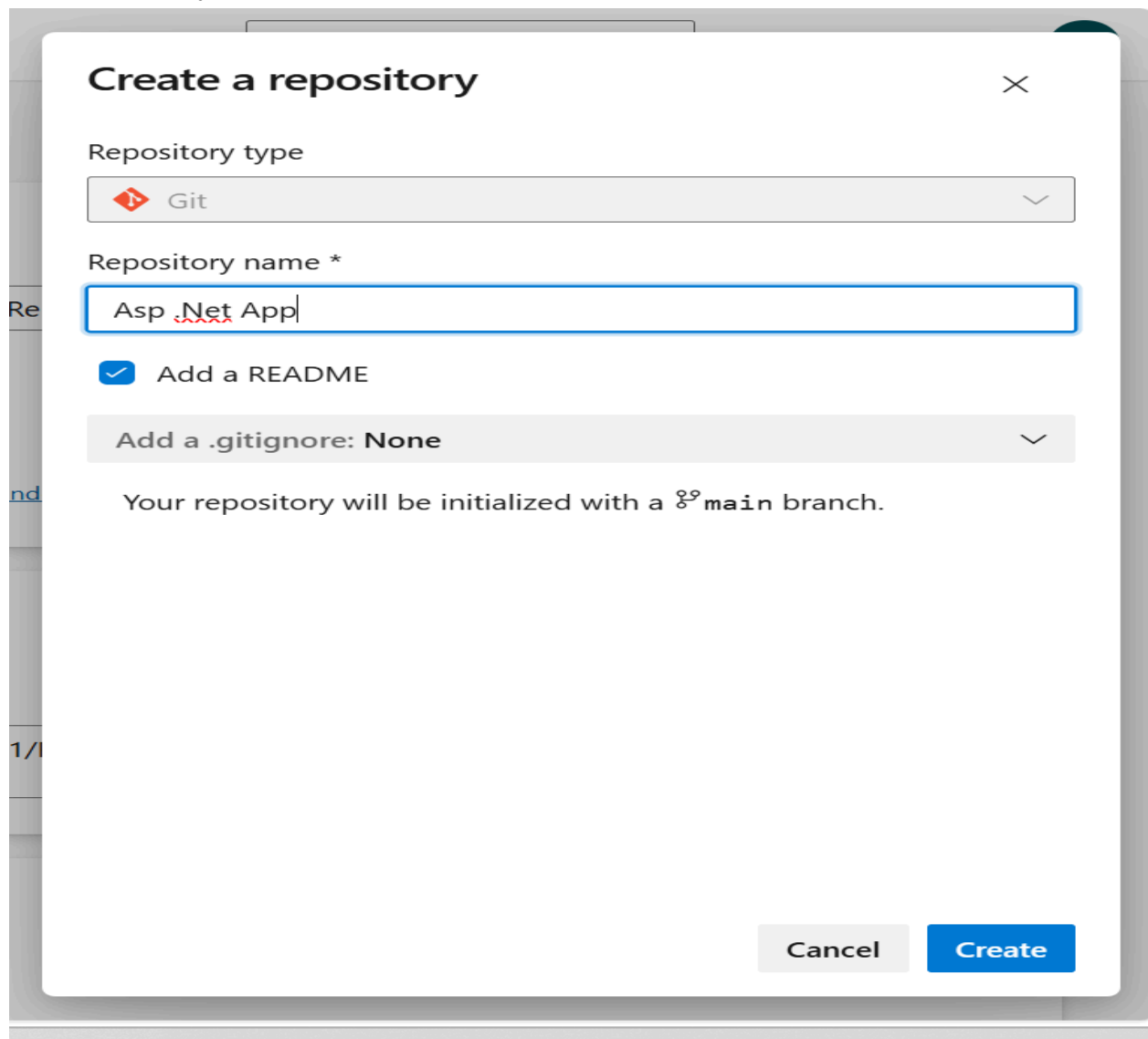



Asp Net App Pipeline on AzureDevops:

