

# Lab 1 – Create Azure resources

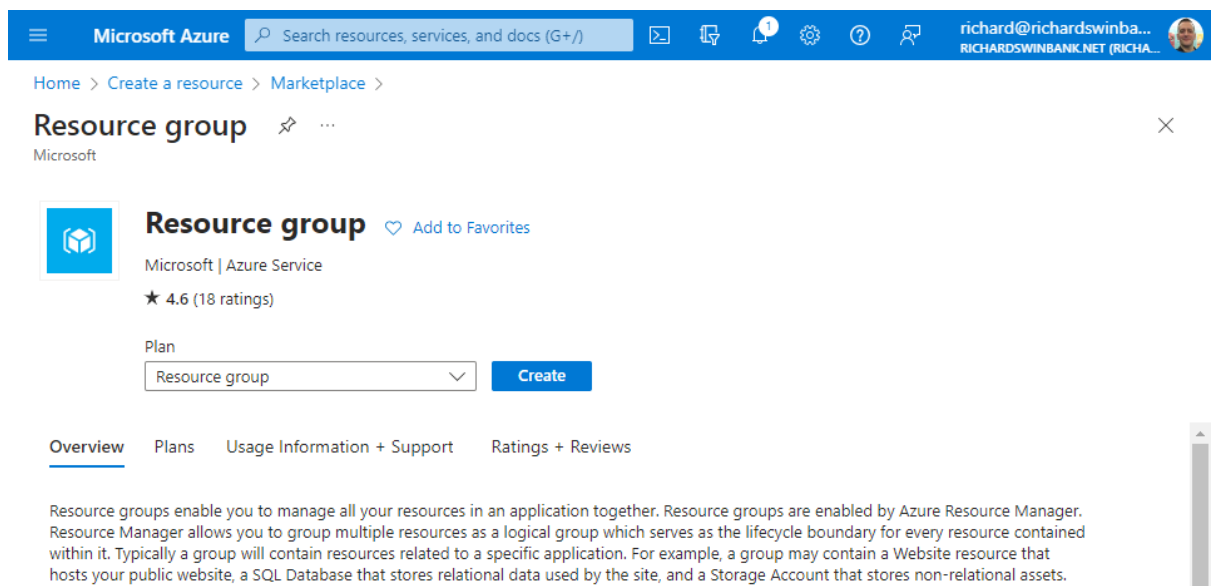
Welcome to Lab 1! To complete this lab series, you're going to need a few things.

- A web browser compatible with Azure Data Factory (ADF) – use Microsoft Edge or Google Chrome.
- An Azure subscription – if you don't have one, you can sign up for a free trial at <https://azure.microsoft.com/en-gb/free/>. Whichever subscription you use, you'll need enough access to create resources in it.
- A GitHub repository where you can store lab code – if you don't already have a GitHub account, you can sign up for one at <https://github.com/signup>. (We use GitHub for these labs, but Azure Data Factory also supports Azure DevOps Repos).

## Lab 1.1 – Create a resource group

Resource groups are logical containers for resources in Azure. In this lab you will create a resource group to contain all the resources you create in later labs. This will make cleaning up easier – when you've finished all the labs, you can just delete the resource group.

1. In the Azure Portal (<https://portal.azure.com/>) click “Create a resource” and use the “Search services and marketplace” textbox to search for “Resource group”.
2. On the “Resource group” overview, click “Create”.



3. Give the resource group a name, and choose the Region geographically closest to you.
4. Click “Review + create”, then “Create”.

## Lab 1.2 – Create data lake storage

Data lake storage is blob storage in an Azure storage account, with one particularly important feature: hierarchical namespaces are **enabled**. This makes certain file operations – renaming file folders, for example – much more efficient.

1. In the portal, click “Create a resource” and search for “Storage account”. Click “Create” on the overview screen.



## 2. Complete the “Basics” tab like this:

- Choose your subscription and the resource group you created in Lab 1.1.
- Enter a storage account name – this must be globally unique (across the entire Azure platform).
- Choose the same location you specified for your resource group. Having storage located close to you makes data movement faster and cheaper.
- Choose redundancy option “Locally-redundant storage”. This is nice and cheap for lab work, but you’ll want something more resilient in a production environment!

