



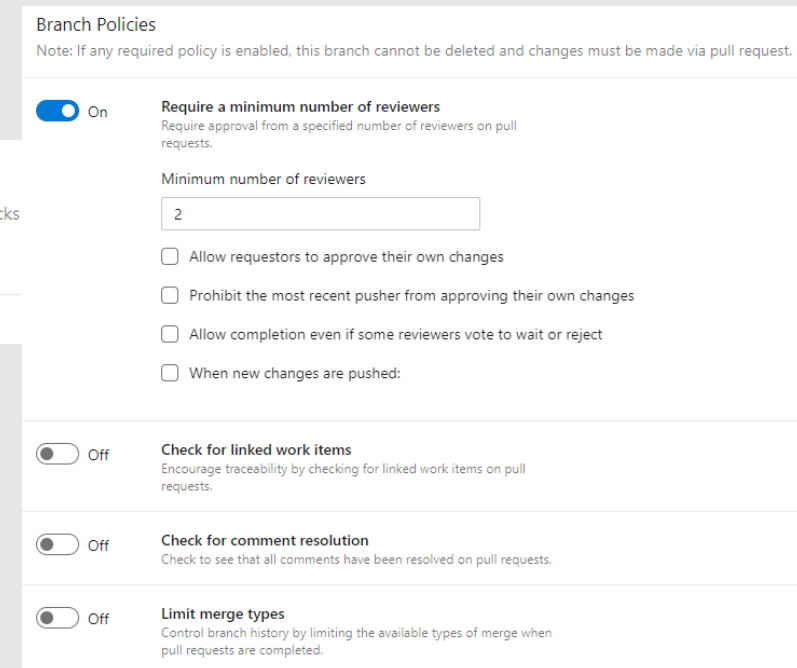
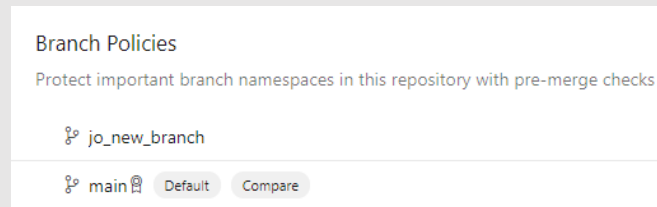
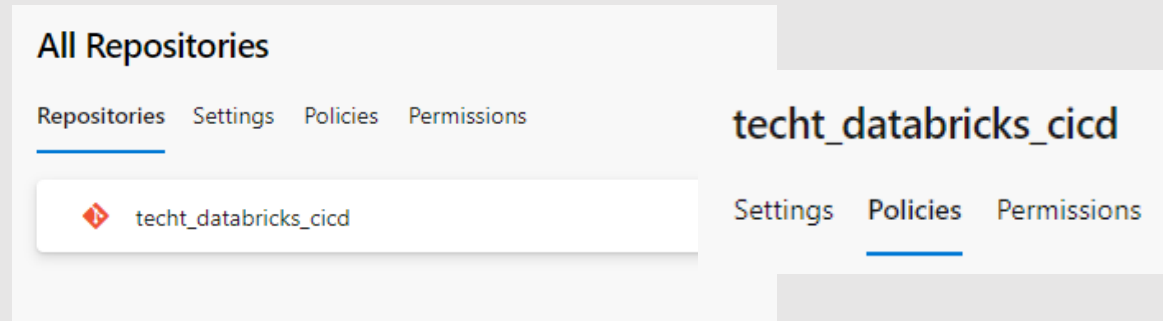
Azure Databricks



