Azure Key Vault Creation and Secret Storage

\*\*Document Name\*\*: Azure\_KeyVault\_Secret\_Guide.docx

## Overview

This guide details the end-to-end process to:

1. Create a service principal with the `Owner` role.

2. Register the `Microsoft.KeyVault` provider.

3. Create an Azure Key Vault (`myKeyVault5698`) in the subscription `b2531278-0a44-46ad-a260-a86bf509f256` and resource group `myResourceGroup`.

4. Store a secret named `mySecret` with the value `mySecretValue` using an inline `curl` command.

5. Verify the secret.

\*\*Key Details\*\*:

- \*\*Service Principal\*\*: `myServicePrincipal`

- Application (client) ID: `aa9ee733-789b-4015-b066-61bb6336e1ac`

- Object ID: `6d00398f-d49b-47e6-9f88-212624a6a62f`

- Client Secret: `DKf8Q~1upNRqyhiAbynpcPddShCyxf~VvAScoavl`

- Tenant ID: `7d656ff7-cdc6-4cdb-9312-2a385eb6b15a`

- \*\*Key Vault\*\*: `myKeyVault5698` (chosen after `myKeyVault` was in use)

- \*\*Scope Distinction\*\*:

- Management plane (create vault): `https://management.azure.com`

- Data plane (store secret): `https://vault.azure.net`

- \*\*Issues Addressed\*\*:

- Vault name conflict (`VaultAlreadyExists`).

- Missing `accessPolicies` (`BadRequest`).

- Incorrect token audience (`AKV10022: Invalid audience`).

---

## Prerequisites

- \*\*Azure Subscription\*\*: `b2531278-0a44-46ad-a260-a86bf509f256`

- \*\*Resource Group\*\*: `myResourceGroup`

- \*\*Tools\*\*: Azure CLI, `curl`, `jq` (optional for parsing JSON).

- \*\*Permissions\*\*: Service principal must have `Owner` or `Contributor` role on the resource group for management operations and `set` permission for secrets in the Key Vault’s access policy.

---

## Step-by-Step Process

### Step 1: Create Service Principal

Create a service principal `myServicePrincipal` with the `Owner` role to manage Key Vault resources.