Home > Create a resource > Marketplace > Storage account >

### Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription \* Azure subscription 1

Resource group \* sqlbits2024-rg  
[Create new](#)

#### Instance details

Storage account name ⓘ \* sqlbits2024sa

Region ⓘ \* (Europe) UK South  
[Deploy to an edge zone](#)

Performance ⓘ \*

☒ Standard: Recommended for most scenarios (general-purpose v2 account)

☐ Premium: Recommended for scenarios that require low latency.

Redundancy ⓘ \* Locally-redundant storage (LRS)

[Review](#) < Previous Next: Advanced > [Give feedback](#)

## 3. On the “Advanced” tab, tick the “Enable hierarchical namespace” checkbox. This step is **essential** to make the storage account a data lake.

Home > Create a resource > Marketplace > Storage account >

### Create a storage account

Basics **Advanced** Networking Data protection Encryption Tags Review

Minimum TLS version ⓘ VERSION 1.2

Permitted scope for copy operations (preview) ⓘ From any storage account

#### Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) [Learn more](#)

Enable hierarchical namespace ☒

#### Access protocols

Blob and Data Lake Gen2 endpoints are provisioned by default [Learn more](#)

- Click “Review” to accept defaults on the remaining tabs and skip to the “Review” tab. Click “Create”. When the data lake finishes deploying (this may take a couple of minutes), click on “Go to resource”.

The “Overview” page of the “Storage account” blade contains an “Essentials” section, above a tabbed pane. The entry in the top left of the “Properties” tab should read “Data Lake Storage”:

Home > sqlbits2024-rg >

sqlbits2024sa  
Storage account

Search

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage browser

Data storage

Containers

File shares

Queues

Tables

Security + networking

Networking

Access keys

Shared access signature

Upload Open in Explorer Delete Move Refresh

Essentials

Resource group (move)  
sqlbits2024-rg

Location  
uksouth

Subscription (move)  
Azure subscription 1

Subscription ID  
6bd2daaf-42fa-4cf4-a2e3-09cca1e5f6c6

Disk state  
Available

Tags (edit)  
Add tags

Performance  
Standard

Replication  
Locally-redundant storage (LRS)

Account kind  
StorageV2 (general purpose v2)

Provisioning state  
Succeeded

Created  
02/03/2024, 14:41:34

Properties Monitoring Capabilities (5) Recommendations (0) Tutorials

Tools + SDKs

Data Lake Storage

Hierarchical namespace  
Enabled

Default access tier

Security

Require secure transfer for REST API operations  
Enabled

If “Blob service” appears instead, you forgot to enable hierarchical namespaces and have created an ordinary blob storage account. In this case, the tab will also indicate that the “Hierarchical namespace” property is “Disabled” – click on the “Disabled” link to upgrade the blob storage account to data lake storage.

- Click on the “Containers” link in the Storage account blade sidebar (also indicated on the left of the screenshot above). Use the “+ Container” button to create a container with the name “lakeroot”. After creation, the container appears in the list.

Home > sqlbits2024sa\_1709390492159 | Overview > sqlbits2024sa

sqlbits2024sa | Containers

Storage account

Search

+ Container Change access level Restore containers Refresh Delete

Search containers by prefix

Show deleted containers

Name	Last modified	Anonymous access l...	Lease state
\$logs	3/2/2024, 2:41:57 PM	Private	Available
lakeroot	3/2/2024, 2:51:07 PM	Private	Available

- Click on the new “lakeroot” entry to open the container. The menu bar above the list now contains a “+ Add Directory” button – use this to create two directories in the container: “Raw” and “Conformed”.

Home > sqlbits2024sa\_1709390492159 | Overview > sqlbits2024sa | Containers >

**lakeroot**  
Container

Search

Upload Add Directory Refresh Rename Delete Change tier

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: lakeroot

Search blobs by prefix (case-sensitive)

Show deleted objects

Name	Modified	Access tier	Archive status
<input type="checkbox"/> Conformed			
<input type="checkbox"/> Raw			

## Lab 1.3 – Create an Azure Data Factory

The main event! It's time to create your Azure Data Factory instance.

- In the Azure portal, click “Create a resource” and search for “Data Factory”. Click “Create” on the overview screen.
- Complete the “Basics” tab like this:
  - Choose your subscription and the resource group you created in Lab 1.1.
  - Enter a data factory name – this must be globally unique.
  - Choose the same location (region) you specified for your storage account. **This has cost implications** – transferring data from a storage account in one region to a data factory in another incurs an outbound data transfer charge.
  - The version dropdown offers only one option – V2.

