



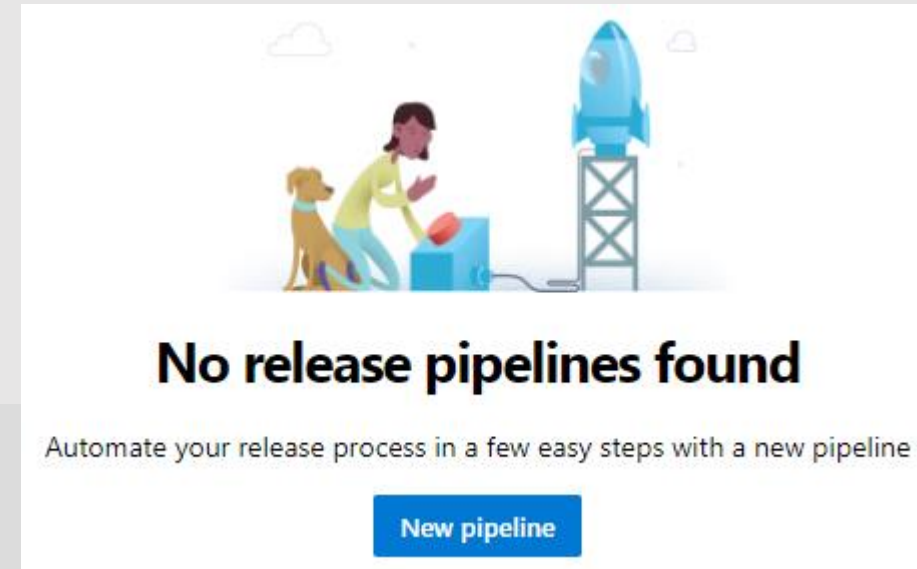
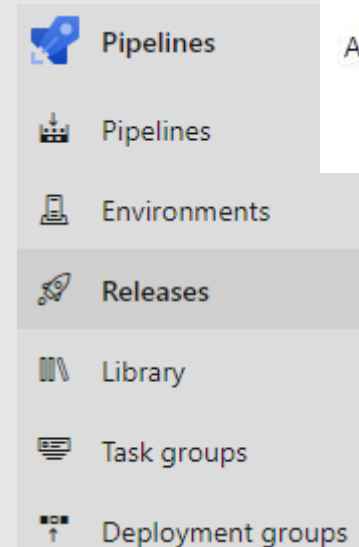
Azure Databricks



Azure DevOps - Releases

# Create new Release Pipeline

- Go to Pipelines > Releases > New Pipeline
- Start with an 'Empty job'



Select a template

Or start with an [Empty job](#)

## Featured



### Azure App Service deployment

Deploy your application to Azure App Service. Choose from Web App on Windows, Linux, containers, Function Apps, or WebJobs.



### Deploy a Java app to Azure App Service

Deploy a Java application to an Azure Web App.



### Deploy a Node.js app to Azure App Service

Deploy a Node.js application to an Azure Web App.



### Deploy a PHP app to Azure App Service and Azure Database for MySQL

Deploy a PHP application to an Azure Web App and database to Azure Database for MySQL.

# Add an Artifact

- Click little 'x' in top right
- “Add an artifact”
- Source type: Azure Repo
- Choose repository and the branch that has the Notebooks you want to deploy to Databricks created

All pipelines > New release pipeline

Pipeline Tasks Variables Retention Options History

Artifacts | Add

Add an artifact

Schedule not set

Stages | Add

Add an artifact

Source type

Build ☒ Azure Re... GitHub TFVC

5 more artifact types

Project \*

Source (repository) \*

Default branch \*

Default version \*

☐ Checkout submodules

☐ Checkout files from LFS

Shallow fetch depth

Source alias \*

# Configure pipeline trigger

- We want to trigger the release pipeline when a pull request to the main branch is completed
- Lightning bolt
- Pull request trigger activates every time a PR is created in the whole repo – not suitable in this scenario

The screenshot displays the configuration interface for a pipeline trigger in Azure DevOps. It is divided into two main sections: 'Artifacts' and 'Stages'.

**Artifacts:** This section contains a trigger for the repository '\_techt\_databricks\_cicd\_REPO'. It features a lightning bolt icon, indicating a pull request trigger, and a 'Schedule not set' option with a clock icon.

**Stages:** This section contains two triggers:

- Continuous deployment trigger:** This trigger is 'Enabled' (indicated by a blue toggle switch). It creates a release every time a Git push occurs in the selected repository. It has a 'Branch filters' section with a table showing the trigger type as 'Include' and the branch as 'main'.
- Pull request trigger:** This trigger is 'Disabled' (indicated by a grey toggle switch). It creates a release every time a selected artifact is available as part of a pull request workflow. A note below it states: 'Enabling this will create a release every time a selected artifact is available as part of a pull request workflow'.

# Add tasks to Stage object

- Add the following tasks to the job → Use Python, Configure Databricks CLI & Deploy Notebooks to Workspace
- Will need to go in to each and add the configurations

The screenshot displays the Databricks workspace interface. At the top, the 'Stages' tab is active, showing a diagram of 'Stage 1' with a link to '1 job, 0 task' and a 'View stage tasks' button. Below this, the 'Tasks' tab is selected for 'Stage 1' (Deployment process). The task list includes:

- Agent job** (Run on agent) with a plus icon to add more jobs.
- Use Python 3.x** (Use Python version) with the Python logo icon.
- Configure Databricks CLI** (Some settings need attention) with the Databricks logo icon and a warning icon.
- Deploy Notebooks to Workspace** (Some settings need attention) with the Databricks logo icon and a warning icon.

# Access Configurations

- Databricks CLI task:
  - Will need the workspace URL → <workspace-ID>.azuredatabricks.net
  - Access token → User settings > generate new token
  - Databricks Access key disappears forever once created
- Deploy Notebook task:
  - Define where the notebooks are within Azure Repo
  - Define where in workspace the notebooks will be deployed

## User Settings

[Access Tokens](#) [Git Integration](#) [Notebook Settings](#)

Personal access tokens can be used for secure authentication to the [Databricks API](#) instead of passwords.

[Generate New Token](#)

Deploy Databricks Notebooks ⓘ

[View YAML](#) [Remove](#)

Task version

0.\* ▼

Display name \*

Deploy Notebooks to Workspace

Notebooks folder \* ⓘ

\$(System.DefaultWorkingDirectory)/\_techt\_databricks\_cicd/databricks\_notebooks ...

Workspace folder \* ⓘ

/Shared/workspace\_notebooks

Control Options ▼

Output Variables ▼

# Create Release

- Can create release manually
- Or
- Create release from the next successfully completed pull request to 'main' ← *[I would suggest this option personally]*

## Create a new release

New release pipeline

⚡ Pipeline ^  
Click on a stage to change its trigger from automated to manual.

⚡ Stage 1

Stages for a trigger change from automated to manual. ⓘ

📦 Artifacts ^  
Select the version for the artifact sources for this release

Source alias	Version
_techt_databricks_cicd_REPO	40466964 (Merged PR 5: Commit to test r... ▾)

Release description

# Create Pull Request

- Make your changes within your notebook on Databricks
- Commit to DevOps Repo
- Create a new Pull Request
- Watch your CI test complete (*hopefully successfully*)
- Once finished, click 'Complete' button in top right to complete PR
- Watch your Release pipeline complete (*hopefully successfully*)
- Go to Databricks Shared workspace to confirm it worked

