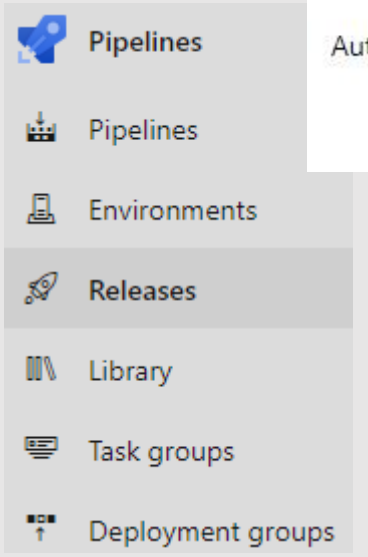
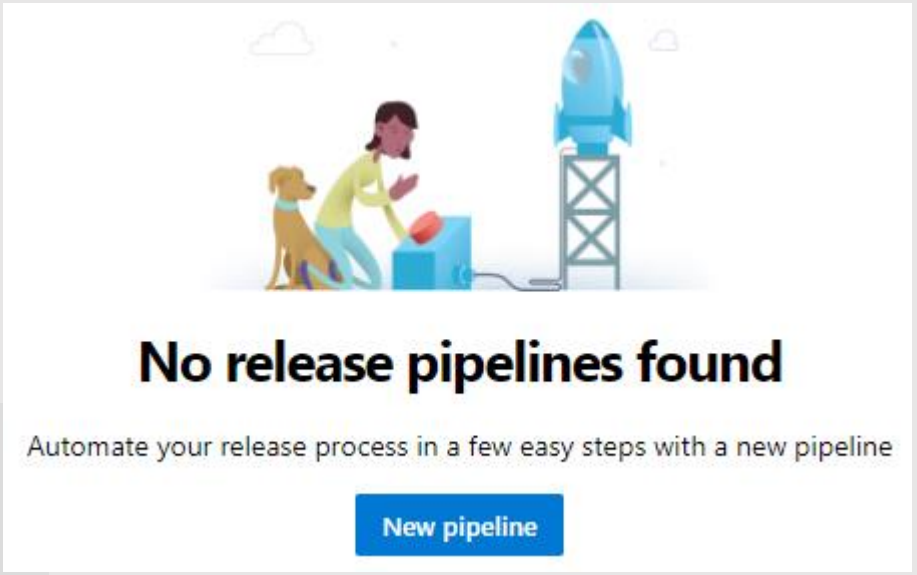
Azure Databricks