Home > Create a resource > Marketplace > Data Factory >

**Create Data Factory**

Basics Git configuration Networking Advanced Tags Review + create

One-click to create data factory with sample pipeline and datasets. [Try it](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* Azure subscription 1

Resource group \* sqlbits2024-rg  
[Create new](#)

**Instance details**

Name \* sqlbits2024-adf ✓

Region \* UK South

Version \* V2

Previous Next Review + create

[Give feedback](#)

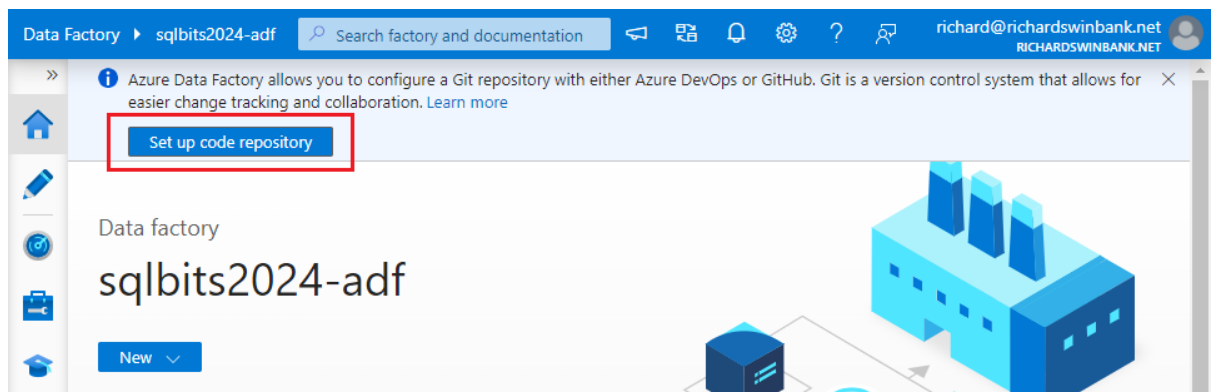
3. On the “Git configuration” tab, make sure that the “Configure Git later” checkbox is ticked – we’ll do this after the factory has been created because it makes setup a bit easier.
4. Click “Review + create” to accept defaults on the remaining tabs and skip to the “Review + create” tab. Click “Create”. When factory deployment is complete, click on “Go to resource”, then on the “Azure Data Factory Studio” tile to launch ADF Studio.