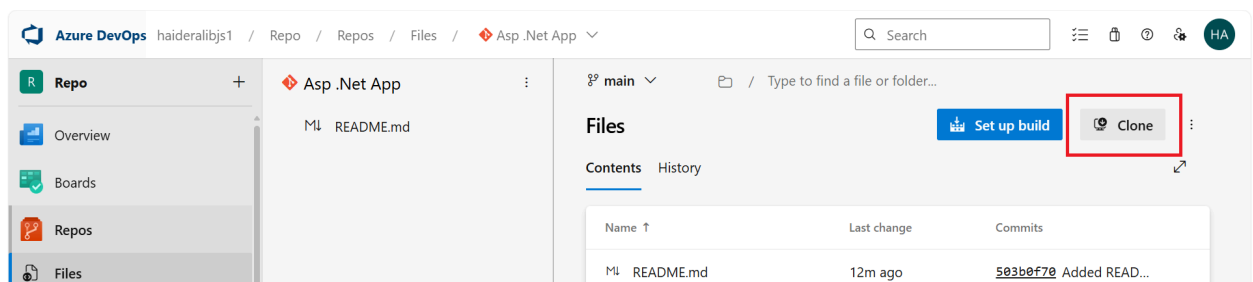
Create repository on AzureDevops



The screenshot shows a modal dialog titled "Create a repository" with a close button (X) in the top right corner. The dialog contains the following elements:

- Repository type:** A dropdown menu showing "Git" with a red Git logo icon and a downward arrow.
- Repository name *:** A text input field containing "Asp .Net App". There is a red squiggly underline under the dot in ".Net".
- Add a README:** A checkbox that is checked, followed by the text "Add a README".
- Add a .gitignore:** A dropdown menu showing "None" with a downward arrow.
- Initialization message:** A line of text stating "Your repository will be initialized with a  main branch."
- Buttons:** At the bottom right, there are two buttons: "Cancel" (disabled, light gray) and "Create" (active, blue).

Click on the Clone button to add this repo on local machine



```
MINGW64:/c/Users/User/Downloads
User@LAPTOP-NS5GPVIA MINGW64 ~/Downloads
$ git clone https://haideralibjs1@dev.azure.com/haideralibjs1/Repo/_git/Asp%20.Net%20App
Cloning into 'Asp%20.Net%20App'...
remote: Azure Repos
remote: Found 3 objects to send. (14 ms)
Unpacking objects: 100% (3/3), 740 bytes | 35.00 KiB/s, done.
```

Create Dockerfile for Asp .Net Core App

```
File Edit Selection View Go Run ...
Asp%20.Net%20App
EXPLORER
ASP%20.NET%20APP
  Dockerfile
  MyWebApp.csproj
  program.cs
  README.md
program.cs U
Dockerfile U
Dockerfile
1 | Stage 1: Build the application
2 | FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
3 | WORKDIR /src
4 |
5 | # Copy project files and restore dependencies
6 | COPY ["MyWebApp.csproj", "./*"]
7 | RUN dotnet restore "MyWebApp.csproj"
8 |
9 | # Copy source code and build
10 | COPY . .
11 | RUN dotnet build "MyWebApp.csproj" -c Release -o /app/build
12 |
13 | # Stage 2: Publish the application
14 | FROM build AS publish
15 | RUN dotnet publish "MyWebApp.csproj" -c Release -o /app/publish /p:UseAppHost=false
16 |
17 | # Stage 3: Create runtime image
18 | FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS runtime
19 | WORKDIR /app
20 | COPY --from=publish /app/publish .
21 |
22 | ENV ASPNETCORE_URLS=http://*:80
23 | EXPOSE 80
24 | ENTRYPOINT ["dotnet", "MyWebApp.dll"]
25 |
```

