**Azure Marketplace Setup for Fabric Workload Provisioning**

This document describes the steps required to deploy and use the Azure Function App that creates a Microsoft Fabric Workspace, assigns it a capacity, and deploys a notebook within it using values retrieved securely from Azure Key Vault.

## Prerequisites

Before using the function app, ensure the following:

1. Fabric Capacity: A Fabric capacity of any F SKUs above F16 (recommended)

2. Azure Key Vault

* Create an Azure Key Vault or use an existing one.
* Please note the Permission Model enabled in the key Vault in Access Configurations of the Key Vault

A screenshot of a computer

AI-generated content may be incorrect.

If **Azure RBAC** (recommended) or **Vault access policy** is enabled, manual intervention (Steps are mentioned in Deployment steps #2) is required to assign the Function App’s managed identity appropriate Key Vault roles. This will require **Resource Group Owner** role to the user

* Store the following secrets in the Key Vault:
  + CLIENT\_ID → Client ID of the Service Principal (SPN) which will have required access mentioned in #3.
  + CLIENT\_SECRET → Client Secret of the SPN
  + TENANT\_ID → Tenant ID
  + CAPACITY\_ID → Capacity ID to which workspaces will be assigned.

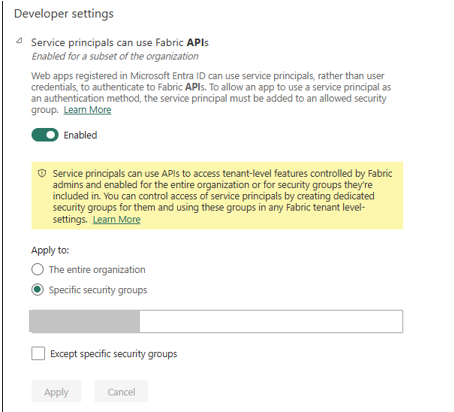
3. Service Principal (SPN) Permissions

Ensure the SPN has the following permissions:

* Fabric Portal Permissions:
  + Admin access to the Capacity to which workspaces will be assigned.
  + Rights to create workspaces in Microsoft Fabric.

A screenshot of a computer

AI-generated content may be incorrect.



* API Permissions (Microsoft Graph or relevant API):
  + Capacity.ReadWrite.All
  + Workspace.ReadWrite.All
* Grant admin consent for these scopes in Azure AD.

3. Azure Resources

* A pre-created Capacity ID in Microsoft Fabric.
* Proper RBAC permissions for the function app to access the Key Vault.

## Deployment Steps

Step 1: Get the offer from Azure Marketplace

* **Configure Offer:** User fills in Function App name, Key Vault, Tenant ID, etc., in the portal UI (from createUiDefinition.json).
* **Review + Create:** Azure Portal validates inputs; user clicks Create.
* ARM Template Runs: mainTemplate.json provisions Function App, App Service Plan, App Settings, Key Vault access, and role assignments.
* **Deploy ZIP Package:** Function App uses WEBSITE\_RUN\_FROM\_PACKAGE to run from the uploaded ZIP.
* **App Starts Running:** Function logic (e.g., Fabric API calls) runs immediately or on trigger.
* **Deployment Complete:** User sees "Deployment succeeded" and can monitor, test, or trigger the Function App.

Step 2: Review the access policies for new function app

1. If Azure RBAC (recommended) is enabled:

* Go to the Key Vault resource on Azure Portal
* Navigate to IAM (Identity Access Management) and add a new assignment for the SPN

A screenshot of a computer

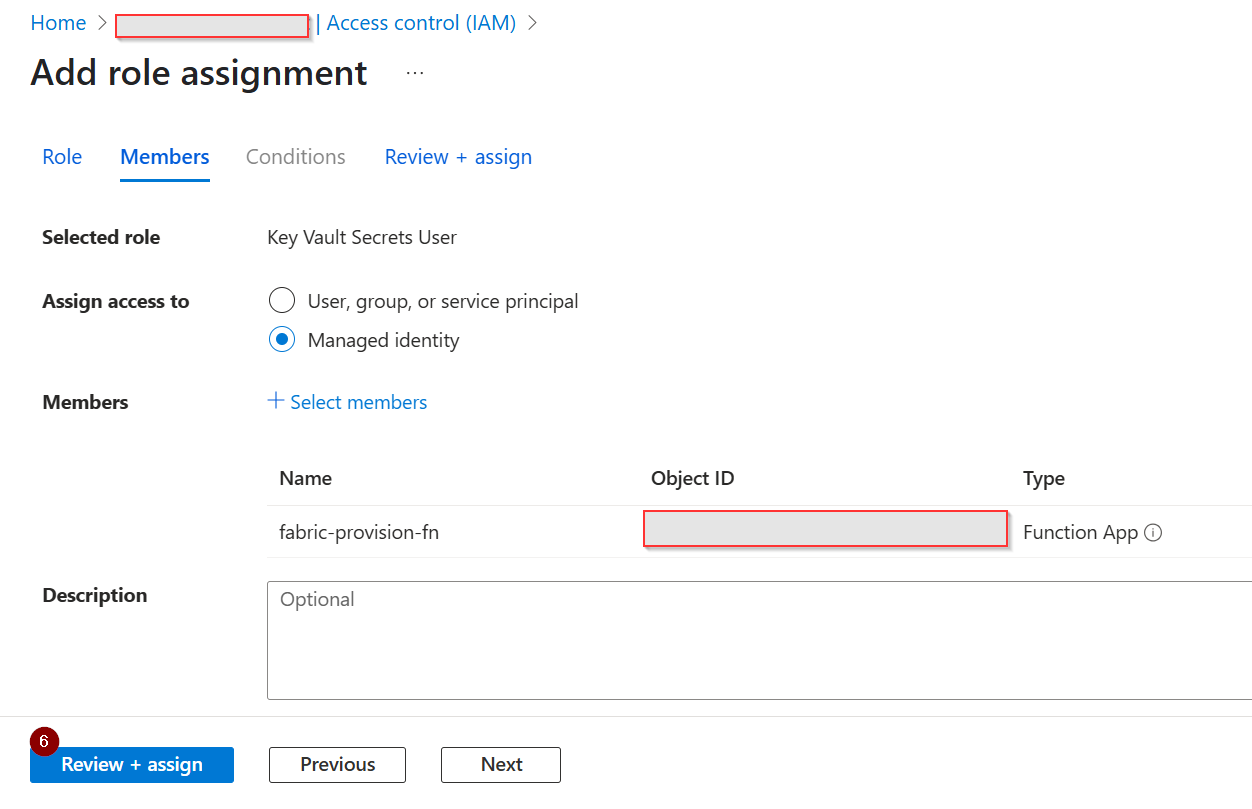
AI-generated content may be incorrect.