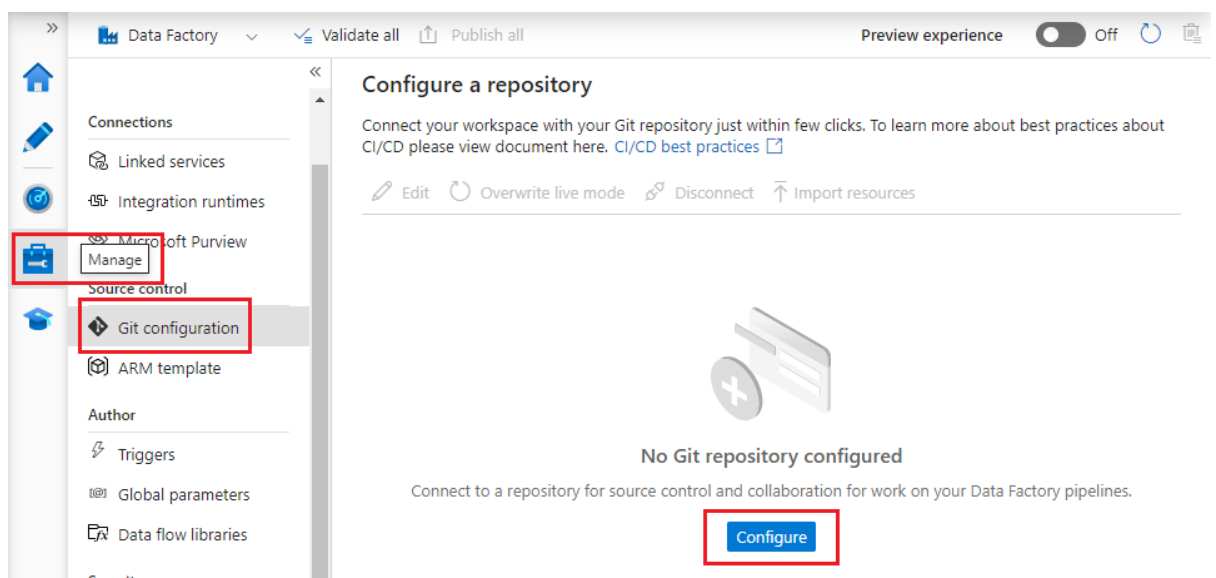
## Azure Data Factory Studio

Launch studio

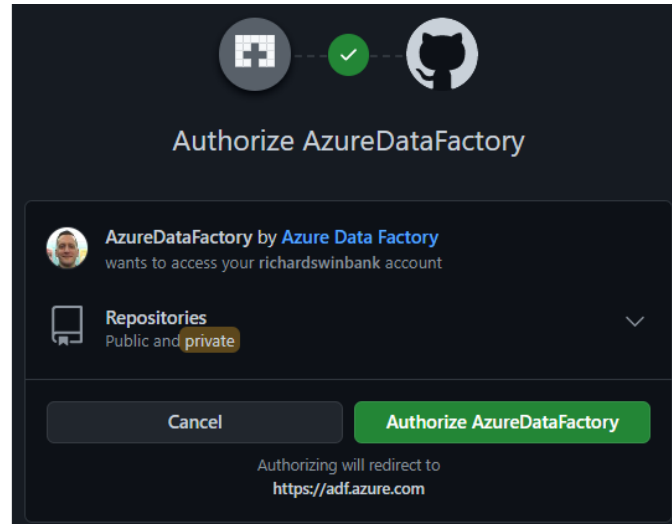
5. Now we’ll connect the new data factory to its GitHub repository. When ADF Studio opens, click the “Set up code repository” button.



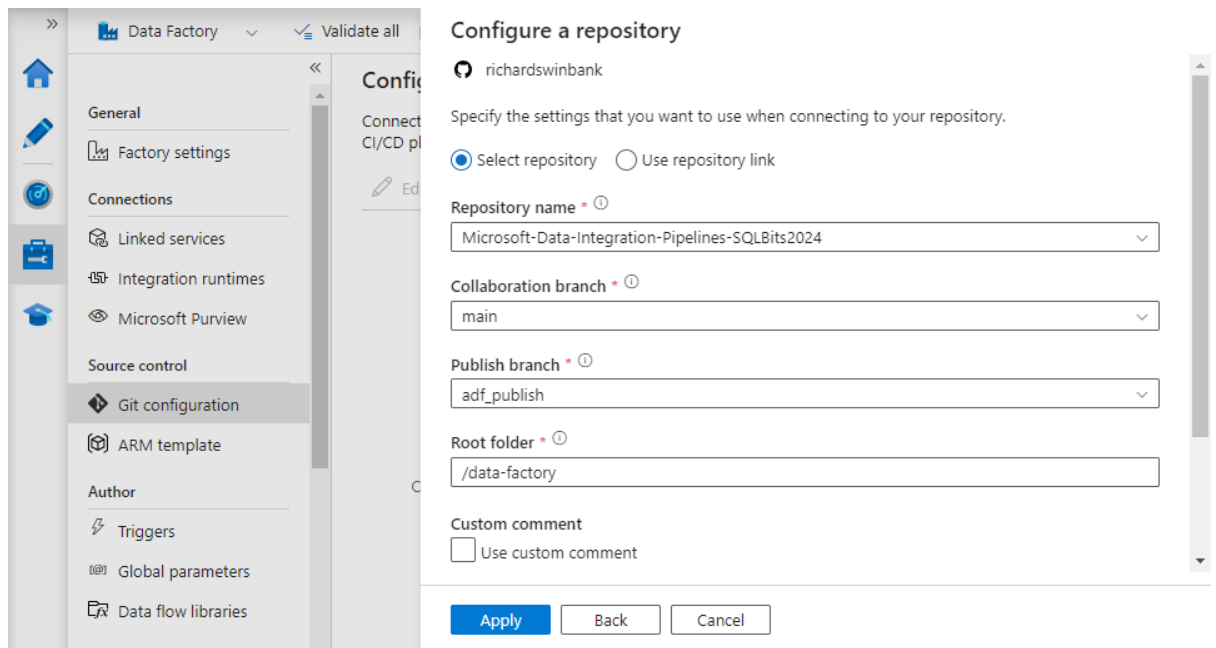
If you don’t see the button, open the management hub using the “Manage” button (toolbox icon) on the leftmost sidebar, then click the “Configure” button on the “Git configuration” page.