Azure DevOps - Repo

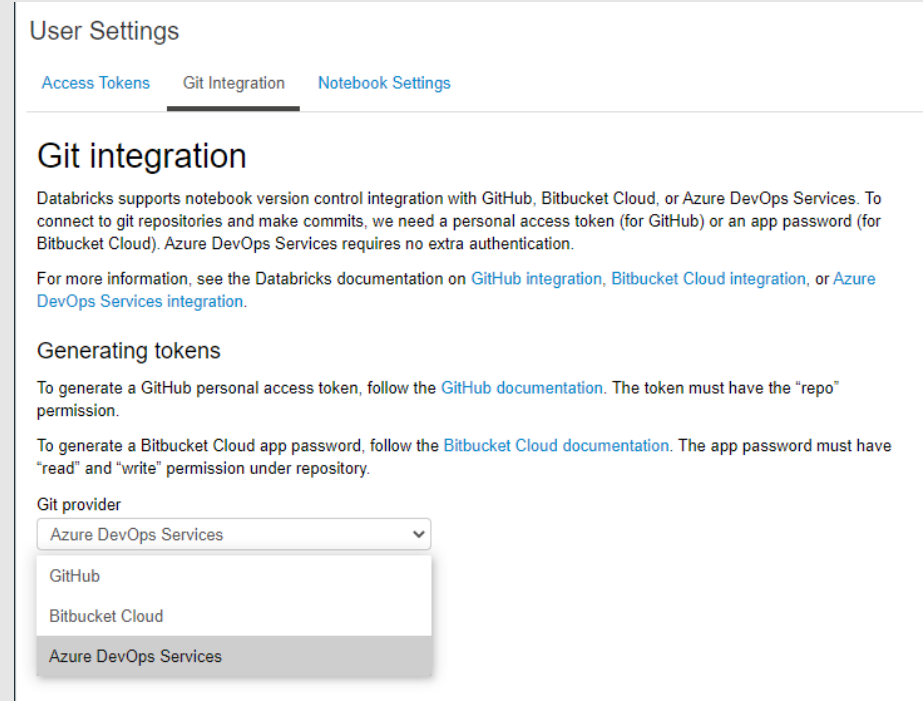
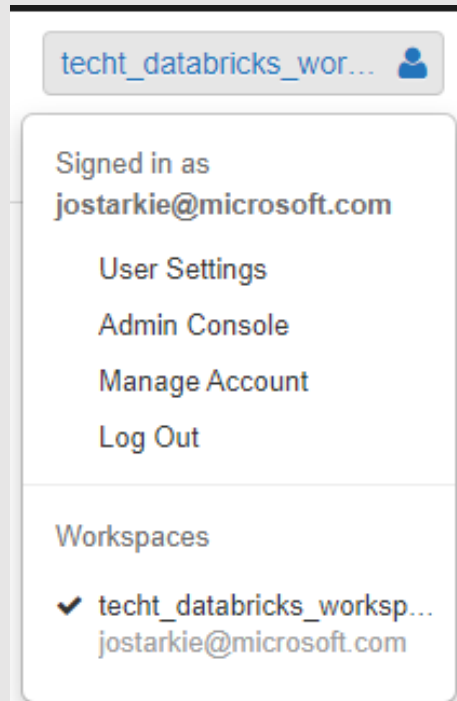
IMPORTANT! Protect the main Repo branch

- Project settings > Repos: Repositories > Select your repository
- Policies > Branch Policies > Select main branch
- Suggested:
 - “Require a minimum number reviewers”
 - “Check for comment resolution”



Set up DevOps/Git Integration

Click on person icon in top right > user settings > git integration > choose Git provider

