**Azure Key Vault for Secrets Management**

Objective: To securely store, access, and manage sensitive information like connection strings, API keys, and certificates using Azure Key Vault.

Steps:

**Setup and Configuration:**

Create an Azure Key Vault instance using the Azure portal, Azure CLI, or ARM templates.

Configure access policies: Determine which users or applications have permissions to access or manage secrets.

**Secrets Management:**

Store a sample secret in the Key Vault (e.g., a database connection string or an API key).

Version secrets: Add a new version of a secret to showcase the versioning capabilities.

**Integrate with a Web Application:**

Create a simple web application. If one exists from a previous project, you can reuse it.

Modify the application to retrieve the secret from Azure Key Vault. This can be done using the Azure SDK for the specific programming language you are using.

Ensure that the application no longer contains hard-coded sensitive information.

**Integrate with Azure DevOps:**

Set up a CI/CD pipeline for the application.

Use Azure DevOps's built-in integration to access secrets from Azure Key Vault during the deployment process. This ensures that secrets are not exposed in the deployment pipeline.

**Logging and Monitoring:**

Enable logging for the Key Vault to monitor access and other activities.

Integrate with Azure Monitor and set up alerts for specific events like unauthorized access attempts.

**Backup and Restore:**

Manually backup your Key Vault.

Restore the Key Vault from the backup, simulating a recovery scenario.

Step 1 – Create Azure SQL Database

Step 2 – Create Azure Key Vault

Step 3 – Store the SQL Password in Key Vault

Step 4 – Create .Net Application in Local and set Connection String with Database. Push Code to Github.

Step 5 – Create .NET App Service (Web App)

Step 6 – Access to Azure Key Vault from Azure App

Step 7 – Set Secrets Github Actions For Azure Web App and save in Github Action secret

Step 8 – Create Github Actions CICD Pipeline

Step 9 – Create App Registration and Federated Credentials

Step 10 – Give Access from Subscription as App registration does not have access to subscription by default.

Step 11 – Monitor Azure Web application and Key vault

**+---------------------+**

**| Azure DevOps CI |**

**| (Build & Release) |**

**+---------------------+**

**|**

**| Fetch secrets at runtime**

**v**

**+---------------------+**

**| Azure Key Vault |**

**| (Secrets, Policies) |**

**+---------------------+**

**^ | |**

**| | |**

**Retrieves Logs/Alerts Backup/Restore**

**via SDK to Azure to Storage**

**Monitor Account**

**| | |**

**| v v**

**+-----------+ +------------------+**

**| Web | | Azure Storage |**

**| Application| | (Backup/Restore) |**

**Step 1 – Create Azure SQL Database**

1. **Go to Azure Portal** → Search for **"SQL Database"** → Click **Create**.
2. **Resource group**: [2tier](https://portal.azure.com/#blade/HubsExtension/ResourceMenuBlade/id/%2Fsubscriptions%2F2ec72b6b-d854-4b60-9f04-9cbe65f37ecc%2FresourceGroups%2F2tier)
3. **Database name**: e.g., databaseN.
4. **Server**: Create new:

Server name: app-sql-server.database.windows.net

Location: Central Canada.

Authentication: **SQL authentication**.

Username: e.g., systemadmin.

Password: Harman@123

1. **Compute + storage**: Choose Basic for testing.
2. A screenshot of a computer

   AI-generated content may be incorrect.Click **Review + Create** → **Create**.

**A screenshot of a computer

AI-generated content may be incorrect.**

**Step 2 – Create Azure Key Vault**

1. In the Azure Portal, search for **Key Vault** → Click **Create**.
2. Select the **same resource group** as your database ([2tier](https://portal.azure.com/#blade/HubsExtension/ResourceMenuBlade/id/%2Fsubscriptions%2F2ec72b6b-d854-4b60-9f04-9cbe65f37ecc%2FresourceGroups%2F2tier)).
3. Name: e.g., HarmanKvault.
4. Type : Access Policy
5. Region: Same as database for performance.
6. A screenshot of a computer

   AI-generated content may be incorrect.Click **Review + Create** → **Create**.

**Step 3 – Store the SQL Password in Key Vault**

1. Open your **Key Vault** → Go to **Secrets** → **Generate/Import**.
2. Name the secret: SqlDbPassword.
3. Value: Enter the SQL admin password you used when creating the database.
4. A screenshot of a computer

   AI-generated content may be incorrect.Click **Create**.

**Step 4 – Create .Net Application in Local and set Connection String with Database. Push Code to Github.**

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 5 – Create .NET App Service (Web App)**

1. In the portal, search for **App Service** → Click **Create**.
2. Resource Group: 2tier.
3. Name: webapp1a3
4. Web app url: webapp1a3.azurewebsites.net
5. Publish: **Code**.
6. Runtime stack: .NET 8 (LTS) (or your preferred version).
7. Operating System: Windows.
8. Region: Same as database.
9. Plan: Create a new free-tier plan for testing.
10. Click **Review + Create** → **Create**.

dotnet new webapp -o demo

dotnet add package Microsoft.Data.SqlClient

dotnet add package System.Data.SqlClient --version 4.8.5

dotnet add package Azure.Identity

dotnet add package Azure.Security.KeyVault.Secrets

A screenshot of a computer

AI-generated content may be incorrect.