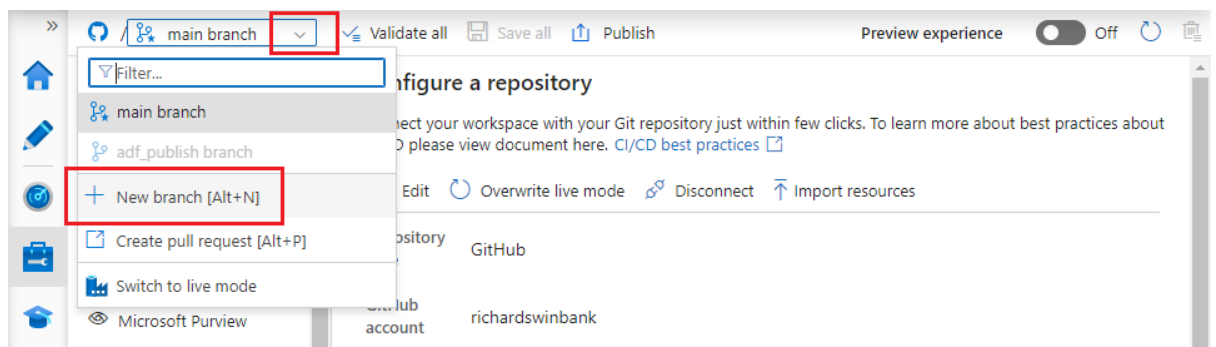
6. Either route in step 5 takes you to the “Configure a repository” flyout. Choose “GitHub” from the “Repository type” dropdown list, then enter your GitHub account name into the “GitHub repository owner” field and click “Continue”.
7. GitHub prompts you to sign in and to authorise the “AzureDataFactory” application. Follow the instructions to sign into your account and confirm authorisation.



8. Once authorised:
  - Select your repository name from the **Repository name** dropdown.
  - Choose a **Collaboration branch**. This is the central codebase – the branch into which all data engineers’ feature branches would be merged – often “main” (or “master” in older repositories).
  - Specify a **Root folder**. This is the repository folder where ADF Studio will save data factory artifacts – you can use the root folder (“/”) if you wish, but specifying a subfolder allows you to store other files in the repository, separate from ADF artifacts.
  - Untick **Use custom comment** – this is for convenience while following these labs.
  - You can leave other settings as they are. Click “Apply”.



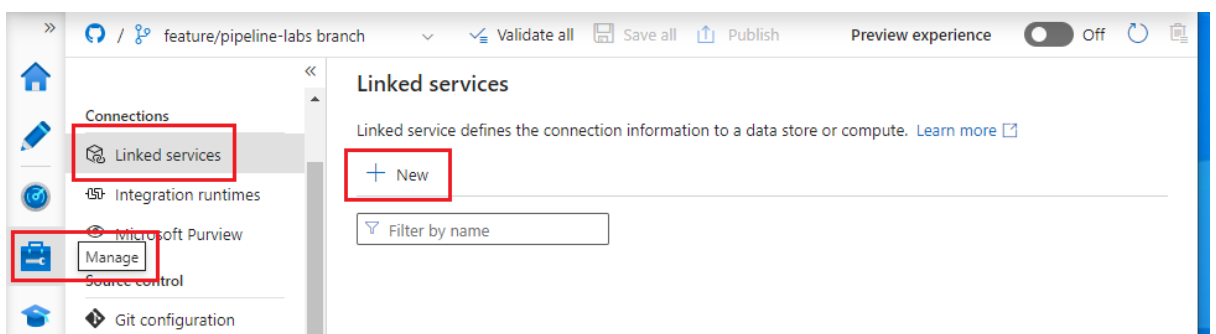
9. ADF Studio sets your initial working branch to your collaboration branch (usually “main”) – as shown in the branch dropdown in the top left. It’s good practice to do development work in a separate feature branch – use the dropdown to create and select a new working branch.



