# Connect to a Git repository

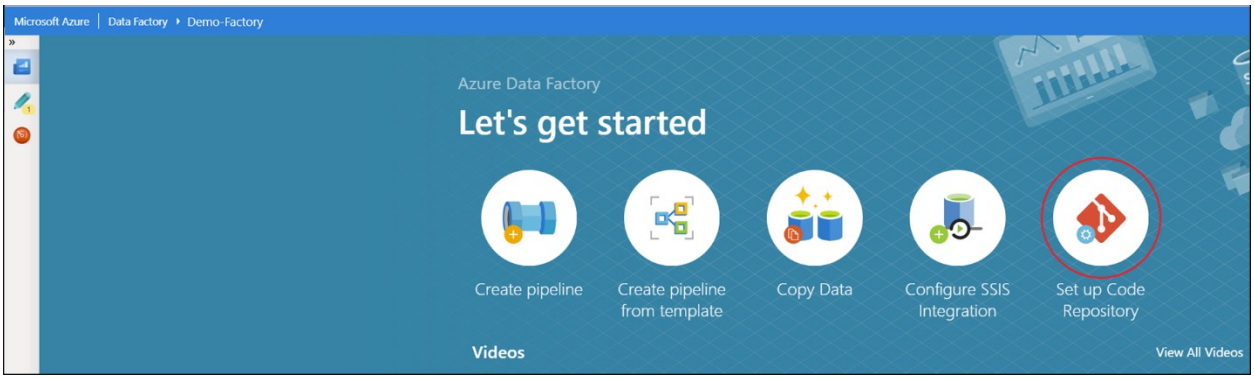
**Note:**

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it's likely that if you try this out you will encounter issues in your system.

**There are four different ways to connect a Git repository to your data factory for both Azure Repos and GitHub. After you connect to a Git repository, you can view and manage your configuration in the management hub under Git configuration in the Source control section.**

## Configuration method 1: Home page

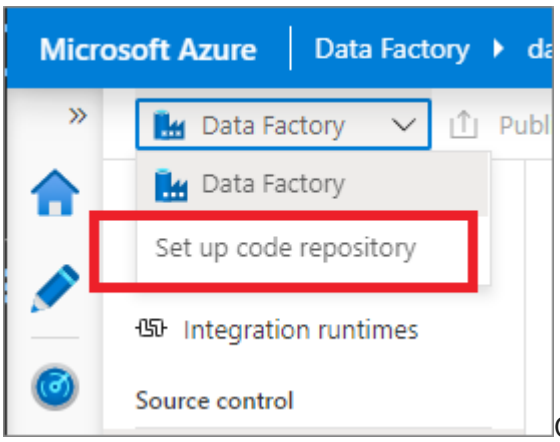
**In the Azure Data Factory home page, select Set up Code Repository.**