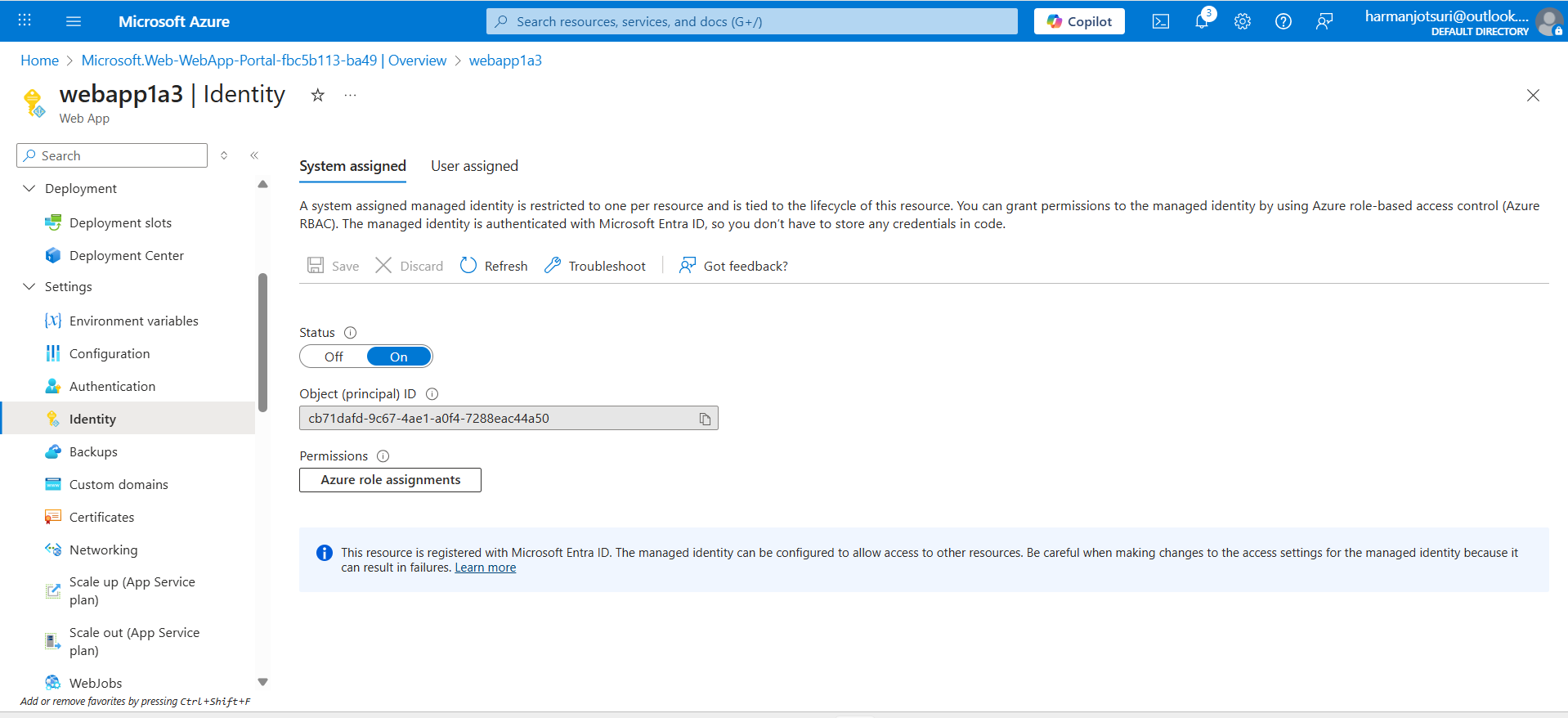
A computer screen with a computer monitor and a coin

AI-generated content may be incorrect.

**Step 6 – Access to Azure Key Vault from Azure App**

We will use Managed Identity.

1. **Go to your Web App** → Identity → System assigned → On → Save.
2. Copy the Object (principal) ID shown: 11f45eec-153a-4670-8b72-68a5f7ea6c6f
3. **Go to your Key Vault** → Access policies → Add Access Policy:
   * Permissions: Under Secret permissions, select Get and List.
   * Principal: Search for your Web App name, select it.
4. Save the policy.



A screenshot of a computer

AI-generated content may be incorrect.

**Step 7 – Set Secrets Github Actions For Azure Web App and save in Github Action secret**

**A screenshot of a computer

AI-generated content may be incorrect.**

Check Configurations of web app

**A close-up of a text

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Step 8 – Create Github Actions CICD Pipeline**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 9 – Create App Registration and Federated Credentials**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 10 – Give Access from Subscription as App registration does not have access to subscription by default.**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Reference: <https://github.com/Azure/login?tab=readme-ov-file#login-with-openid-connect-oidc-recommended>

**Step 11 – Monitor Azure Web application and Key vault**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.