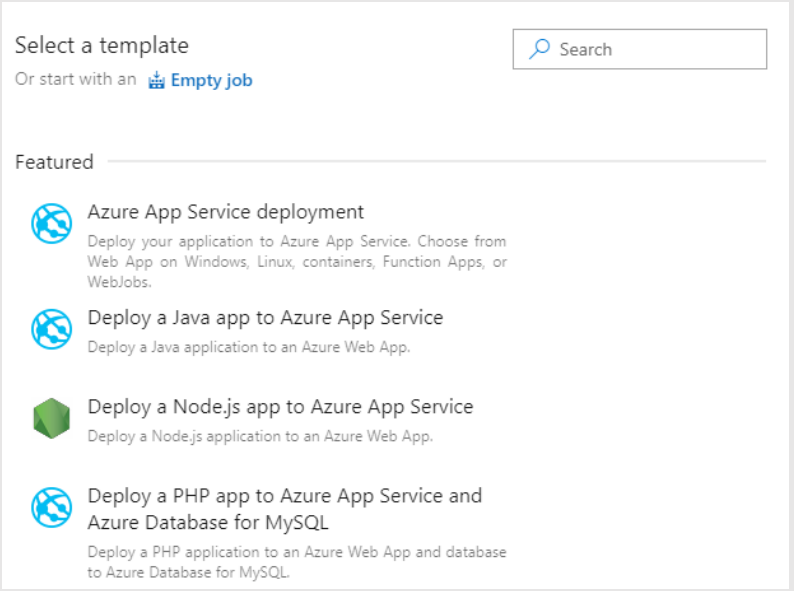
Azure DevOps - Releases

Create new Release Pipeline

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

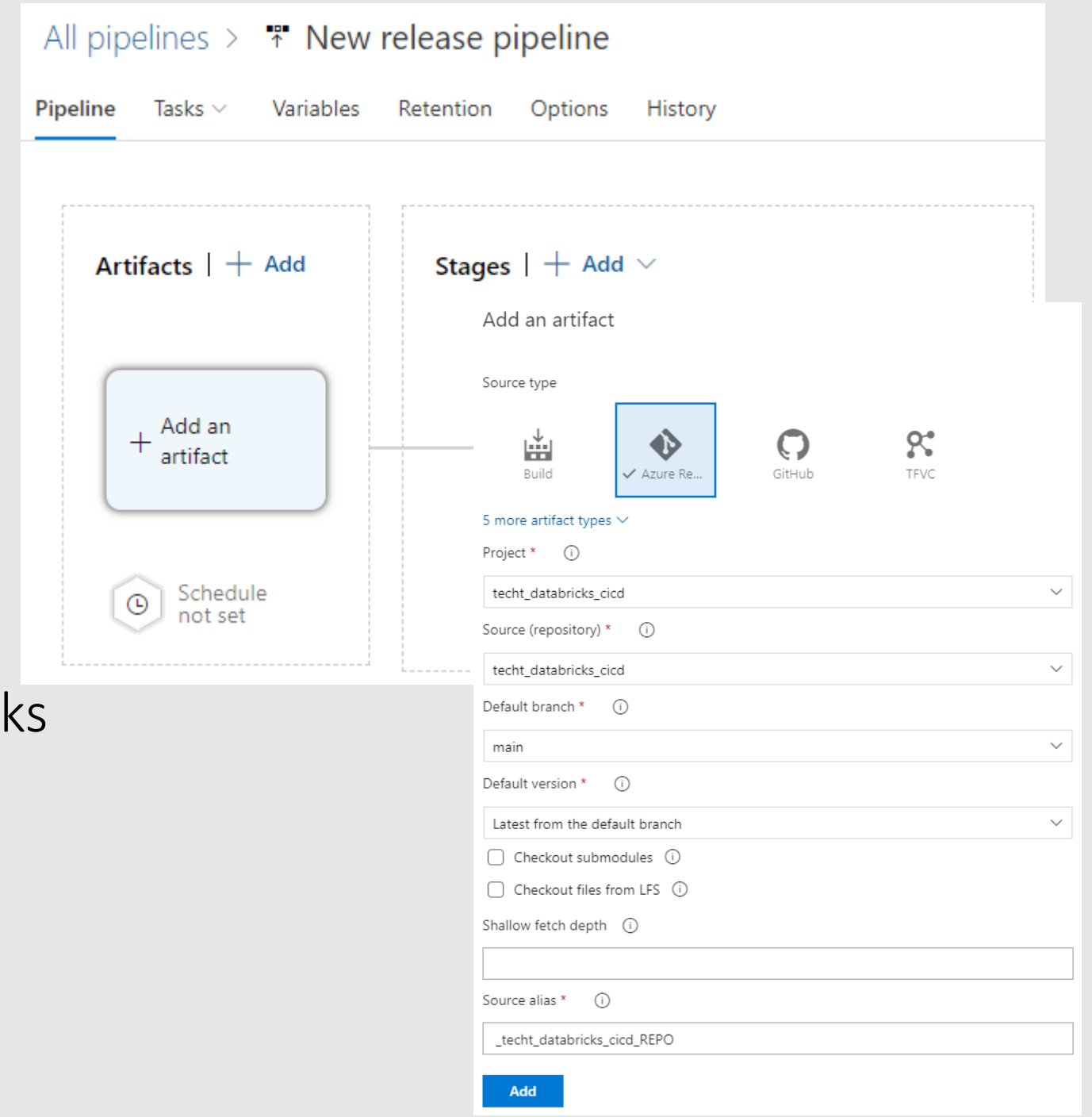