Configure a code repository from home page

## Configuration method 2: Authoring canvas

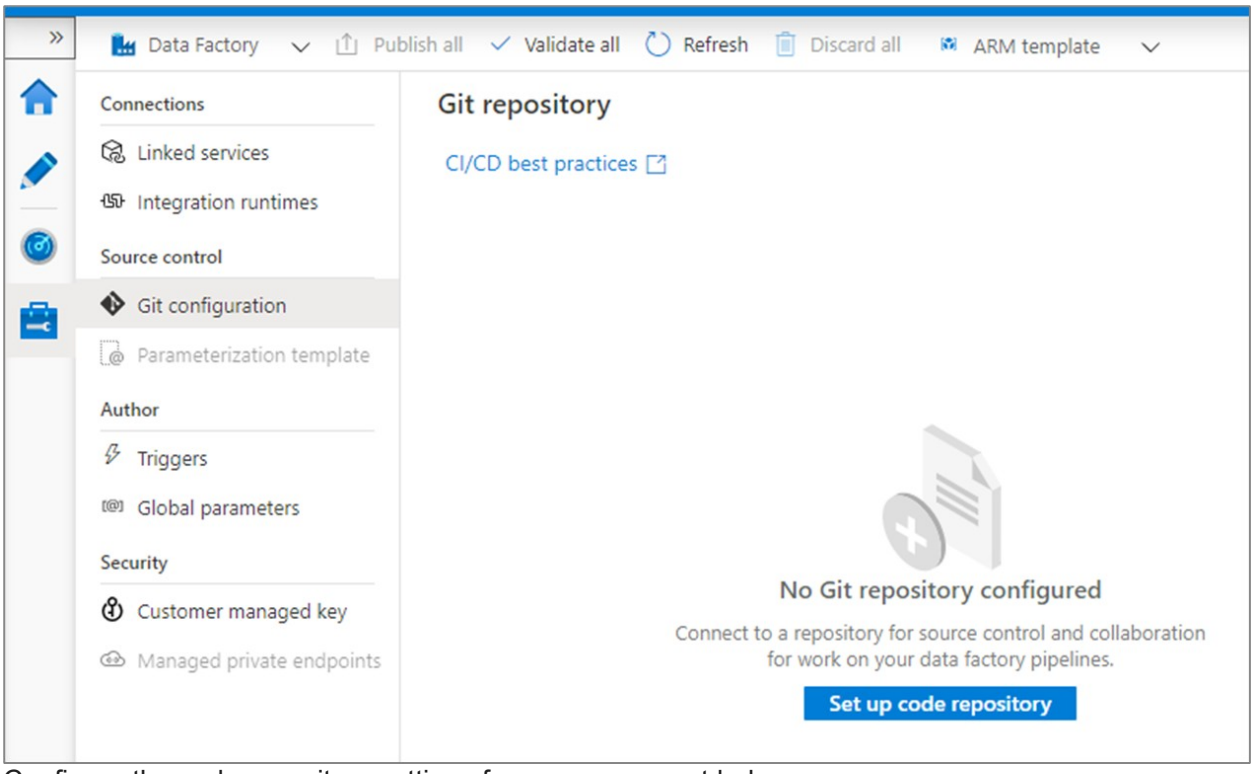
**In the Azure Data Factory UX authoring canvas, select the Data Factory drop-down menu, and then select Set up Code Repository.**



Configure the code repository settings from authoring

### Configuration method 3: Management hub

**Go to the management hub in the Azure Data Factory UX. Select Git configuration in the Source control section. If you have no repository connected, click Set up code repository.**



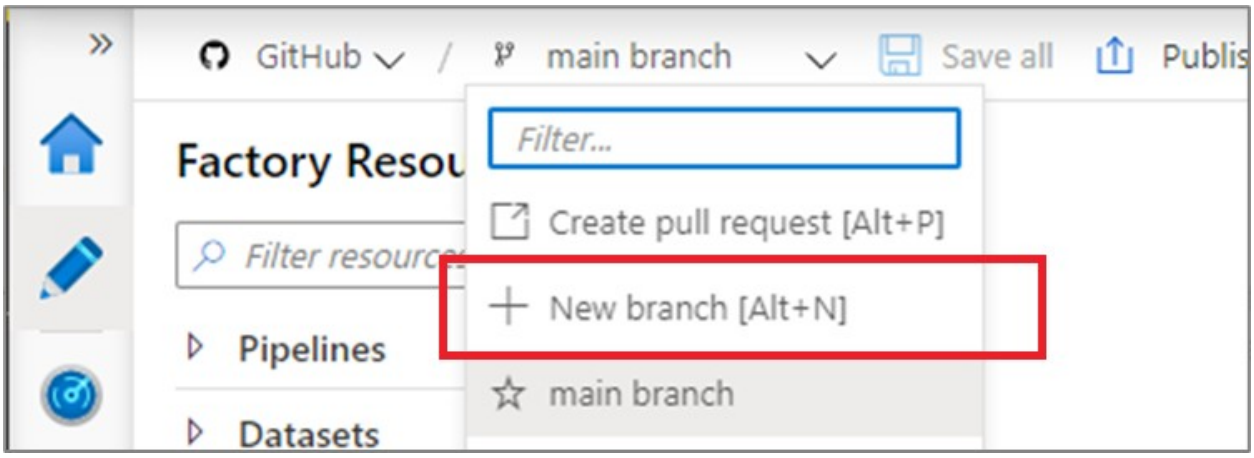
Configure the code repository settings from management hub