Push Code to Azure Devops

```
User@LAPTOP-NS5GPVIA MINGW64 ~/Downloads/Asp%20.Net%20App (main)
$ git add .
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'MyWebApp.csproj', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'program.cs', LF will be replaced by CRLF the next time Git touches it
User@LAPTOP-NS5GPVIA MINGW64 ~/Downloads/Asp%20.Net%20App (main)
$ git commit -m "First Commit"
[main 42a6286] First Commit
 3 files changed, 39 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 MyWebApp.csproj
 create mode 100644 program.cs
User@LAPTOP-NS5GPVIA MINGW64 ~/Downloads/Asp%20.Net%20App (main)
$ git branch
* main
User@LAPTOP-NS5GPVIA MINGW64 ~/Downloads/Asp%20.Net%20App (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 989 bytes | 323.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Analyzing objects... (5/5) (4 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (88 ms)
remote: Storing index... done (33 ms)
remote: Updating refs... done (96 ms)
To https://dev.azure.com/haideralibjs1/Repo/_git/Asp%20.Net%20App
 503b0f7..42a6286 main -> main
User@LAPTOP-NS5GPVIA MINGW64 ~/Downloads/Asp%20.Net%20App (main)
$
```

Add Agent: Click on Project Settings

The screenshot shows the Azure DevOps web interface for a project named 'Asp .Net App'. The left sidebar contains a navigation menu with options: Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, Pipelines, Test Plans, Artifacts, and Project settings. The 'Project settings' option is highlighted with a red rectangle. The main content area shows the 'Files' section with a table of files and their commit history.

Name	Last change	Commits
Dockerfile	2m ago	42a6286b First Commit ...
MyWebApp.csproj	2m ago	42a6286b First Commit ...
program.cs	2m ago	42a6286b First Commit ...
README.md	26m ago	503b0f70 Added READ...

Add New Agent pools:




The screenshot shows the 'Agent pools' section of the Azure DevOps interface. The left sidebar is titled 'Project Settings' and contains a navigation menu with options: General (Overview, Teams, Permissions, Notifications, Service hooks, Dashboards), Boards (Project configuration, Team configuration, GitHub connections), Pipelines (Agent pools, Parallel jobs, Settings), and Settings. The 'Agent pools' option is highlighted with a red rectangle. The main content area shows a table of agent pools with columns: Name, Queued jobs, and Running jobs.

Name	Queued jobs	Running jobs
Azure Pipelines		
Default		
devops		

Project Settings
Repo

Agent pools

Security **Add pool**

Name	Queued jobs	Running jobs
 Azure Pipelines Azure Pipelines		
 Default Azure Pipelines		
 devops Haider Ali		

Add New Agent:

Add agent pool

Agent pools are shared across an organization.

Reduce the effort spent in maintaining custom agents by creating a Microsoft managed pool of scalable agents. [Learn more.](#)

☒ **Self-hosted**

Create a pool of custom agents hosted on your own infrastructure for maximum control and flexibility. [Learn more.](#)

☐ **Azure virtual machine scale set**

Create a pool of custom agents based on an Azure Virtual machine scale set hosted in your own Azure subscription. [View configuration instructions.](#)

Name:

devops1

Description (optional):

Markdown supported.

Pipeline permissions:

☒ Grant access permission to all pipelines

Create

Click on Agent pools:

The screenshot shows the 'Agent pools' section of the Azure DevOps Project Settings. On the left, the 'General' tab is selected. The main area displays a table of agent pools:

Name	Queued jobs	Running jobs
Azure Pipelines		
Default		
devops		
devops1		

The 'devops1' pool is highlighted with a red box. The 'Add pool' button is visible in the top right corner.

The screenshot shows the 'devops1' agent pool details page. The 'Agents' tab is selected. The 'New agent' button is highlighted with a red box. The 'Update all agents' button is also visible.

Download Agent:

The screenshot shows the 'Download Agent' page. The 'x64' architecture is selected. The 'Download' button is highlighted with a red box. The page includes the following text:

System prerequisites

Configure your account

Configure your account by following the steps outlined [here](#).