\*\*Command\*\*:

```bash

az ad sp create-for-rbac --name "myServicePrincipal" --role Owner --scopes /subscriptions/b2531278-0a44-46ad-a260-a86bf509f256/resourceGroups/myResourceGroup

Output:

json

Copy

{

"appId": "aa9ee733-789b-4015-b066-61bb6336e1ac",

"displayName": "myServicePrincipal",

"password": "DKf8Q~1upNRqyhiAbynpcP Verify this is correct and secure it~VvAScoavl",

"tenant": "7d656ff7-cdc6-4cdb-9312-2a385eb6b15a"

}

Details:

Application ID: aa9ee733-789b-4015-b066-61bb6336e1ac

Client Secret: DKf8Q~1upNRqyhiAbynpcPddShCyxf~VvAScoavl (secure this; avoid exposure).

Tenant ID: 7d656ff7-cdc6-4cdb-9312-2a385eb6b15a

Object ID: 6d00398f-d49b-47e6-9f88-212624a6a62f (used in access policies).

Role: Owner on resource group, sufficient for Key Vault creation and management.

Security Note: Protect the client secret as per Azure CLI warning. Store in a secure location (e.g., environment variable, secret manager).

Step 2: Register Microsoft.KeyVault Provider

Register the Microsoft.KeyVault provider to enable Key Vault operations.

Command:

bash

Copy

az provider register --namespace Microsoft.KeyVault

Verify Registration:

bash

Copy

az provider show -n Microsoft.KeyVault --query registrationState

Expected Output: "Registered"

Details:

If "Registering", wait a few minutes and recheck.

Ensures subscription b2531278-0a44-46ad-a260-a86bf509f256 can create Key Vaults.

Step 3: Create Key Vault

Create the Key Vault myKeyVault5698 after resolving the VaultAlreadyExists error with myKeyVault (likely soft-deleted) and adding accessPolicies to fix BadRequest.

Get Management Token (scope: https://management.azure.com):

bash

Copy

mgmt\_token=$(curl -X POST -d "grant\_type=client\_credentials&client\_id=aa9ee733-789b-4015-b066-61bb6336e1ac&client\_secret=DKf8Q~1upNRqyhiAbynpcPddShCyxf~VvAScoavl&resource=https://management.azure.com" "https://login.microsoftonline.com/7d656ff7-cdc6-4cdb-9312-2a385eb6b15a/oauth2/token" | jq -r .access\_token)

Create Key Vault:

bash

Copy

curl -X PUT -H "Authorization: Bearer $mgmt\_token" -H "Content-Type: application/json" -d '{"location":"eastus","properties":{"sku":{"family":"A","name":"standard"},"tenantId":"7d656ff7-cdc6-4cdb-9312-2a385eb6b15a","accessPolicies":[{"tenantId":"7d656ff7-cdc6-4cdb-9312-2a385eb6b15a","objectId":"6d00398f-d49b-47e6-9f88-212624a6a62f","permissions":{"keys":["get","list","update","create","import","delete","recover","backup","restore"],"secrets":["get","list","set","delete","recover","backup","restore"],"certificates":["get","list","update","create","import","delete","recover","backup","restore"]}}],"enabledForDeployment":true,"enabledForDiskEncryption":true,"enabledForTemplateDeployment":true,"enableSoftDelete":true,"enablePurgeProtection":true}}' "https://management.azure.com/subscriptions/b2531278-0a44-46ad-a260-a86bf509f256/resourceGroups/myResourceGroup/providers/Microsoft.KeyVault/vaults/myKeyVault5698?api-version=2019-09-01"

Output:

json

Copy

{

"id": "/subscriptions/b2531278-0a44-46ad-a260-a86bf509f256/resourceGroups/myResourceGroup/providers/Microsoft.KeyVault/vaults/myKeyVault5698",

"name": "myKeyVault5698",

"type": "Microsoft.KeyVault/vaults",

"location": "eastus",

"tags": {},

"properties": {

"sku": { "family": "A", "name": "standard" },

"tenantId": "7d656ff7-cdc6-4cdb-9312-2a385eb6b15a",

"accessPolicies": [

{

"tenantId": "7d656ff7-cdc6-4cdb-9312-2a385eb6b15a",

"objectId": "6d00398f-d49b-47e6-9f88-212624a6a62f",

"permissions": {

"keys": ["get", "list", "update", "create", "import", "delete", "recover", "backup", "restore"],

"secrets": ["get", "list", "set", "delete", "recover", "backup", "restore"],

"certificates": ["get", "list", "update", "create", "import", "delete", "recover", "backup", "restore"]

}

}

],

"enabledForDeployment": true,

"enabledForDiskEncryption": true,

"enabledForTemplateDeployment": true,

"enableSoftDelete": true,

"enablePurgeProtection": true,

"vaultUri": "https://myKeyVault5698.vault.azure.net",

"provisioningState": "RegisteringDns"

}

}

Details:

Endpoint: PUT https://management.azure.com/.../myKeyVault5698?api-version=2019-09-01

Scope: https://management.azure.com (management plane).

Access Policy: Grants set permission for secrets to Object ID 6d00398f-d49b-47e6-9f88-212624a6a62f.

State: RegisteringDns (DNS propagation may take minutes; likely complete by May 14, 2025, 01:23 AM EDT).

Verify Creation:

bash

Copy

az keyvault show --name myKeyVault5698 --resource-group myResourceGroup --subscription b2531278-0a44-46ad-a260-a86bf509f256 --query properties.provisioningState

Expected Output: "Succeeded"

Step 4: Verify Access Policy

Ensure the service principal has set permission for secrets.