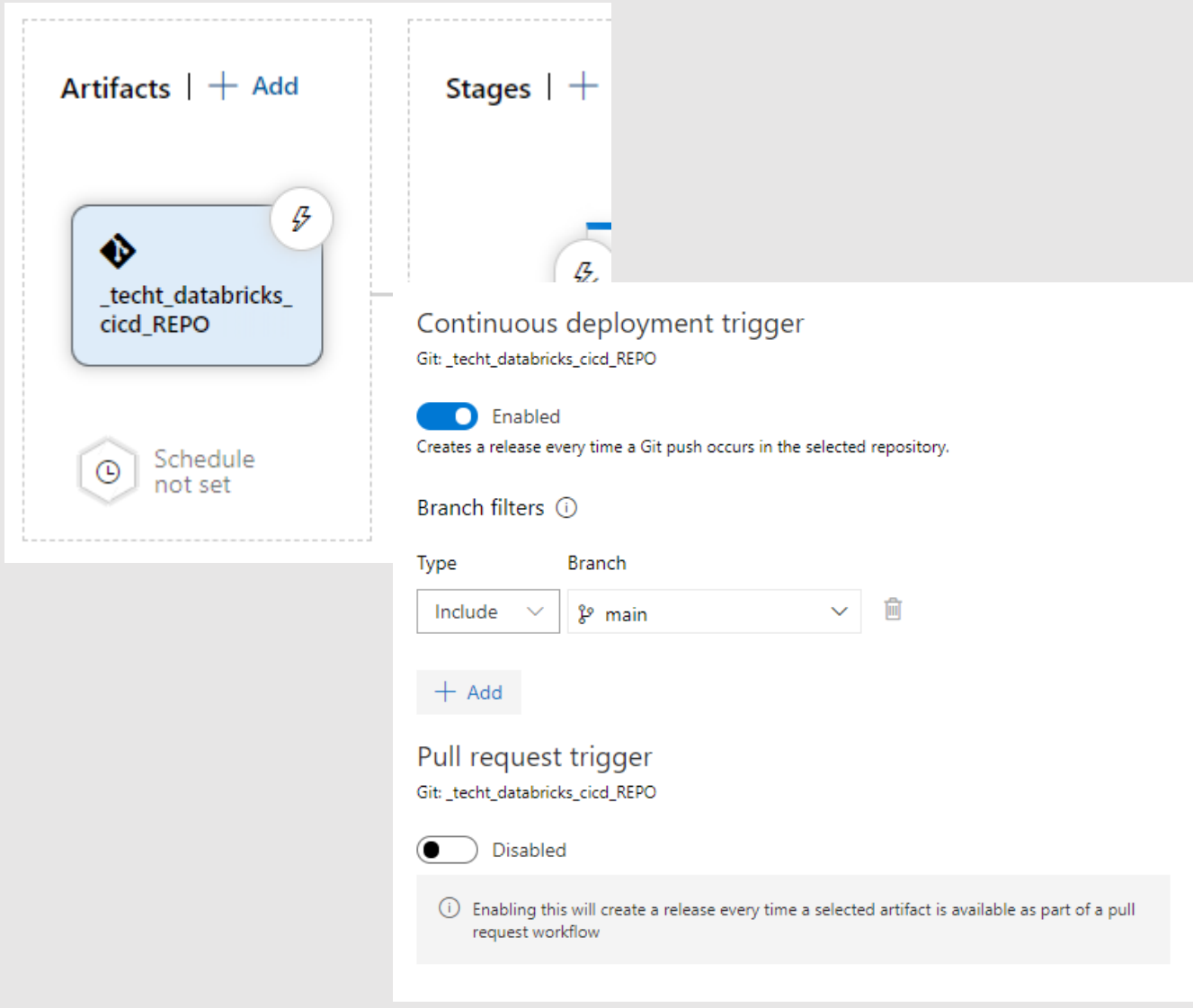
Add an Artifact

- 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



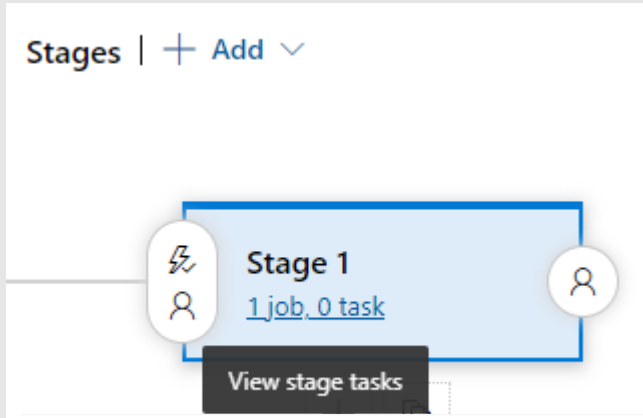
Configure pipeline trigger

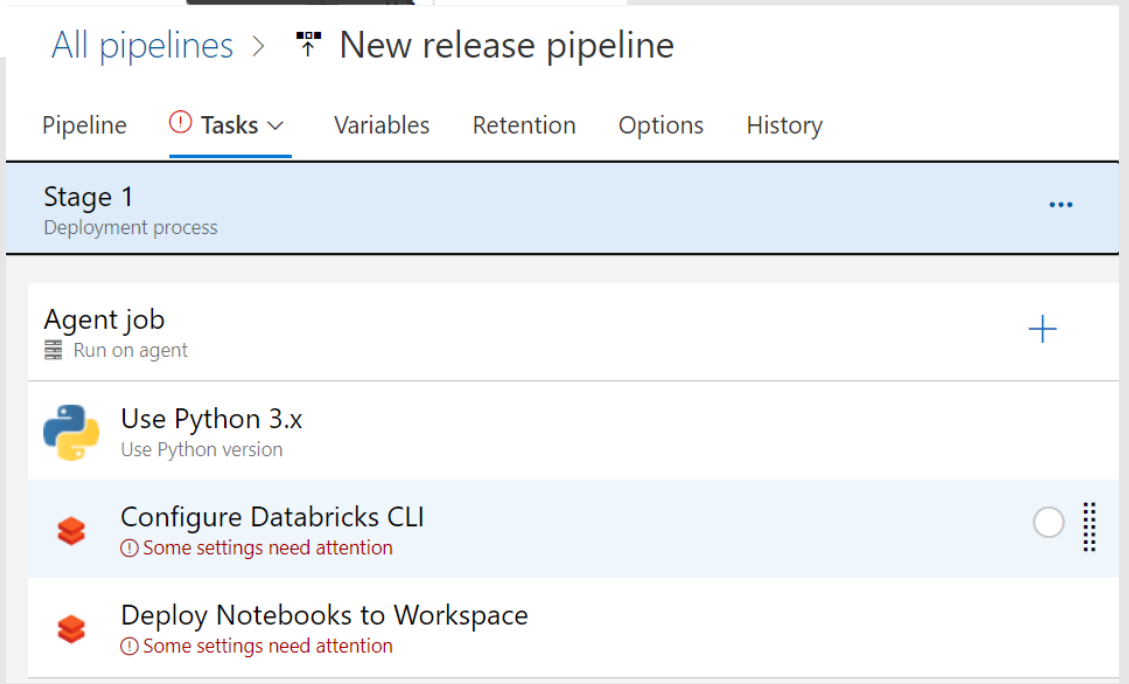
- We want to trigger the release pipeline when a pull request to the main branch is completed
- Lightening bolt
- Pull request trigger activates every time a PR is created in the whole repo – not suitable in this scenario