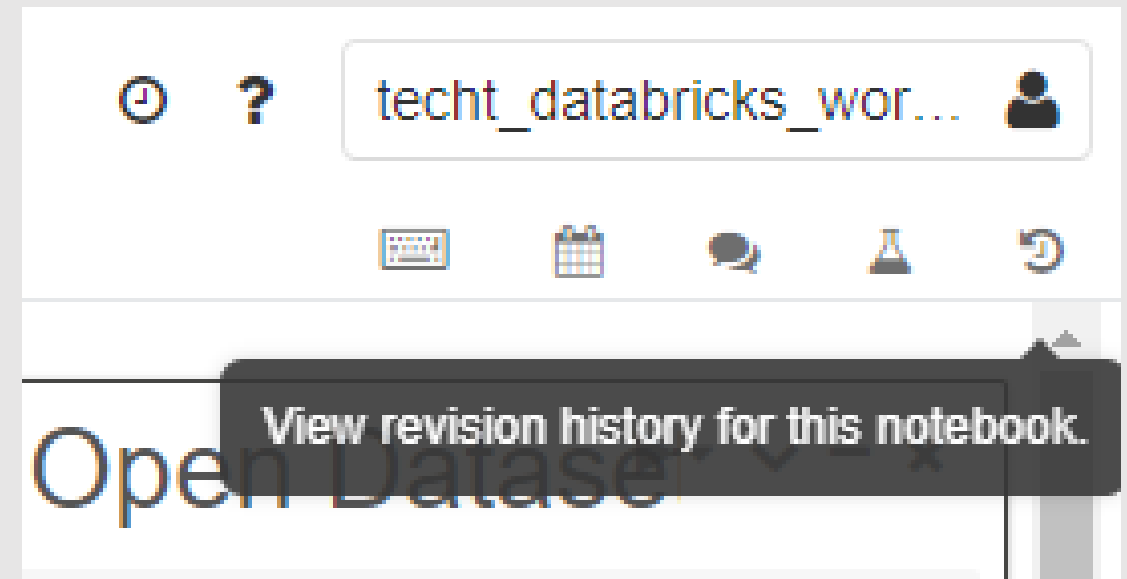
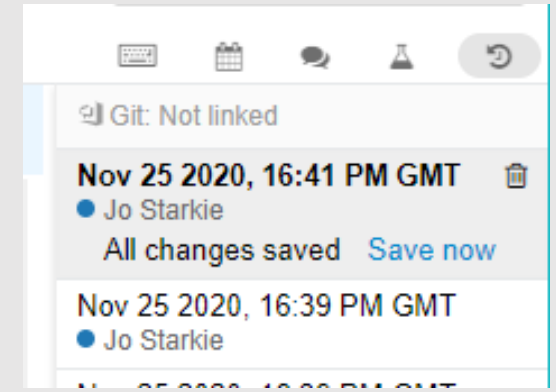
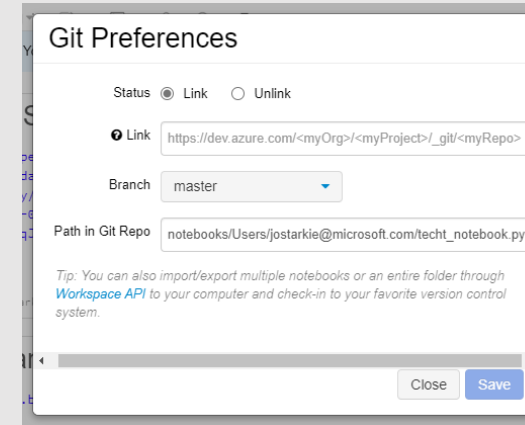


If Git tab doesn't exist, then enable via these instructions:

<https://docs.microsoft.com/en-gb/azure/databricks/administration-guide/workspace/notebooks#manage-git-versioning>

Notebook integration

- Navigate to notebook
- Revision history icon > Git: Not linked
- Linked radio button
- Link: URL of the Azure DevOps repository you'd like to link to
- Branch: best practice not to work on the default master/main branch > create new branch
- Path in Git Repo: define where in the Git repo you would like to save your file to – if file already exists in repo, you can put the file path and link directly – if file doesn't exist in repo, can define file name and format here.
- Ensure you include any sub-folders before the '/'
- Make sure to include the file extension otherwise will default to .txt