# Version control

**Version control systems (also known as *source control*) let developers collaborate on code and track changes that are made to the code base. Source control is an essential tool for multi-developer projects.**

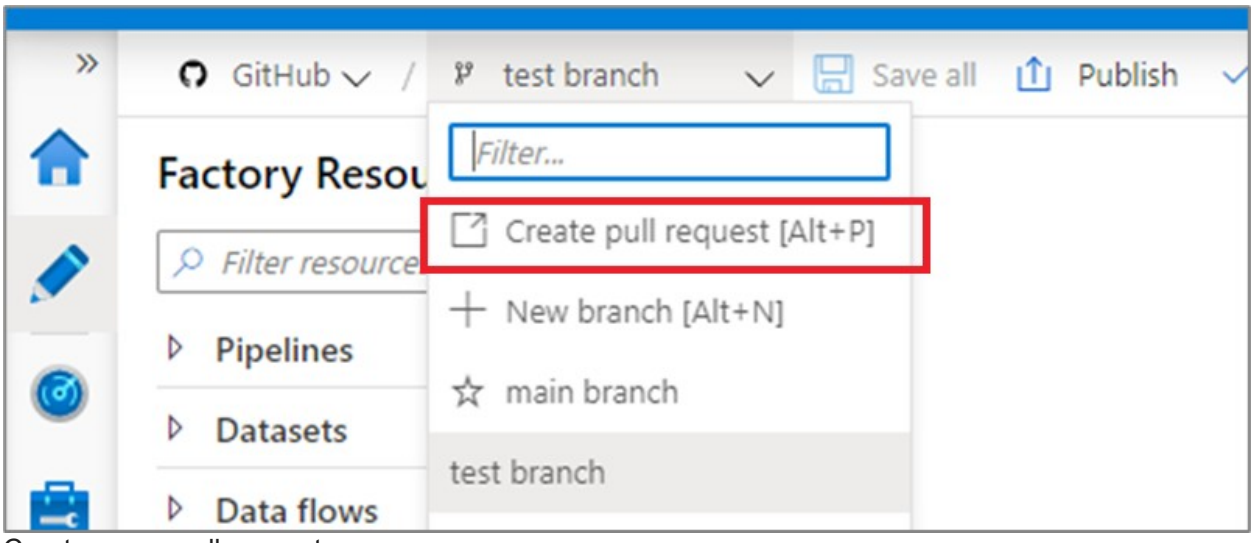
# Creating feature branches

**Each Azure Repos Git repository that's associated with a data factory has a collaboration branch. (`main` is the default collaboration branch). Users can also create feature branches by clicking + New Branch in the branch dropdown. Once the new branch pane appears, enter the name of your feature branch.**



Create a new branch

**When you are ready to merge the changes from your feature branch to your collaboration branch, click on the branch dropdown and select Create pull request. This action takes you to Azure Repos Git where you can raise pull requests, do code reviews, and merge changes to your collaboration branch. You are only allowed to publish to the Data Factory service from your collaboration branch.**



