**Project: Secure Access to Azure Key Vault using System-assigned Managed Identity**

**Objective**

* Create a VM with **system-assigned MI**.
* Store an application secret (like a DB password) in **Key Vault**.
* Allow the VM to **retrieve the secret without credentials**.

**Step-by-Step Project Guide**

**Step 1: Create a Resource Group**

1. In Azure Portal → **Resource groups → + Create**.
2. Name it: RG-MI-Project.
3. Choose a region (e.g., East US) → Review + Create.

**Step 2: Create a Virtual Machine**

1. Go to **Virtual Machines → + Create → Virtual machine**.
2. Fill in:
   * Name: MI-VM.
   * Resource group: RG-MI-Project.
   * Image: Ubuntu or Windows Server.
   * Size: Standard\_B2s (or similar).
   * Username/password or SSH key for login.
3. On **Management tab** → **Identity → System-assigned → On**.
4. Click **Review + Create → Create**.

✅ Now VM has its own managed identity in Azure AD.

**Step 3: Create a Key Vault**

1. Go to **Key Vaults → + Create → Key Vault**.
2. Name: kv-miproject.
3. Select the same resource group: RG-MI-Project.
4. Review + Create.

**Step 4: Add a Secret in Key Vault**

1. Open kv-miproject.
2. Go to **Secrets → + Generate/Import**.
3. Name: DBPassword.
4. Value: MySecurePassword123!.
5. Click **Create**.

**Step 5: Assign Key Vault Access to the VM’s Identity**

1. In Key Vault → **Access Control (IAM)** → + Add role assignment.
2. Role: **Key Vault Secrets User** (or Key Vault Secrets Officer if you need more).
3. Assign access to: **Managed identity**.
4. Select your VM MI-VM.
5. Save.

✅ Now the VM’s managed identity can read secrets.

**Step 6: Test from the VM**

1. Go to **Virtual Machine → Connect → RDP/SSH** into the VM.
2. Run these commands:

**If Linux VM:**

#Install AzureCLI

apt install azure-cli

#upgrade AzureCLI

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

# Login with system-assigned MI

az login –identity

# Verify login

az account show

Retrieve the secret

az keyvault secret show --name DBPassword --vault-name kv-miproject --query value -o tsv

✅ Expected output:

MySecurePassword123!

**Final Architecture**

* **VM (System-assigned MI)** → **Azure AD (authenticates identity)** → **Key Vault (retrieves secrets)**.

**Learning Outcomes**

* Enable **System-assigned Managed Identity**.
* Store and secure secrets in **Key Vault**.
* Assign **RBAC permissions** to managed identity.
* Retrieve secrets without any passwords or keys.

**2.**Top of Form**Extended Project: Test Isolation with Multiple VMs**

**Objective**

* Create a **second VM**.
* Only the **first VM** has Key Vault access.
* Verify that the second VM **cannot read secrets** until permissions are granted.

**Steps**

**Step 1: Create a Second VM**

1. In Azure Portal → **Virtual Machines → + Create VM**.
2. Name: MI-VM2.
3. Use same Resource Group: RG-MI-Project.
4. In **Management tab** → Identity → **System-assigned → On**.
5. Finish and create.

✅ Now both MI-VM and MI-VM2 have **separate managed identities** in Azure AD.

**Step 2: Check Key Vault Role Assignments**

1. Go to **Key Vault (kv-miproject) → Access Control (IAM) → Role Assignments**.
2. Confirm that only **MI-VM** has **Key Vault Secrets User** role.
3. MI-VM2 should **not** have any access.

**Step 3: Test Access from VM2**

1. Connect to MI-VM2.
2. Run:

az login --identity

az account show

az keyvault secret show --name DBPassword --vault-name kv-miproject --query value -o tsv

1. Expected output ❌:

ERROR: (Forbidden) Client <ObjectID\_of\_MI-VM2> does not have permission to perform action...

This proves **VM2 cannot read secrets** because it lacks access.

**Step 4: Grant Access to VM2**

1. Go to Key Vault → **Access Control (IAM) → + Add Role Assignment**.
2. Select role: **Key Vault Secrets User**.
3. Assign it to **MI-VM2 (managed identity)**.
4. Save.

**Step 5: Retest from VM2**

Now in MI-VM2:

az login --identity

az keyvault secret show --name DBPassword --vault-name kv-miproject --query value -o tsv

✅ Expected output:

MySecurePassword123!

**Learning Outcome**

* Each VM’s **system-assigned managed identity is unique**.
* Permissions must be explicitly granted per resource identity.
* This prevents unintended access → **principle of least privilege**.Bottom of Form

**🔹 Extended Project: Use a User-assigned Managed Identity (UAMI) Across Two VMs**

**Objective**

* Create a **User-assigned Managed Identity (UAMI)**.
* Attach it to both VMs (MI-VM1 and MI-VM2).
* Give only the UAMI access to **Key Vault**.
* Test that both VMs can now read secrets (using UAMI), while their **system-assigned MIs are still isolated**.

**Steps**

**Step 1: Create a User-assigned Managed Identity**

1. In Azure Portal → **Managed Identities → + Create**.
2. Name: AppIdentity.
3. Resource group: RG-MI-Project.
4. Region: same as your VMs.
5. Click **Create**.

✅ Now you have a reusable managed identity called AppIdentity.

**Step 2: Assign UAMI to Both VMs**

1. Go to **MI-VM1 → Identity → User-assigned → + Add**.
2. Select AppIdentity → Save.
3. Repeat the same for **MI-VM2**.

✅ Now both VMs have **two identities** each:

* Their own **System-assigned MI**.
* The shared **User-assigned MI (AppIdentity)**.

**Step 3: Grant UAMI Access to Key Vault**

1. Go to **Key Vault (kv-miproject) → Access Control (IAM)**.
   * Add Role Assignment.
2. Role: **Key Vault Secrets User**.
3. Assign access to: **AppIdentity (user-assigned MI)**.
4. Save.

✅ Only the **User-assigned identity** can now access secrets.

**Step 4: Test from VM1**

On **MI-VM1**:

# Login using system-assigned MI (should still fail, unless already granted)

az login --identity

az keyvault secret show --name DBPassword --vault-name kv-miproject --query value -o tsv

❌ Expected: Forbidden (if you didn’t grant VM1 system MI access earlier).

Now login with **UAMI** instead:

az login --identity --username <ClientID\_of\_AppIdentity>

az keyvault secret show --name DBPassword --vault-name kv-miproject --query value -o tsv

✅ Expected output:

MySecurePassword123!

**Step 5: Test from VM2**

On **MI-VM2** (same test):

az login --identity --username <ClientID\_of\_AppIdentity>

az keyvault secret show --name DBPassword --vault-name kv-miproject --query value -o tsv

✅ Expected output:

MySecurePassword123!

**Learning Outcome**

* **System-assigned MI** = unique per resource (isolation).
* **User-assigned MI** = reusable and can be shared across multiple resources.
* You can **mix both**:
  + Use System-assigned MI for per-resource security.
  + Use UAMI for shared scenarios (clusters, apps, pipelines).