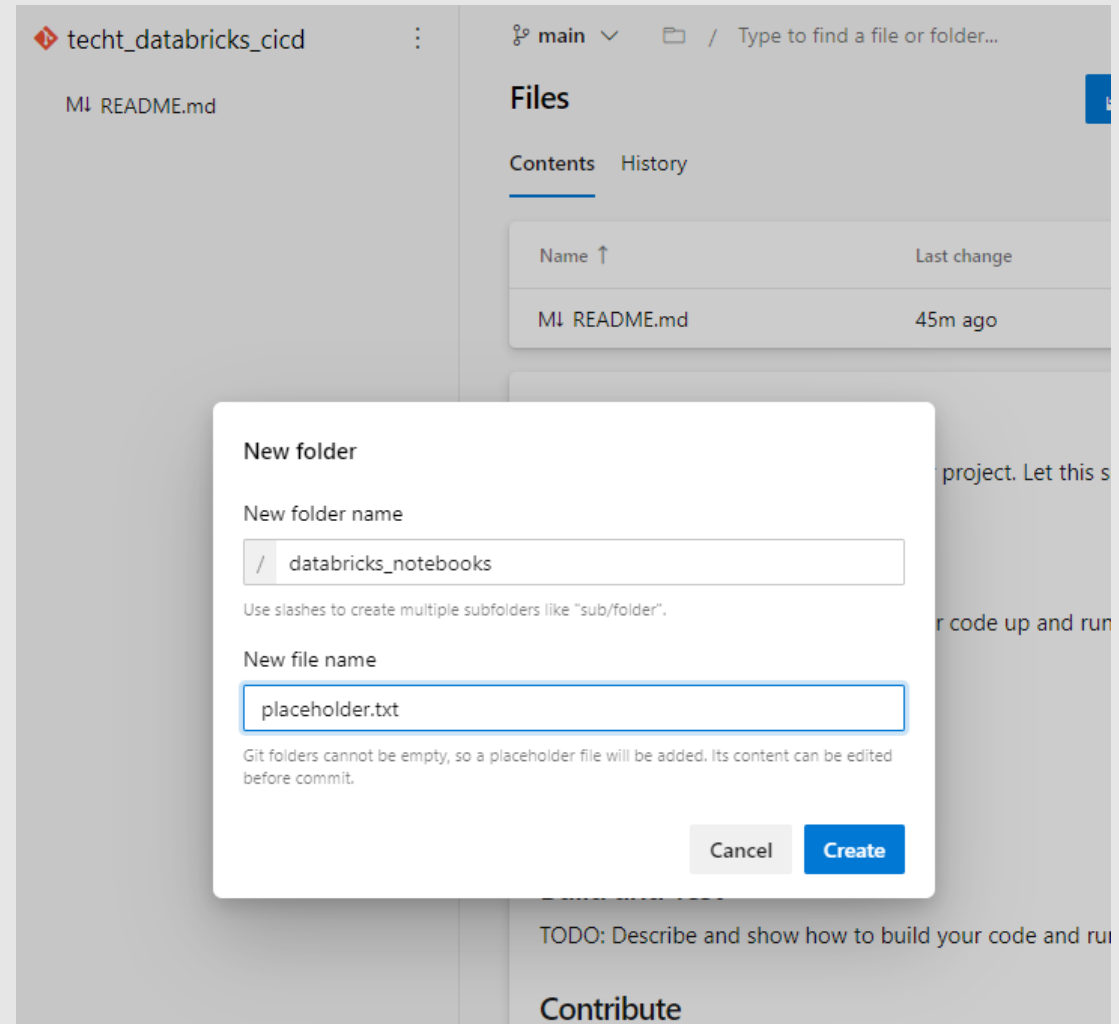
Creating sub-folders in DevOps

Click 3 dots, new > folder

Name folder

You must put a file in there to keep the new sub-folder. This file can be removed later

