity normal

    - name: Publish
      run: dotnet publish -c Release -o ./publish

    - name: Upload Artifacts
      uses: actions/upload-artifact@v3
      with:
        name: published-app
        path: ./publish
```

3.2. Service Connection Method

We go to **Project settings** in the bottom left corner of your Azure DevOps project

Under **Pipelines**, we select **Service connections**

We click on **New service connection** and choose **GitHub**



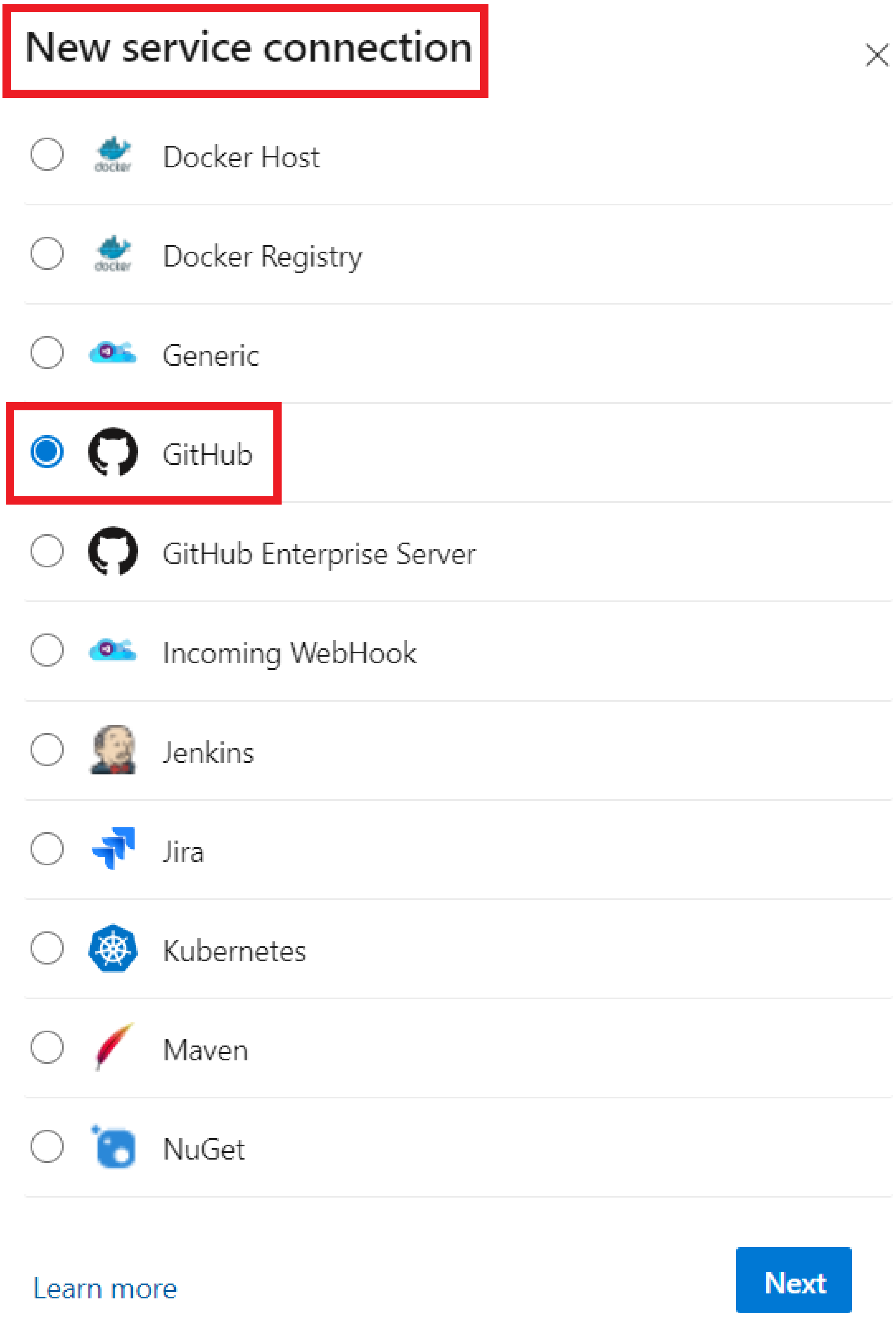
dev.azure.com/luiscocoenriquez0130/mysecondproject/_settings/adminservices

luiscocoenriquez0130 / mysecondproject / Settings / Service connections

Create your first service connection

Service connections help you manage, protect, and reuse authentications to external services.