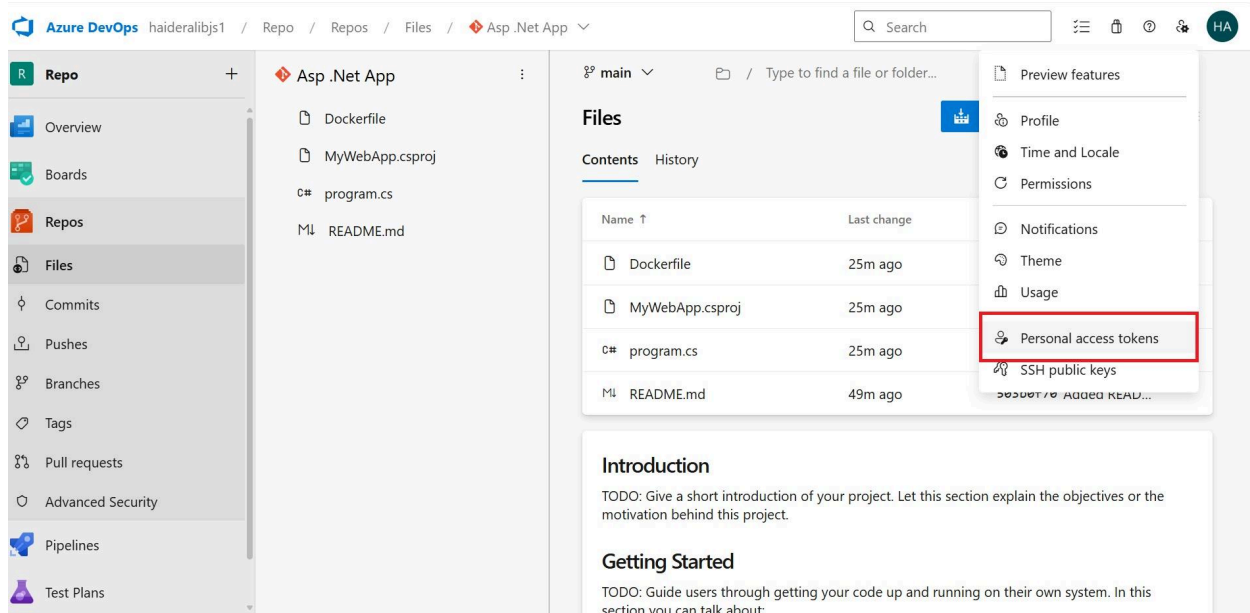
Download the agent

Create the agent

Extract this folder and open in cmd

The screenshot shows a Windows File Explorer window displaying the contents of the 'vsts-agent-win-x64-4.261.0' folder. The folder is located at 'Windows-SSD (C:) > Users > User > Downloads > vsts-agent-win-x64-4.261.0'. The contents include:

Name	Date modified	Type	Size
externals	11/1/2025 10:26 PM	File folder	
bin	11/1/2025 10:25 PM	File folder	
config.cmd	8/25/2025 9:37 AM	Windows Command ...	3 KB
reauth.cmd	8/25/2025 9:37 AM	Windows Command ...	3 KB
run.cmd	8/25/2025 9:37 AM	Windows Command ...	4 KB



Create New Token:

Create a new personal access token

Name

Asp Net App

Organization

haideralibjs1

Expiration (UTC)

Custom defined

10/31/2026

Scopes

Authorize the scope of access associated with this token

Scopes ☒ Full access

☐ Custom defined

Create

Cancel

Add Agent:

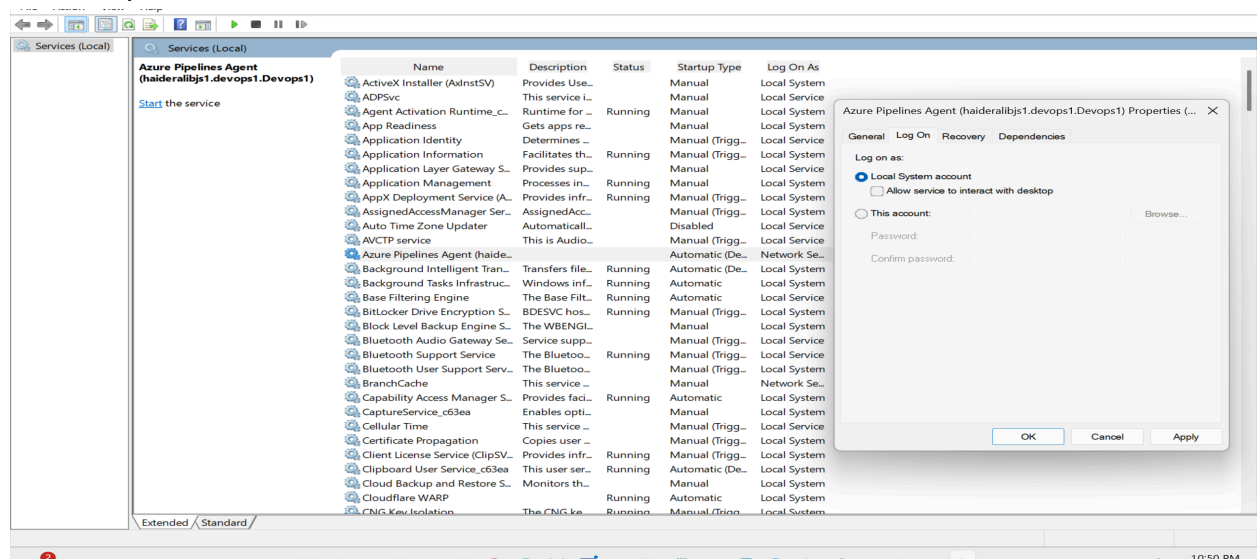
```
Administrator: Command Prompt
C:\Users\User\Downloads\vssts-agent-win-x64-4.261.0> .\config.cmd

Azure Pipelines
agent v4.261.0 (commit 45f3f01)

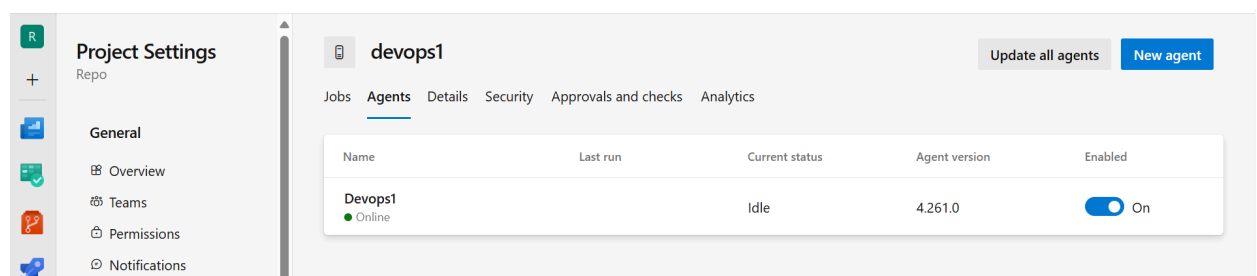
>> Connect:
Enter server URL > https://dev.azure.com/haideralibj1
Enter authentication type (press enter for PAT) >
Enter personal access token > *****
Connecting to server ...

>> Register Agent:
Enter agent pool (press enter for default) > Devops1
Enter agent name (press enter for LAPTOP-NSSGPVIA) > Devops1
Enter replace? (Y/N) (press enter for N) > Y
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
Enter work folder (press enter for work) >
2025-11-01 17:46:03Z: Settings Saved.
Enter run agent as service? (Y/N) (press enter for N) > Y
Enter enable SERVICE_SID_TYPE_UNRESTRICTED for agent service (Y/N) (press enter for N) > Y
Enter User account to use for the service (press enter for NT AUTHORITY\NETWORK SERVICE) >
Granting file permissions to 'NT AUTHORITY\NETWORK SERVICE'.
Service vsstagent.haideralibj1.devops1.Devops1 successfully installed
Service vsstagent.haideralibj1.devops1.Devops1 successfully set recovery option
Service vsstagent.haideralibj1.devops1.Devops1 successfully set delayed auto start
Service vsstagent.haideralibj1.devops1.Devops1 successfully set SID type
Service vsstagent.haideralibj1.devops1.Devops1 successfully configured
Enter whether to prevent service starting immediately after configuration is finished? (Y/N) (press enter for N) > Y
C:\Users\User\Downloads\vssts-agent-win-x64-4.261.0>
```