Alternatively can be created through IDE, push to your new branch



First commit

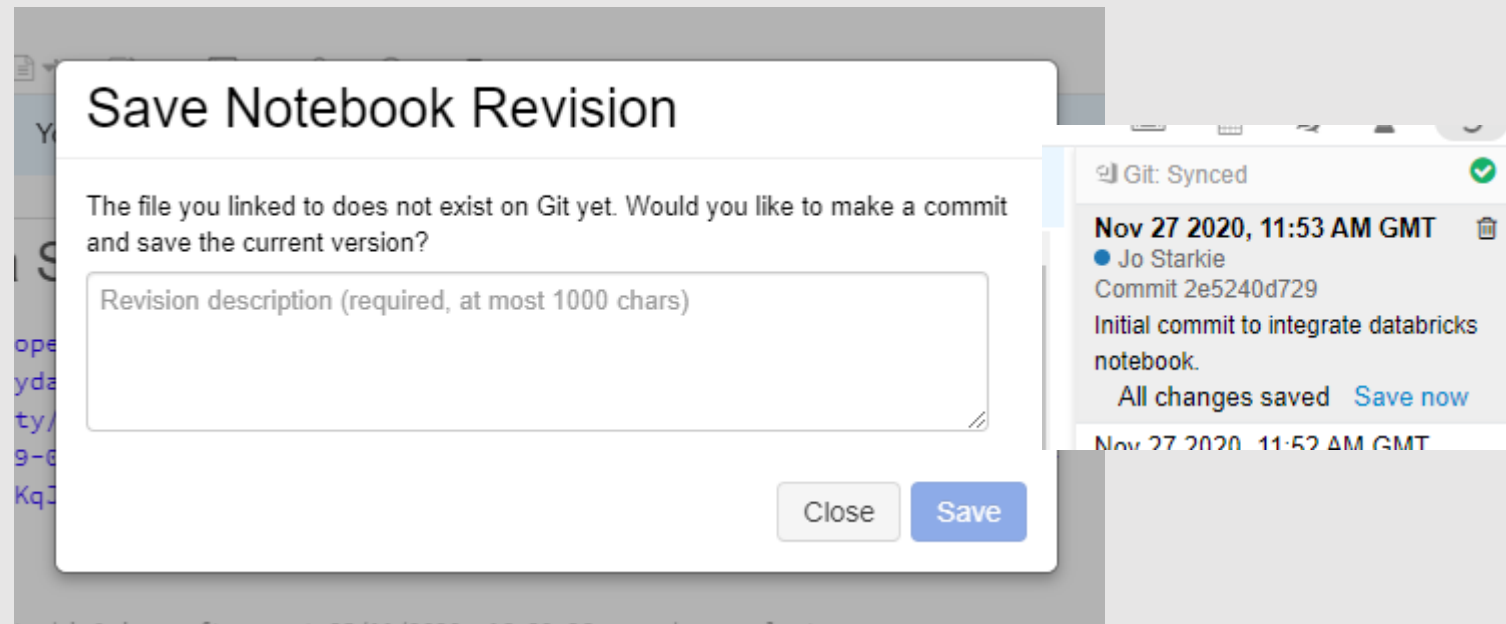
If the file doesn't exist, you'll be asked to commit to the branch

This will add a commit to the revision history.

Any further changes to the file will be added to a new line in the history tab

Once you're ready to push your changes to the repository click "save now"



Add your commit message and ensure "Also commit to Git" is checked to push through to remote repo



Revert changes

You can revert any of the changes you have committed to the repository from within Databricks revision history