Create a new pull request

## Configure publishing settings

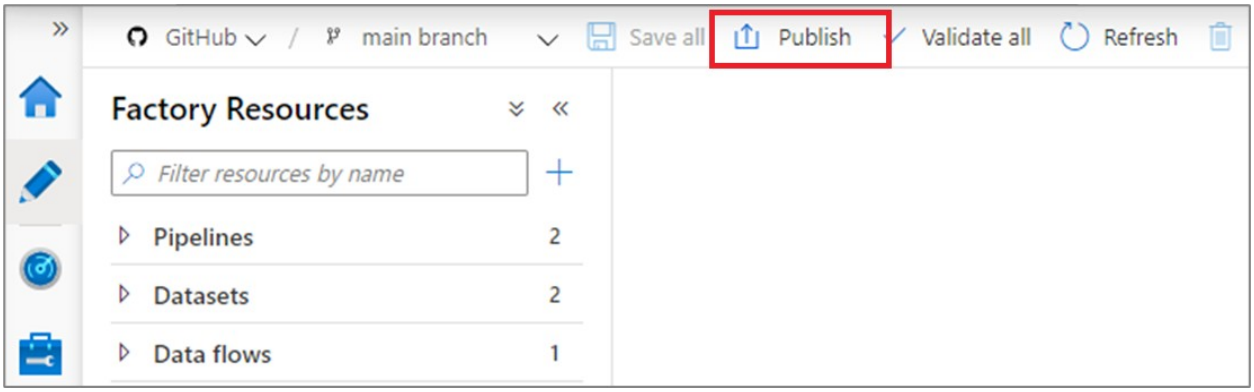
**By default, data factory generates the Resource Manager templates of the published factory and saves them into a branch called `adf_publish`. To configure a custom publish branch, add a `publish_config.json` file to the root folder in the collaboration branch. When publishing, Azure Data Factory reads this file, looks for the field `publishBranch`, and saves all Resource Manager templates to the specified location. If the branch doesn't exist, data factory will automatically create it. And example of what this file looks like is below:**

```
{
    "publishBranch": "factory/adf_publish"
}
```

**Azure Data Factory can only have one publish branch at a time. When you specify a new publish branch, Data Factory doesn't delete the previous publish branch. If you want to remove the previous publish branch, delete it manually. Note: Data Factory only reads the `publish_config.json` file when it loads the factory. If you already have the factory loaded in the portal, refresh the browser to make your changes take effect.**