* While adding role assignment, assign the access to “Managed Identity” and select the function app name which you wish to deploy in the Members section

A screenshot of a computer

AI-generated content may be incorrect.

* Review the identities and create the role assignment



A screenshot of a computer

AI-generated content may be incorrect.

**Note: We are working on this issue**

1. If **Vault Access Policy** is enabled:
   * Go to the Key Vault resource on Azure Portal
   * Navigate to “Access Policies” and create new one for function app deployed.

A screenshot of a chat

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

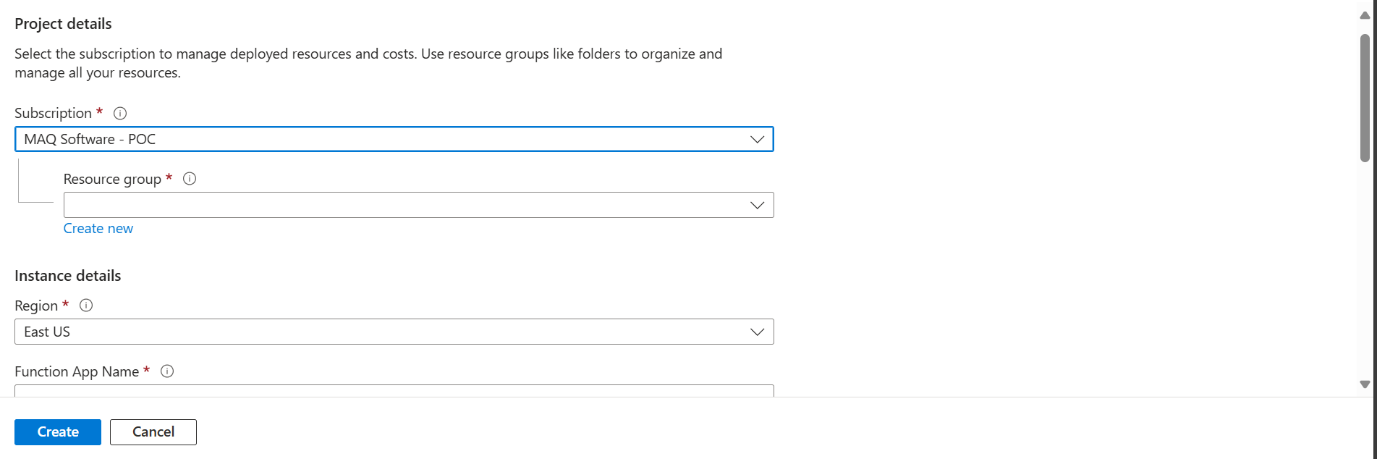
Select function app A screenshot of a computer

AI-generated content may be incorrect.

Create access policy.

## Run Function App

After the function app is deployed and has read permissions on the key Vault, the UI will look like below:



A close-up of a line

AI-generated content may be incorrect.

A close-up of a white background

AI-generated content may be incorrect.

1. Select the Azure Subscription and Resource Group.
2. Enter the below fields which are required to create a function app
   1. Function App Name (name of the function app to be created)
   2. Storage Account Name (name of the storage account used by the Function App for managing triggers, logs, and deployment files – can be existing or new)
   3. Key Vault Name (name of the azure key vault where the secrets are stored)
   4. Tenant ID Secret Name (secret name for tenant ID)
   5. Client ID Secret Name (secret name for client ID)
   6. Client Secret Name (secret name for client secret)
   7. Capacity ID Secret Name (secret name for capacity ID)
   8. Fabric Workspace Name (name of the workspace to be created)
   9. Notebook Name (name of the notebook to be created)
   10. User Email for Workspace Access (user email to provide Member access to the Workspace)

## Function Behaviour

Once triggered (via HTTP request, timer, etc.), the function app will:

1. Authenticate using secrets from the Key Vault.
2. Create a new Microsoft Fabric Workspace with the name provided.
3. Assign the Workspace to the specified Capacity.
4. Create a notebook with the given name inside the workspace.

## Troubleshooting

* 403 Forbidden: Check that the SPN has the correct Graph API scopes and Fabric roles.
* Notebook or workspace not created: Validate input values and ensure the SPN has create permissions.
* Key Vault Access Denied: Confirm correct RBAC assignments to the Key Vault.