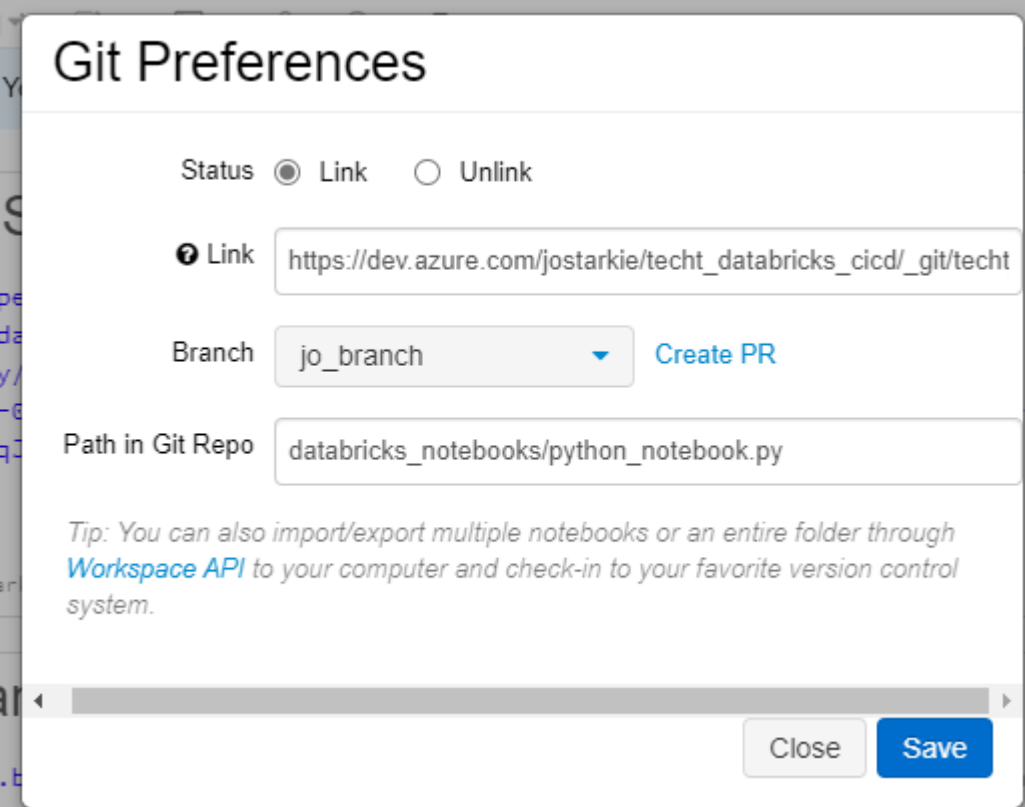
Click on version you want to go back to and click “Restore this revision”

Git: Synced 	
Nov 27 2020, 13:43 PM GMT	
● Jo Starkie	
Commit a74f9882c8	
Testing 2	
Nov 27 2020, 13:42 PM GMT	
● Jo Starkie	
Testing	
Nov 27 2020, 13:38 PM GMT	
● Jo Starkie	
Fixing folder issues.	
Nov 27 2020, 11:59 AM GMT	
● Jo Starkie	
Committing file changes to Azure D...	
Nov 27 2020, 11:54 AM GMT	
● Jo Starkie	
Nov 27 2020, 11:53 AM GMT	
● Jo Starkie	
Initial commit to integrate databrick...	
Nov 27 2020, 11:52 AM GMT 	
● Jo Starkie	
Restore this revision	
Nov 25 2020, 16:41 PM GMT	
● Jo Starkie	
Nov 25 2020, 16:39 PM GMT	
● Jo Starkie	
Nov 25 2020, 16:37 PM GMT	
● Jo Starkie	
Nov 25 2020, 16:34 PM GMT	
● Jo Starkie	
Nov 25 2020, 16:30 PM GMT	
● Jo Starkie	
Nov 25 2020, 16:28 PM GMT	
● Jo Starkie	

Pull Requests

Clicking “Create PR” opens a new tab directing to DevOps to complete the PR details

You can also create a pull request directly in DevOps: Repos > Pull Request

A screenshot of a 'Git Preferences' dialog box. The dialog has a title bar 'Git Preferences'. Inside, there's a 'Status' section with two radio buttons: 'Link' (selected) and 'Unlink'. Below that is a 'Link' section with an information icon and a text input field containing the URL 'https://dev.azure.com/jostarkie/techt_databricks_cicd/_git/techt'. The 'Branch' section has a dropdown menu showing 'jo_branch' and a 'Create PR' button. The 'Path in Git Repo' section has a text input field containing 'databricks_notebooks/python_notebook.py'. At the bottom, there's a tip: 'Tip: You can also import/export multiple notebooks or an entire folder through [Workspace API](#) to your computer and check-in to your favorite version control system.' At the very bottom right are 'Close' and 'Save' buttons.