Add tasks to Stage object

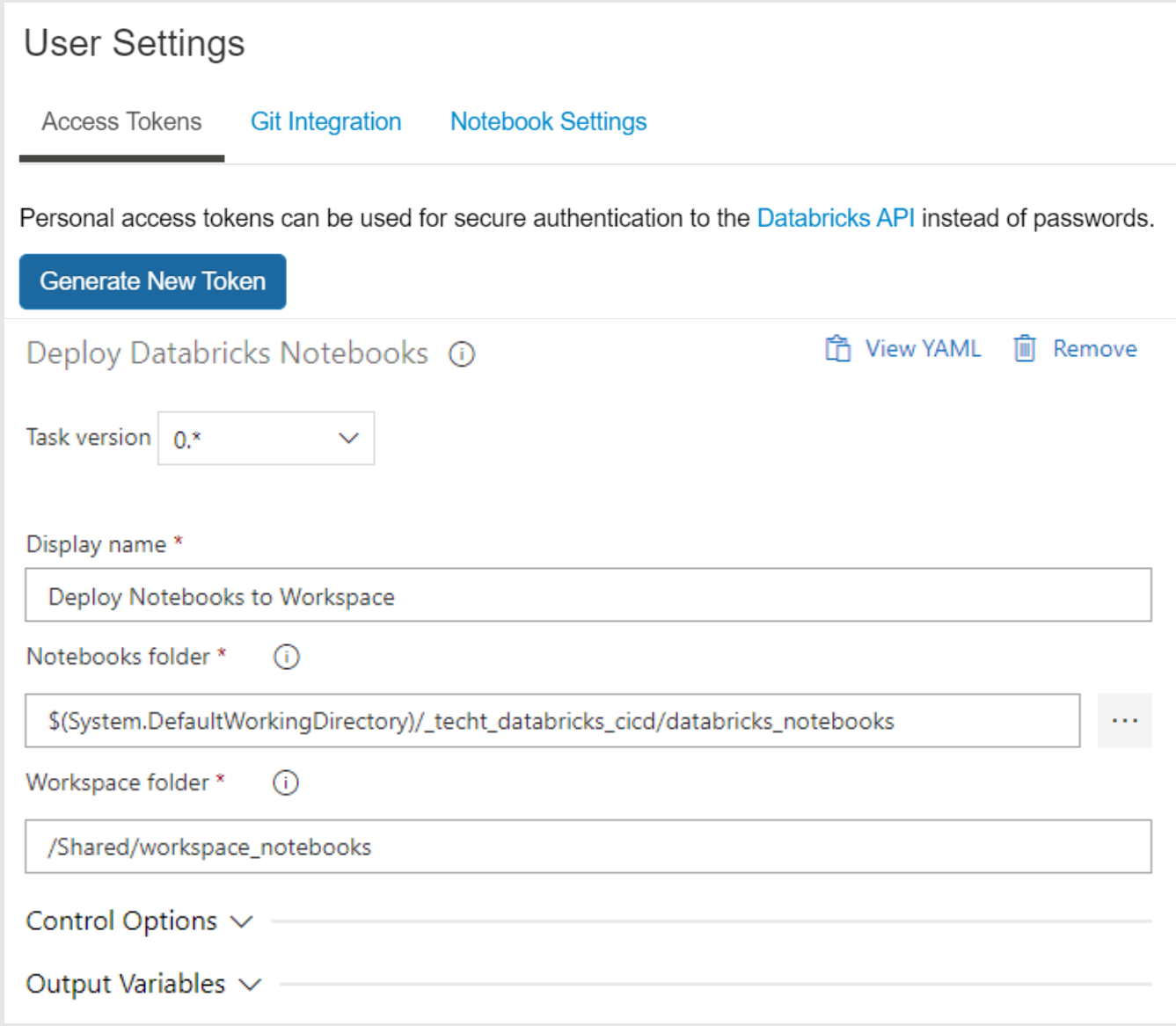
- Add the following tasks to the job → Use Python, Configure Databricks CLI & Deploy Notebooks to Workspace
- Will need to go in to each and add the configurations