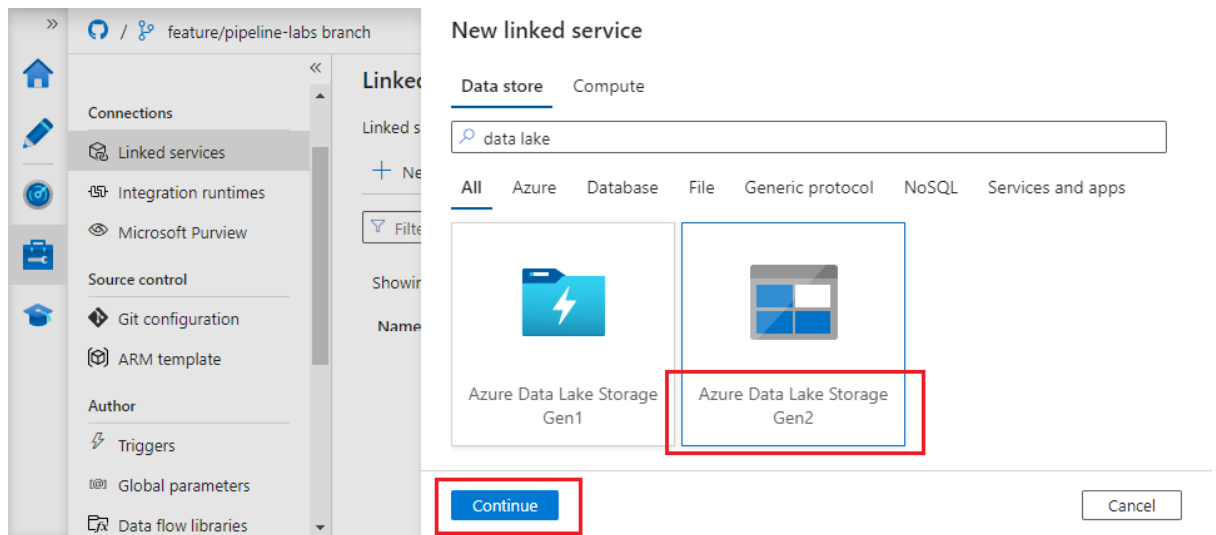
## Lab 1.4 – Connect to the data lake

To enable ADF pipelines to use data in the lake you will need a data factory linked service connection.