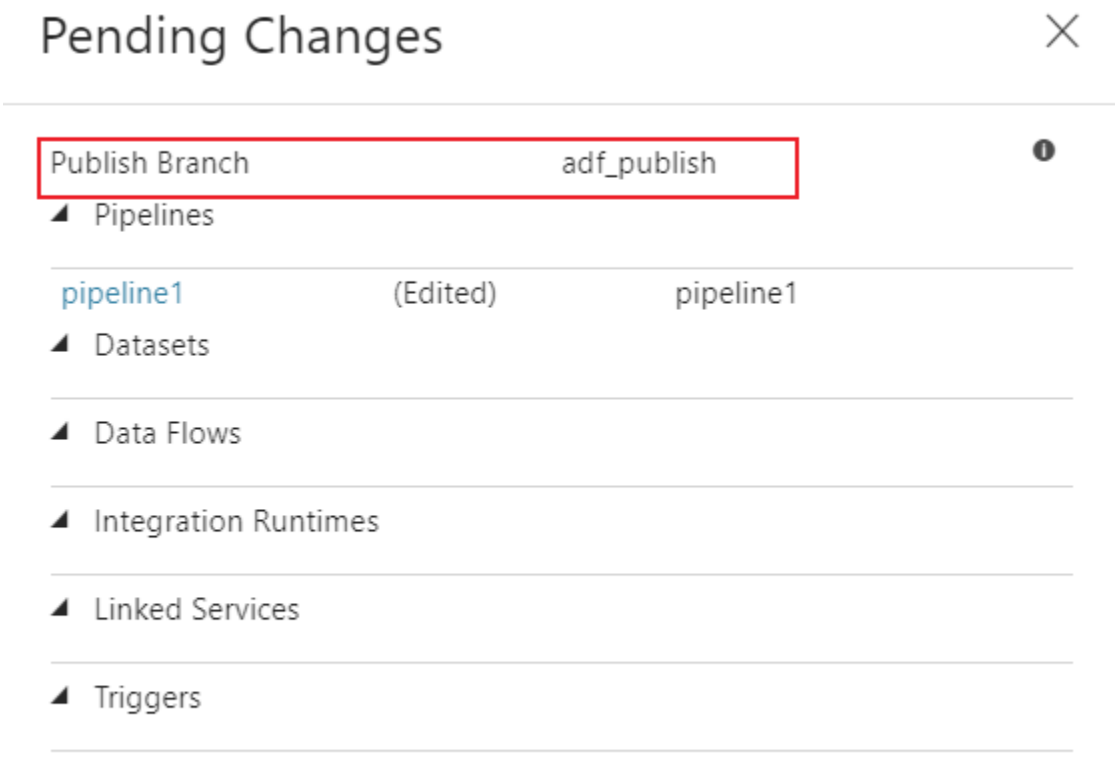
## Publish code changes

**After you have merged changes to the collaboration branch , click Publish to manually publish your code changes in the collaboration branch to the Data Factory service.**




Publish changes to the Data Factory service

A side pane will open where you confirm that the publish branch and pending changes are correct. Once you verify your changes, click **OK** to confirm the publish.




Confirm the correct publish branch

**Important: The collaboration branch is not representative of what's deployed in the Data Factory service. The collaboration branch *must* be published manually to the Data Factory service.**

# Best practices for Git integration

## Permissions

### Typically you don't want every team member to have permissions to update the Data Factory. The following permissions settings are recommended:

- All team members should have read permissions to the Data Factory.
- Only a select set of people should be allowed to publish to the Data Factory. To do so, they must have the **Data Factory contributor** role on the **Resource Group** that contains the Data Factory.

### It's recommended to not allow direct check-ins to the collaboration branch. This restriction can help prevent bugs as every check-in will go through a pull request review process.