We choose **GitHub** via **Personal Access Token (PAT)**, we enter the PAT you generated earlier, and we give your connection a name



New service connection

○ Docker Host

○ Docker Registry

○ Generic

● GitHub

○ GitHub Enterprise Server

○ Incoming WebHook

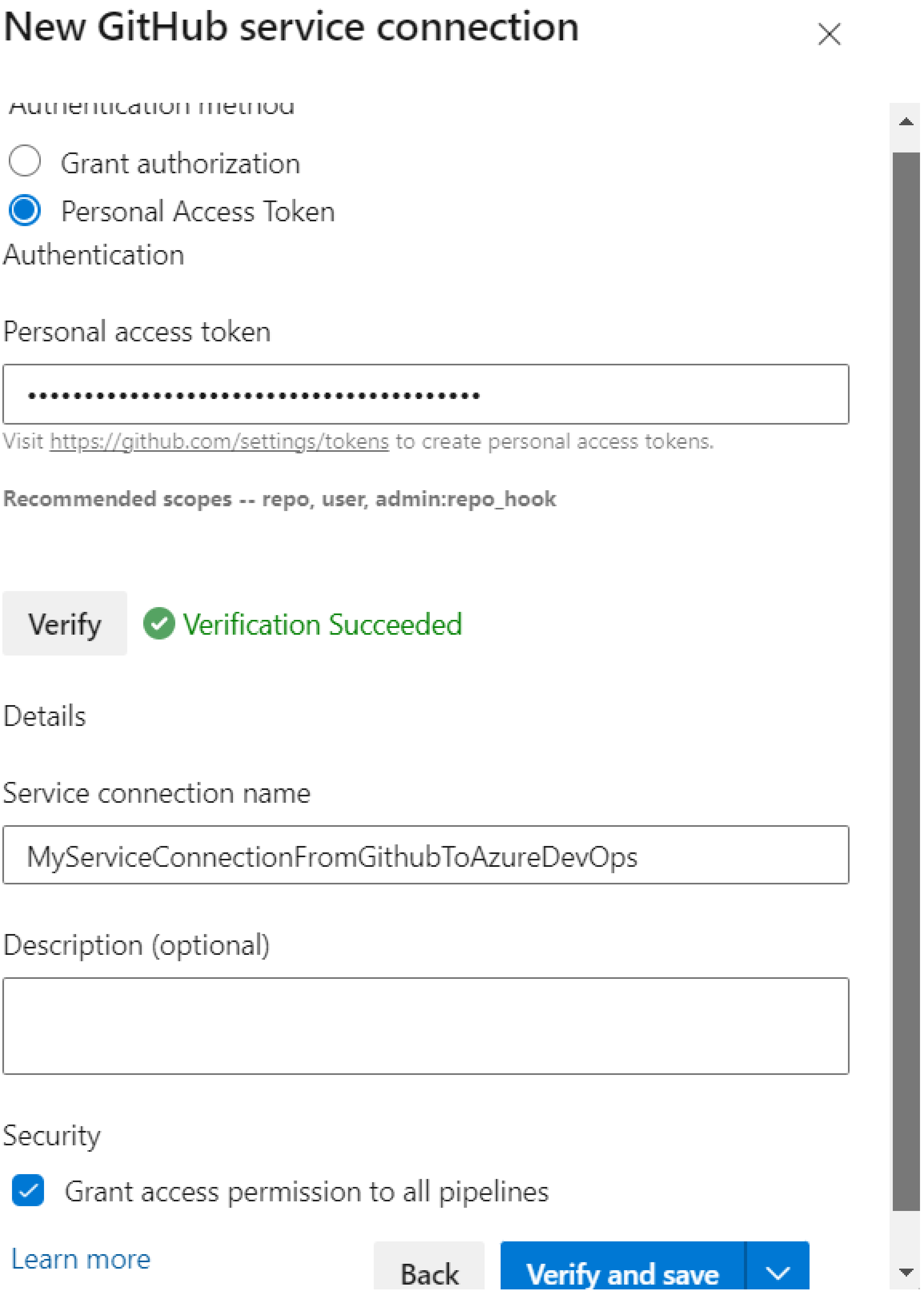
○ Jenkins

○ Jira

○ Kubernetes

○ Maven

○ NuGet



New GitHub service connection

Authentication method

○ Grant authorization

● Personal Access Token

Authentication

Personal access token

••

Visit https://github.com/settings/tokens to create personal access tokens.

Recommended scopes -- repo, user, admin:repo_hook

Verification Succeeded

Details

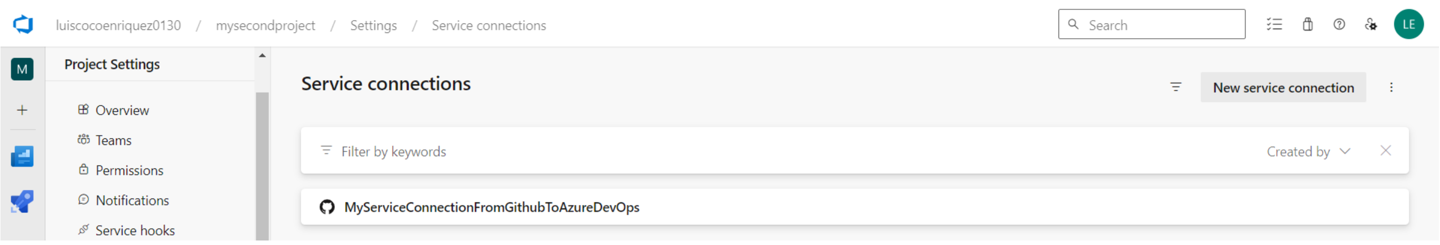
Service connection name

MyServiceConnectionFromGithubToAzureDevOps

Description (optional)

Security

☑ Grant access permission to all pipelines



luiscocoenriquez0130 / mysecondproject / Settings / Service connections

Project Settings

Service connections

MyServiceConnectionFromGithubToAzureDevOps

After we setting up the connection, we can use it in your pipelines to access our GitHub repositories