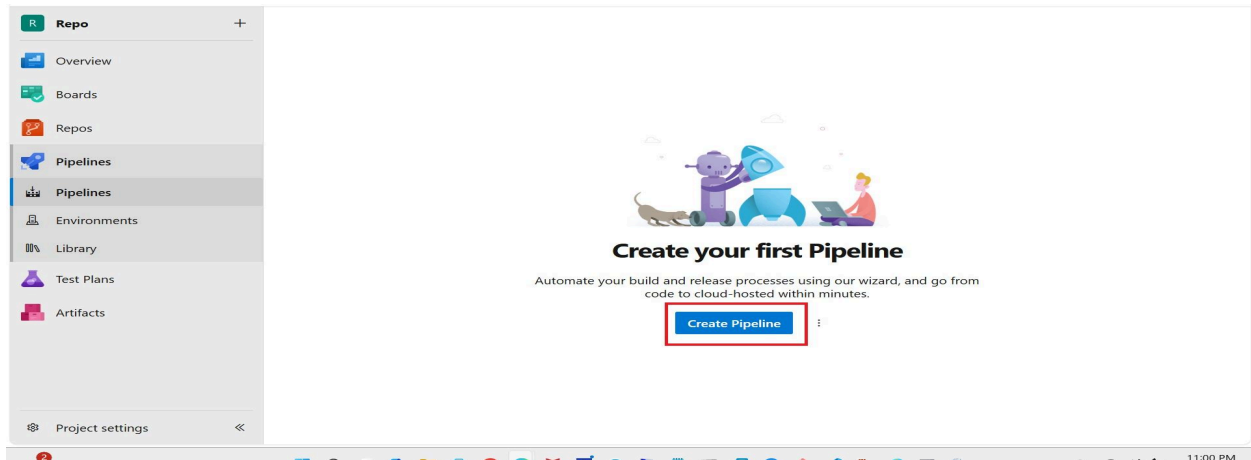
Switch Pipeline user to local on window:



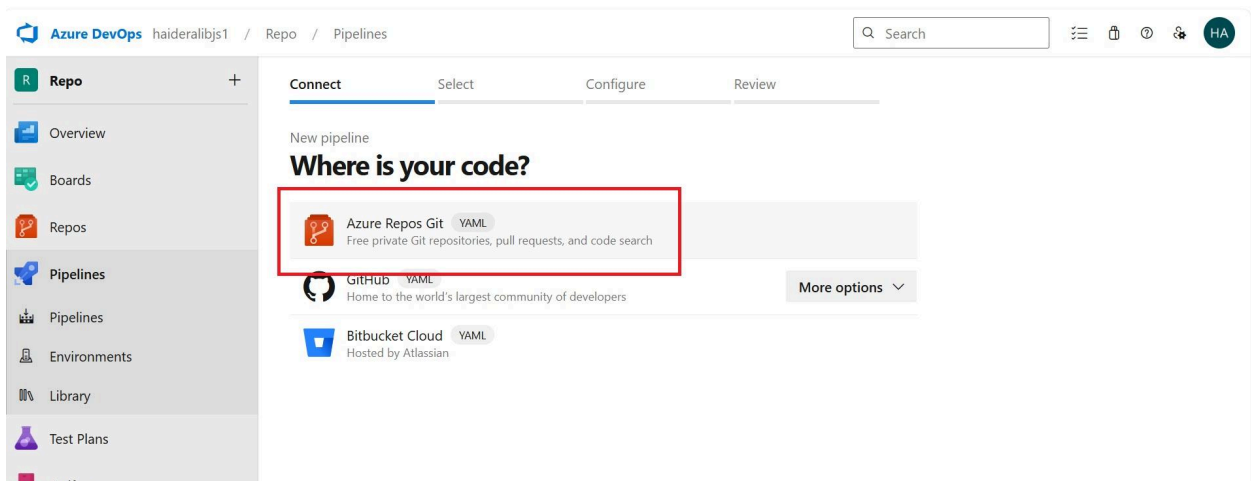
Check Agent is live



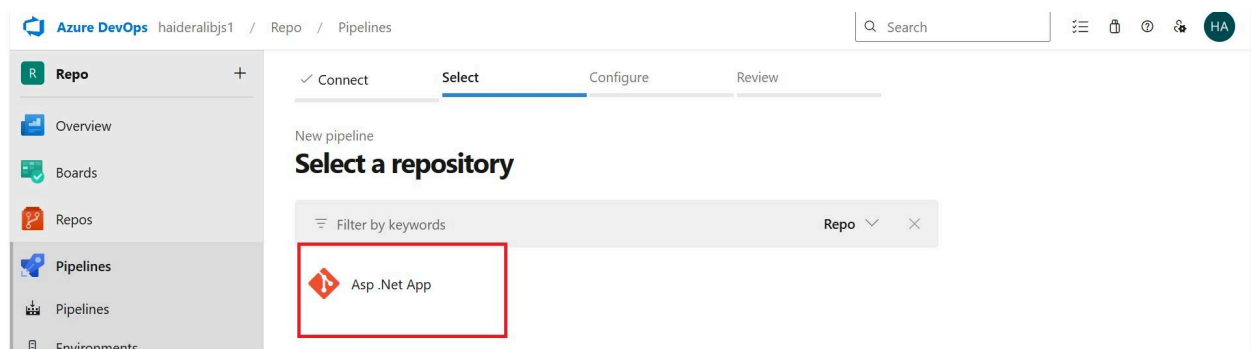
Create Pipeline



Select Azure Repos Git:



Select specific repo:



new pipeline

Configure your pipeline



Docker
Build a Docker image



Docker
Build and push an image to Azure Container Registry



Deploy to Azure Kubernetes Service
Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Service



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.

Show more

Add Pipeline:

Contents History Compare Blame



```
1 trigger:
2 - main # or your branch name
3
4 pool:
5 | name: devops1 # your self-hosted agent pool
6
7 steps:
8 # Step 1: Checkout source
9 - checkout: self
10 | displayName: 'Checkout source code'
11
12 # Step 2: Build Docker image
13 - script: |
14 |   echo "Building Docker image..."
15 |   docker build -t mywebapp .
16 |   displayName: 'Build Docker Image'
17
18 # Step 3: Remove old container if exists & run new one
19 - script: |
20 |   echo "Stopping and removing existing container (if any)..."
21 |   docker rm -f mywebapp_container || echo "No existing container"
22 |   echo "Running new container..."
23 |   docker run -d -p 8080:80 -e ASPNETCORE_URLS=http://+:80 --name mywebapp_container mywebapp
24 |   displayName: 'Run Docker Container'
25
26 # Step 4: Verify container is running
27 - powershell: |
28 |   Write-Host "Listing running containers..."
29 |   docker ps
30 |   Write-Host "Testing application endpoint..."
31 |   Invoke-WebRequest -Uri http://localhost:8080 -UseBasicParsing
32 |   displayName: 'Verify App is Running'
33
34
```

Check Pipeline Run Successfully:

Repo

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Library

Test Plans

Artifacts

← Jobs in run #20251101.5

Asp .Net App

Jobs

Job

1m 34s

Initialize job

22s

Checkout source code

9s

Build Docker Image

43s

Run Docker Container

9s

Verify App is Running

7s

Post-job: Checkout so...

<1s

Finalize Job

<1s

Report build status

<1s

Report build status

View raw log

1 Set build status for commit f63d9d.

Check App Is Working Successfully:

localhost:8080

Hello from .NET 8 Minimal API!