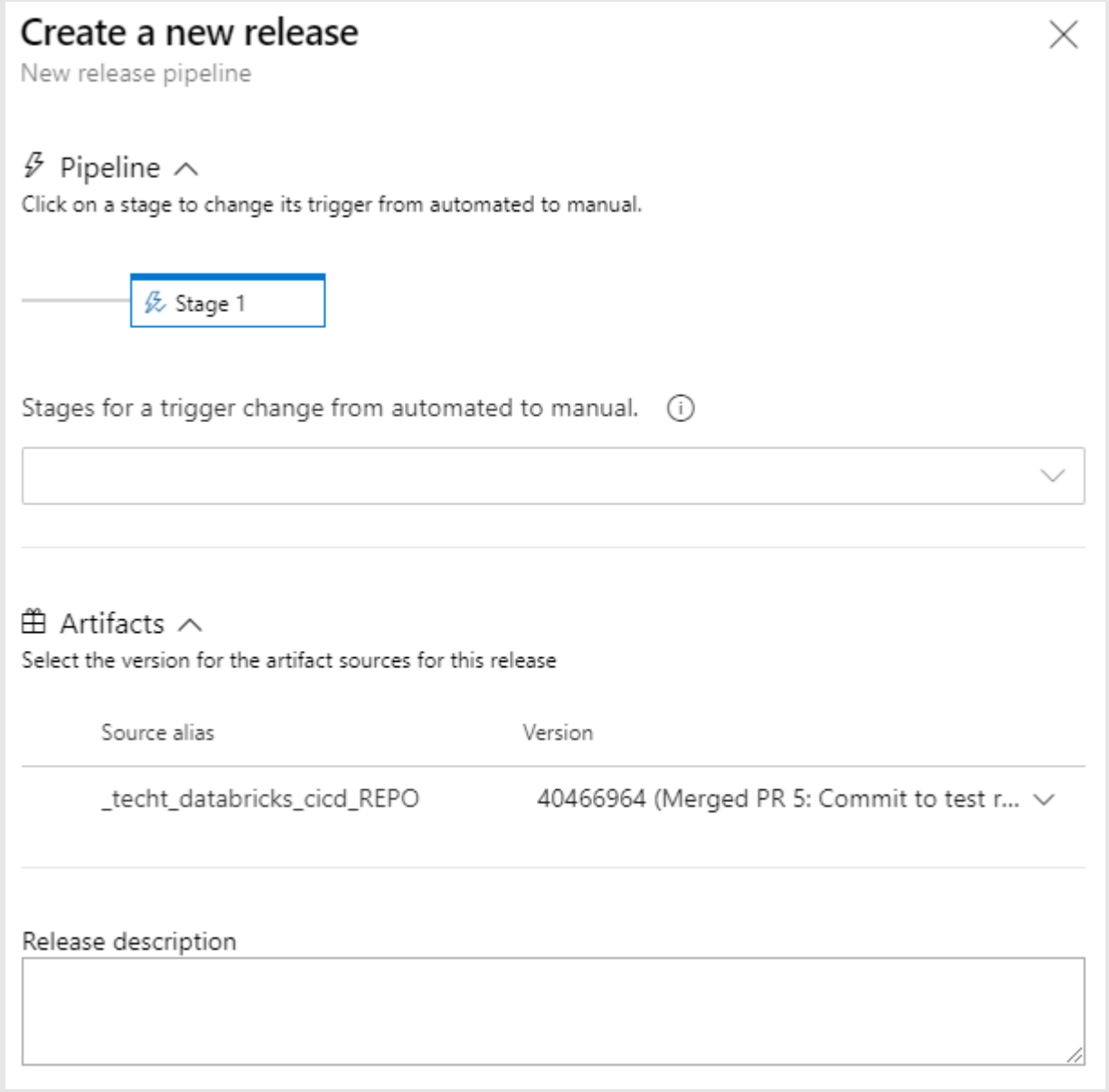
Access Configurations

- 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



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 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