Git Preferences

Status ☒ Link ☐ Unlink

Link

Branch [Create PR](#)

Path in Git Repo

Tip: You can also import/export multiple notebooks or an entire folder through [Workspace API](#) to your computer and check-in to your favorite version control system.

Changing Branches

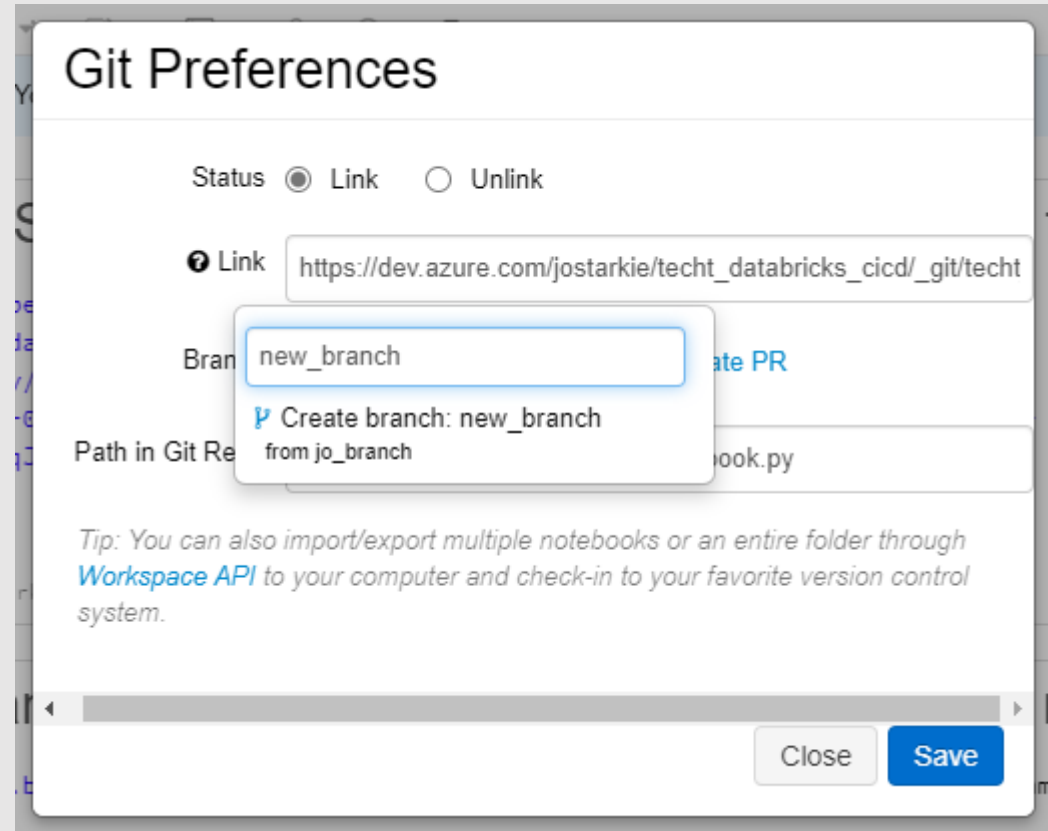
Click on the branch drop down

Start typing the name of your new branch

It will always default to the current selected branch

Example creating from jo_branch

If you want to create from master/main, need to switch branches first



Best Practices

- Code reviews:
 - You can link a notebook to your own fork and choose a branch.
 - We recommend using separate branches for each notebook.
 - Once you are happy with your changes, you can use the Create PR link in the Git Preferences dialog to take you to GitHub's pull request page.
 - The Create PR link displays only if you're not working on the default branch of the parent repository.