



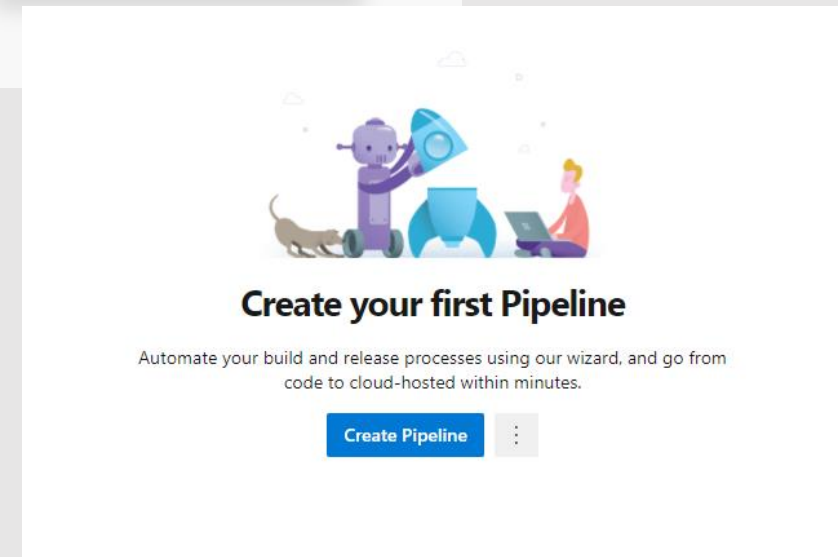
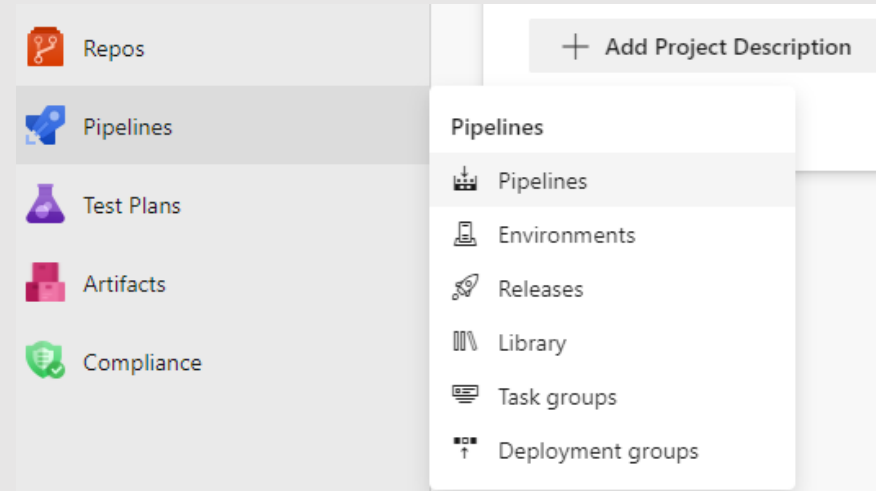
Azure Databricks



Azure DevOps - **Pipelines**

Build Pipeline

- Navigate to the pipelines area within DevOps, create new pipeline



Connect, Select, Configure, Review

Connect


Select


Configure


Review


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

 Other Git
Any generic Git repository

 Subversion
Centralized version control by Apache

Use the classic editor to create a pipeline without YAML

✓ Connect


Select


Configure


Review


New pipeline


Configure your pipeline


 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 blob storage


 .NET Core Function App to Windows
Build a .NET Core function app and deploy it to Windows

 .NET Desktop
Build and run tests for .NET Desktop or Windows

 Android
Build your Android project with Gradle.

 Ant
Build your Java projects and run tests with Ant

 ASP.NET
Build and test ASP.NET projects.

 ASP.NET Core
Build and test ASP.NET Core projects targeting .NET Core

✓ Connect

✓ Select

Configure

Review

New pipeline

Review your pipeline YAML

techt_databricks_cicd / azure-pipelines.yml * 

```
1 # Starter pipeline
2 # Start with a minimal pipeline that you can customize to build and deploy your code.
3 # Add steps that build, run tests, deploy, and more:
4 # https://aka.ms/yaml
5
6 trigger:
7   - main
8
9 pool:
10  - vmImage: 'ubuntu-latest'
11
12 steps:
13   - script: echo Hello, world!
14     displayName: 'Run a one-line script'
15
16   - script: |
17     echo Add other tasks to build, test, and deploy your project.
18     echo See https://aka.ms/yaml
19     displayName: 'Run a multi-line script'
20
```