1. Navigate to the management hub (as described in Lab 1.3), then open the “Linked services” page and click “+ New”.



2. Search for “data lake”, then choose “Azure Data Lake Storage Gen2” and click “Continue”.

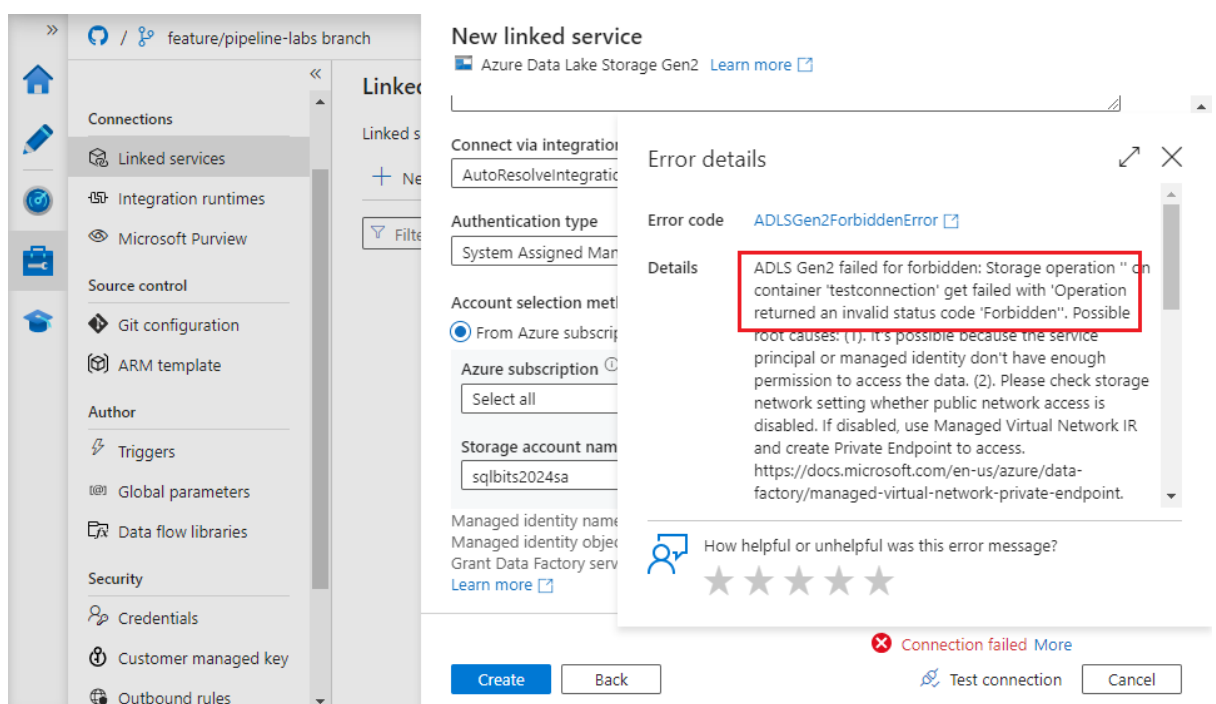


3. Configure linked service details on the “New linked service (Azure Data Lake Storage Gen2)” blade like this:

- Give it a name.
- Choose Authentication type “System Assigned Managed Identity”. The default method (“Account key”) requires extra work to pass keys around securely – a “System-Assigned Managed Identity” is a Microsoft Entra ID service principal, created automatically for your data factory when you created it.
- Choose your data lake storage account from the “Storage account name” dropdown.

4. Click “Test connection” at the bottom of the blade. The connection test will fail, because the factory’s MSI does not have access to the data lake yet – you will receive an error message like “Operation returned an invalid status code ‘Forbidden’”.

You will grant the necessary access in a moment – for now, just click “Create” to save the linked service. You do not need explicitly to save linked service changes – they are automatically committed to your GitHub repository.