Command:

bash

Copy

az keyvault show --name myKeyVault5698 --resource-group myResourceGroup --subscription b2531278-0a44-46ad-a260-a86bf509f256 --query properties.accessPolicies

Expected Output (partial):

json

Copy

[

{

"objectId": "6d00398f-d49b-47e6-9f88-212624a6a62f",

"permissions": {

"secrets": ["get", "list", "set", "delete", "recover", "backup", "restore"],

...

},

...

}

]

Update if Needed:

bash

Copy

az keyvault set-policy --name myKeyVault5698 --resource-group myResourceGroup --subscription b2531278-0a44-46ad-a260-a86bf509f256 --object-id 6d00398f-d49b-47e6-9f88-212624a6a62f --secret-permissions get list set delete recover backup restore

Step 5: Obtain Access Token for Data Plane

The data plane API (for storing secrets) requires a token with scope https://vault.azure.net, unlike the management plane’s https://management.azure.com. The AKV10022: Invalid audience error was due to using the wrong scope.

Command:

bash

Copy

access\_token=$(curl -X POST -d "grant\_type=client\_credentials&client\_id=aa9ee733-789b-4015-b066-61bb6336e1ac&client\_secret=DKf8Q~1upNRqyhiAbynpcPddShCyxf~VvAScoavl&resource=https://vault.azure.net" "https://login.microsoftonline.com/7d656ff7-cdc6-4cdb-9312-2a385eb6b15a/oauth2/token" | jq -r .access\_token)

Details:

Scope: https://vault.azure.net (data plane).

Tool: jq extracts the token. Without jq, manually copy access\_token from the JSON response.

Alternative (Azure CLI):

bash

Copy

az login --service-principal -u aa9ee733-789b-4015-b066-61bb6336e1ac -p DKf8Q~1upNRqyhiAbynpcPddShCyxf~VvAScoavl --tenant 7d656ff7-cdc6-4cdb-9312-2a385eb6b15a

access\_token=$(az account get-access-token --resource https://vault.azure.net --query accessToken -o tsv)

Step 6: Store Secret

Store a secret mySecret with value mySecretValue using the data plane API and the https://vault.azure.net scope.

JSON Payload:

Store Secret Request

json

Show inline

Inline curl Command:

bash

Copy

curl -X PUT -H "Authorization: Bearer $access\_token" -H "Content-Type: application/json" -d '{"value":"mySecretValue","contentType":"text/plain","attributes":{"enabled":true}}' "https://myKeyVault5698.vault.azure.net/secrets/mySecret?api-version=7.4"

Details:

Endpoint: https://myKeyVault5698.vault.azure.net/secrets/mySecret?api-version=7.4

Scope: https://vault.azure.net (data plane, avoids AKV10022 error).

API Version: 7.4 (current as of May 2025).

Expected Output:

json

Copy

{

"value": "mySecretValue",

"id": "https://myKeyVault5698.vault.azure.net/secrets/mySecret/<version>",

"attributes": {

"enabled": true,

"created": "<timestamp>",

"updated": "<timestamp>",

"recoveryLevel": "Recoverable+Purgeable"

},

"contentType": "text/plain"

}

Step 7: Verify Secret

Confirm the secret was stored.

Command:

bash

Copy

az keyvault secret show --vault-name myKeyVault5698 --name mySecret --subscription b2531278-0a44-46ad-a260-a86bf509f256

Expected Output (partial):

json

Copy

{

"value": "mySecretValue",

"id": "https://myKeyVault5698.vault.azure.net/secrets/mySecret/<version>",

"attributes": { "enabled": true, ... },

"contentType": "text/plain"

}

Troubleshooting

DNS Propagation:

If https://myKeyVault5698.vault.azure.net is inaccessible, check:

bash

Copy

az keyvault show --name myKeyVault5698 --resource-group myResourceGroup --subscription b2531278-0a44-46ad-a260-a86bf509f256 --query properties.provisioningState

Wait if "RegisteringDns".

401 Unauthorized:

Verify https://vault.azure.net scope.

Regenerate token if expired:

bash

Copy

access\_token=$(curl -X POST -d "grant\_type=client\_credentials&client\_id=aa9ee733-789b-4015-b066-61bb6336e1ac&client\_secret=DKf8Q~1upNRqyhiAbynpcPddShCyxf~VvAScoavl&resource=https://vault.azure.net" "https://login.microsoftonline.com/7d656ff7-cdc6-4cdb-9312-2a385eb6b15a/oauth2/token" | jq -r .access\_token)

403 Forbidden:

Confirm set permission in access policy.

Verify Object ID:

bash

Copy

az ad sp show --id aa9ee733-789b-4015-b066-61bb6336e1ac --query id

400 Bad Request:

Check JSON syntax or API version 7.4.

Vault Name Conflict:

Check soft-deleted vaults:

bash

Copy

az keyvault list-deleted --subscription b2531278-0a44-46ad-a260-a86bf509f256

Purge if needed:

bash

Copy

az keyvault purge --subscription b2531278-0a44-46ad-a260-a86bf509f256 --name myKeyVault5698

Notes

Scope Distinction: Management plane (https://management.azure.com) for creating the vault; data plane (https://vault.azure.net) for secret operations.

Security: Protect the client secret. Use environment variables or a secret manager in production.

Role: Owner is broad; consider Key Vault Contributor for production.

API Versions: Management: 2019-09-01; Data plane: 7.4.