Adjust Pipeline Trigger

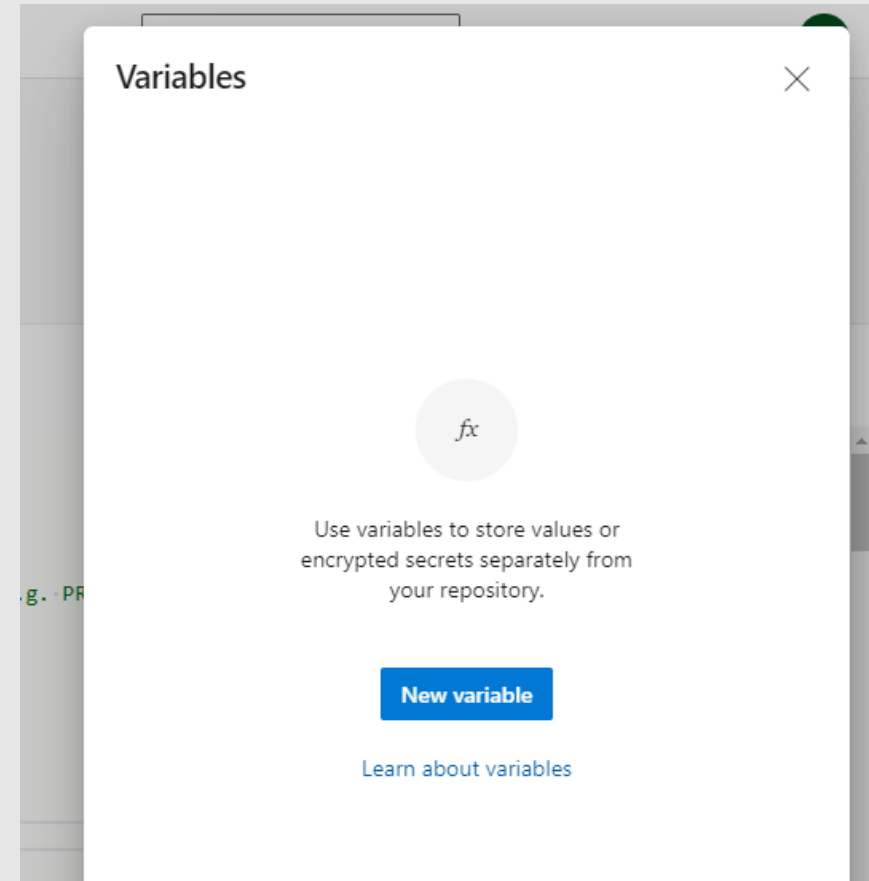
- Change the trigger in .yaml file to to 'none'

```
main ▼ techt_databricks_cicd / azure-pipelines.yml

1  # Starter pipeline
2  # Start with a minimal pipeline that you can customize to build and deploy your code.
3  # Add steps that build, run tests, deploy, and more:
4  # https://aka.ms/yaml
5
6  # the selected branch that will initiate the test
7  trigger: none
8
9  # where this test is going to run -- can choose Ubuntu
10 pool:
11   vmImage: 'windows-latest'
12
13 # Other options available:
14 # vmImage: 'ubuntu-latest'
15 # vmImage: 'macOS-latest'
16
17 # steps to run in this test
18 steps:
19   - script: echo Hello, world!
20     displayName: 'Run a one-line script'
21
22   - script: |
23     echo Add other tasks to build, test, and deploy your project.
24     echo See https://aka.ms/yaml
25     displayName: 'Run a multi-line script'
26
```

Environment Variables

- Can utilise variables to store encryption secrets and passwords separately to the repository



Save and Commit CI

- Give your commit a message and description for visibility
- Due to main branch protection policies (needing additional reviewers, if specified), you will need to 'Create a new branch for this commit' – then create a pull request to merge in to the main branch for use

Save

Saving will commit azure-pipelines.yml to the repository.

Commit message

Set up CI with Azure Pipelines

Optional extended description

Add an optional description...

☒ Commit directly to the main branch

☐ Create a new branch for this commit

Enforce the CI test script

- Project settings > Repos: Repositories > Select your repository
- Policies > Branch Policies > Select main branch
- Build Validation > '+' > Select the pipeline we just created
- Trigger = automatic
- Policy requirement = required
- Build Expiration = after 12 hours
- Display Name = you choose
- Ensure to have these options selected to ensure the pull requests do not merge until a successful build has been completed – this will trigger your CI tests every time a Pull Request is created against the main branch

The image shows a GitHub interface with two overlapping panels. The top panel, titled 'All Repositories', has tabs for 'Repositories', 'Settings', 'Policies', and 'Permissions'. The 'Repositories' tab is active, showing a list with 'techt_databricks_cicd'. The bottom panel, titled 'tech_databricks_cicd', has tabs for 'Settings', 'Policies', and 'Permissions'. The 'Policies' tab is active, showing the 'Add build policy' dialog. The dialog has a close button (X) in the top right. It contains the following fields and options:

- Build pipeline ***: A dropdown menu with 'techt_databricks_cicd' selected.
- Path filter (optional)**: An empty text input field with a help icon (i) on the right.
- Trigger**: Two radio button options: 'Automatic (whenever the source branch is updated)' (selected) and 'Manual'.
- Policy requirement**: Two radio button options: 'Required' (selected, with subtext 'Build must succeed in order to complete pull requests.') and 'Optional' (with subtext 'Build failure will not block completion of pull requests.').
- Build expiration**: Three radio button options: 'Immediately when all default branches is updated', 'After hours if all default branches has been updated' (selected), and 'Never'.
- Display name**: A text input field with 'CI Pipeline before PR Completion' entered.

The bottom panel also shows a 'Branch Policies' section with the subtitle 'Protect important branch namespaces in this repository with pre-merge checks and policies'. It lists two policies: 'jo_new_branch' and 'main'. The 'main' policy is selected, showing 'Default' and 'Compare' buttons.