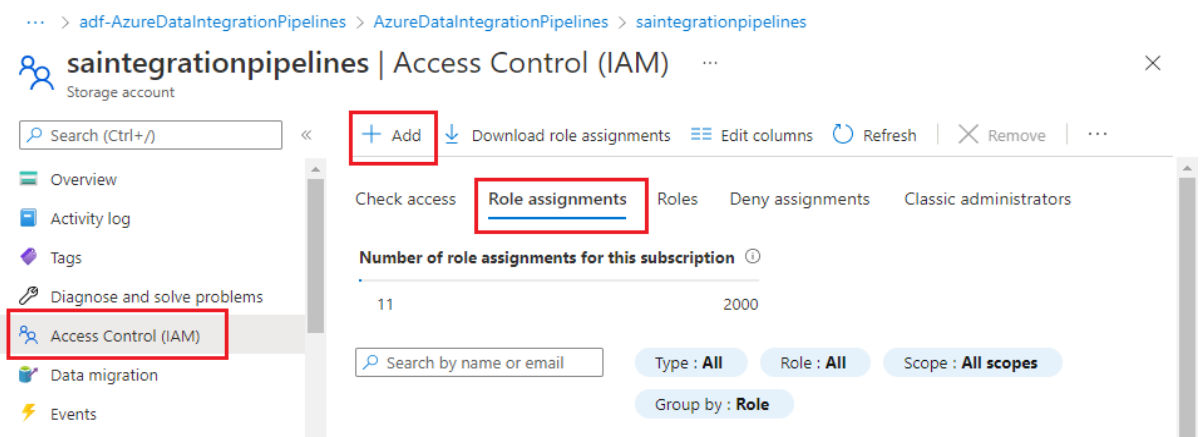




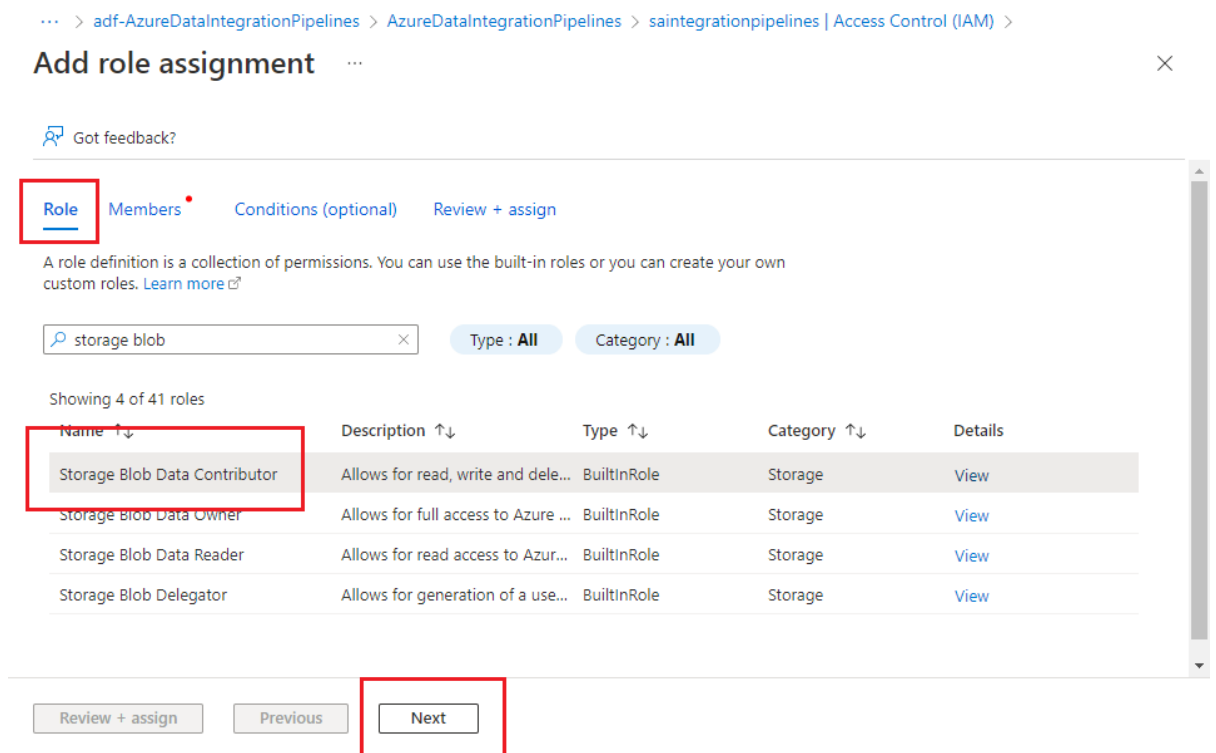
## Lab 1.5 – Grant access to the data lake

You can manage access to Azure resources in the Azure portal. Open a new browser tab to allow you to keep ADF Studio open.

1. Browse to your data lake resource blade – you can find it in the list of resources on the portal home page, or by selecting your resource group to see resources inside it, or by using the search box in the portal's top menu bar.
2. Click “Access control (IAM)” in the storage account (data lake) resource blade, then select the “Role assignments” tab. Click “+ Add” and select “Add role assignment” from the dropdown list.



3. On the “Role” tab of the “Add role assignment” blade, choose the “Storage Blob Data Contributor” role – this authorises read, write and delete access in your data lake – then click “Next”.



- On the “Members” tab, set “Assign access to” to “Managed identity”, then click “+ Select members”.

Home > sqlbits2024sa | Access Control (IAM) >

## Add role assignment

Role **Members** Conditions Review + assign

**Selected role**  
Storage Blob Data Contributor

**Assign access to**  
☐ User, group, or service principal  
☒ **Managed identity**

**Members**  
**+ Select members**

Name	Object ID
No members selected	

**Description**

Review + assign Previous Next **Select** Close

Feedback

### Select managed identities

Some results might be hidden due to your ABAC condition.

Subscription \*  
Azure subscription 1

Managed identity  
Data factory (V2) (9)

Select ⓘ  
Search by name

- aduyexample20240223-shared-aui /subscriptions/6bd2daaf-...
- powerbiprodev-adf /subscriptions/6bd2daaf-...
- qainvestigations-adf

Selected members:  
sqlbits2024-adf /subscriptions/6bd2daaf-... Remove

Choose your Azure Data Factory instance from the “Select managed identities” flyout, click “Select” to dismiss the flyout, then click “Review + assign”. On the “Review + assign” tab, click the button of the same name.

- Return to the management hub in ADF Studio and click on your data lake linked service to re-open the editing flyout. Click “Test connection” again and verify that this time the connection test succeeds. If the test fails, wait a few minutes, then try again – it may take a short time for permission changes to take effect. Click “Cancel” to close the editing blade.

## Recap

In Lab 1 you:

- created an Azure resource group
- created data lake storage
- created an instance of Azure Data Factory
- created and authorised a linked service connection from the factory to your data lake storage.

