

Azure Setup

This article describes the setup in Azure to use User Led Migration to migrate to an azure blob or azure file store in a storage account.

Minimum Permissions:

App Registration Setup

To create the app registration so that ULM can connect to the Storage Accounts in Azure, the following settings need to be configured.

1. Create a standard app registration and configure a client secret.
2. The authentication Redirection URI should be:
<http://localhost:18080/ulm/authorized>
3. Access Tokens and ID tokens should be enabled. Public flows should be disabled, and it should be restricted to the current tenant.
4. Api Permissions should be the following
 - a. **Azure Service Management**
 - i. *user_impersonation* - delegated - this provides the ability to list the storage accounts in the ULM application
 - b. **Azure Storage**
 - i. *user_impersonation* - delegated - this provides access to the storage accounts to upload files.
 - c. **Microsoft Graph** - these are the basic permissions typically given to any app registration
 - i. *email* - delegated - used to get details about the user - view the user's email address
 - ii. *offline_access* - delegated - allows refresh tokens so we can maintain a session longer then the access token timeout
 - iii. *openid* - delegated - allows users to sign in
 - iv. *profile* - delegated - allows access to the user's profile so we know who is doing the migration
 - v. *User.Read* - delegate - ability to sign in and read the user's profile so we know who is doing the migration

⚠️ Granting tenant-wide consent may revoke permissions that have already been granted/tenant-wide for that application. Permissions that users have already granted on their own behalf are

ℹ️ The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for Shinylab

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Azure Service Management (?)				...
user_impersonation	Delegated	Access Azure Resource Manager as organization users	No	✔️ Granted for Shinylab
▼ Azure Storage (?)				...
user_impersonation	Delegated	Access Azure Storage	No	✔️ Granted for Shinylab
▼ Microsoft Graph (?)				...
email	Delegated	View users' email address	No	✔️ Granted for Shinylab
offline_access	Delegated	Maintain access to data you have given it access to	No	✔️ Granted for Shinylab
openid	Delegated	Sign users in	No	✔️ Granted for Shinylab
profile	Delegated	View users' basic profile	No	✔️ Granted for Shinylab
User.Read	Delegated	Sign in and read user profile	No	✔️ Granted for Shinylab

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).



Storage Account Setup

For the storage account to work with ULM, the following needs to be configured.

Blob Storage

The storage account needs to be setup with the following permissions in order for ULM to be able to list the blob storage containers, and to upload files to the blob storage.

Action	Required Role (On Storage Account)
List Blob Containers	Storage Blob Data Contributor
Upload to Blob Container	Storage Blob Data Contributor

File Share Storage

The storage account needs to be setup to allow identity based access to the file shares. This is typically done by linking it to Azure AD, Entra DS or Azure Kerberos. If you plan on using just Entra ID to connect to it, the quickest route is Azure Kerberos.

The following permissions need to be given to the user(s) that are uploading files to the file share.

Action	Required Role (On Storage Account)
List File Shares	Reader
Upload to Azure File Share (using APIs)	Storage File Data Privileged Contributor (this is the required privilege when using the APIs, and Entra ID for authentication)