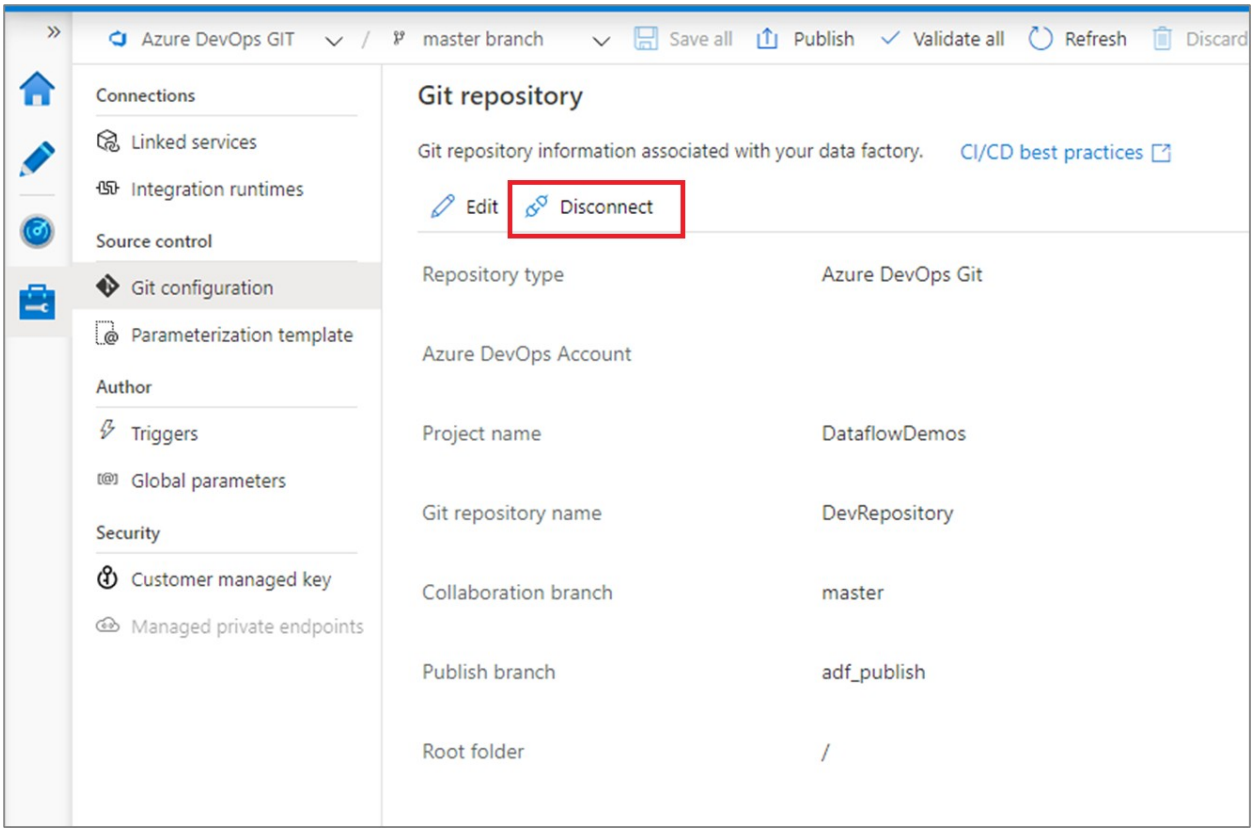
## Using passwords from Azure Key Vault

### It's recommended to use Azure Key Vault to store any connection strings or passwords or managed identity authentication for Data Factory Linked Services. For security reasons, data factory doesn't store secrets in Git. Any changes to Linked Services containing secrets such as passwords are published im../mediately to the Azure Data Factory service.

### Using Key Vault or MSI authentication also makes continuous integration and deployment easier as you won't have to provide these secrets during Resource Manager template deployment.

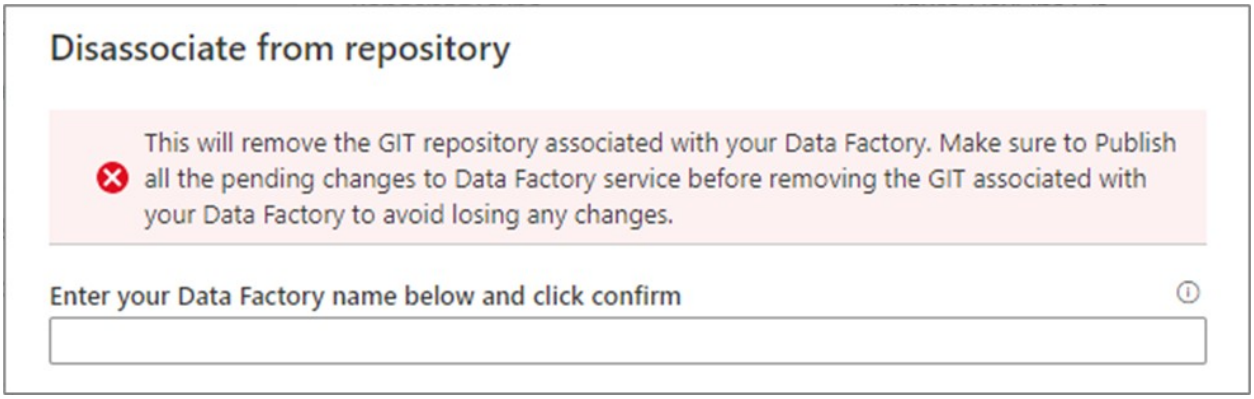
# Switch to a different Git repository

### To switch to a different Git repository, go to Git configuration page in the management hub under **Source control**. Select **Disconnect**.



Disconnect the current Git repo

**Enter your data factory name and click confirm to remove the Git repository associated with your data factory.**



Remove the association with the current Git repo

**After you remove the association with the current repo, you can configure your Git settings to use a different repo and then import existing Data Factory resources to the new repo.**

**Important: Removing Git configuration from a data factory doesn't delete anything from the repository. The factory will contain all**

**published resources. You can continue to edit the factory directly against the service.**