

Azure Key Vault

Azure Key Vault is a cloud service provided by Microsoft Azure that allows you to securely store and manage sensitive information such as keys, secrets, certificates, and connection strings. Here's a brief overview of Azure Key Vault:

1. **Secure Storage:** Azure Key Vault provides a centralized and secure storage solution for cryptographic keys, secrets, and certificates. It helps safeguard sensitive information from unauthorized access and provides robust security features such as access policies, access logging, and auditing.
2. **Key Management:** Key Vault enables you to generate, import, and manage cryptographic keys used for encryption, decryption, signing, and verification of data. It supports various types of keys, including RSA, AES, and elliptic curve keys, and offers key rotation and versioning capabilities to enhance security.
3. **Secrets Management:** Key Vault allows you to securely store and manage secrets such as passwords, API keys, and connection strings. It provides features for automatic rotation of secrets, versioning, and granular access control to ensure compliance with security policies.
4. **Certificate Management:** Key Vault supports the management of X.509 certificates used for securing applications, websites, and communication channels. It offers features for certificate lifecycle management, including certificate issuance, renewal, and revocation, and integrates seamlessly with Azure services like Azure App Service and Azure Kubernetes Service.
5. **Integration with Azure Services:** Key Vault seamlessly integrates with various Azure services, enabling secure access to keys, secrets, and certificates from applications and services running in Azure. It provides client libraries, SDKs, and REST APIs for programmatic access to stored secrets and keys.
6. **Compliance and Governance:** Key Vault helps organizations meet compliance requirements and adhere to security best practices by offering features such as access policies, role-based access control (RBAC), and audit logging. It supports compliance certifications like SOC, ISO, PCI DSS, and HIPAA.
7. **Developer Productivity:** Key Vault enhances developer productivity by providing easy-to-use APIs and client libraries for accessing stored secrets and keys from applications and services. It simplifies key management tasks, such as encryption and decryption, and helps mitigate security risks associated with hardcoding sensitive information in code.

The end goal of this lab is to demonstrate how to securely access and retrieve secrets from Azure Key Vault using an application registered in Azure Active Directory (Azure AD). This setup ensures that sensitive information, such as database passwords, is securely managed and accessed, adhering to best practices for security and compliance.

 **To begin with the Lab:**

1. Now, in this lab we will go through a program that can be used to fetch a secret from the Azure Key Vault Service.
2. But first we need to create a Key Vault service. In the marketplace search for Key vault and choose this service accordingly.

The screenshot shows the Azure Marketplace search results for 'Key Vault'. At the top, there is an orange icon with a key symbol. Next to it, the text 'Key Vault' is displayed in a large, bold, dark blue font. To the right of the title is a blue 'Add to Favorites' button with a heart icon. Below the title, it says 'Microsoft | Azure Service'. Underneath that, a rating of '★ 4.1 (687 ratings)' is shown. A 'Plan' section follows, with a dropdown menu set to 'Key Vault' and a blue 'Create' button to its right.

3. Now you need to choose your resource group and then give your vault a name after that choose your region.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Azure Pass - Sponsorship
Resource group *	demo-resource-group
	Create new

Instance details

Key vault name * ⓘ	demoVault120
Region *	North Europe
Pricing tier * ⓘ	Standard

Soft-delete ⓘ

Enabled

Days to retain deleted vaults * ⓘ	7
-----------------------------------	---

Purge protection ⓘ

Disable purge protection (allow key vault and objects to be purged during retention period)

Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

4. Then in the access configuration choose vault access policy and move to review page and create you vault.

Basics Access configuration Networking Tags Review + create

Configure data plane access for this key vault

To access a key vault in data plane, all callers (users or applications) must have proper authentication and authorization. Authentication establishes the identity of the caller. Authorization determines which operations the caller can execute. [Learn more](#)

Permission model

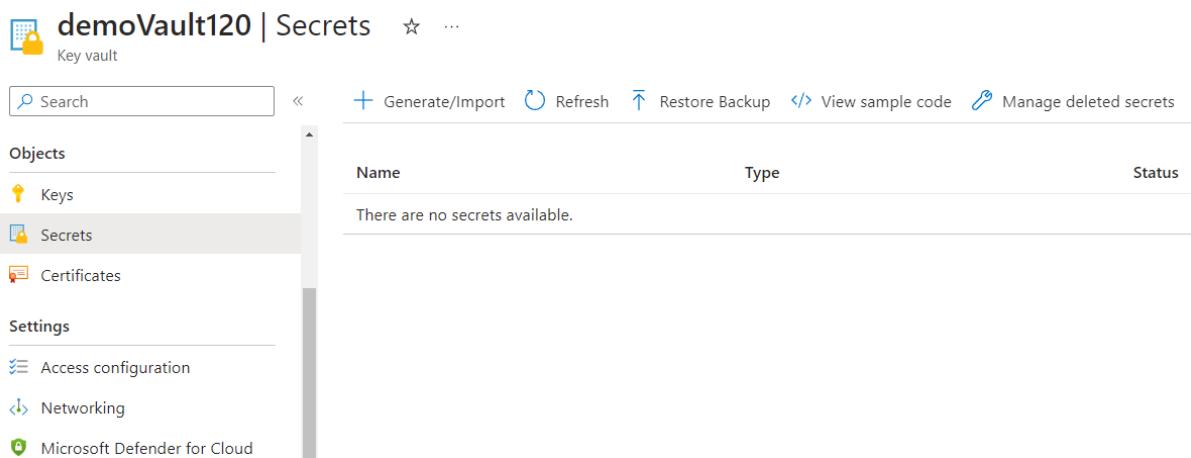
Grant data plane access by using a [Azure RBAC](#) or [Key Vault access policy](#)

- Azure role-based access control (recommended) [?](#)
 Vault access policy [?](#)

Resource access

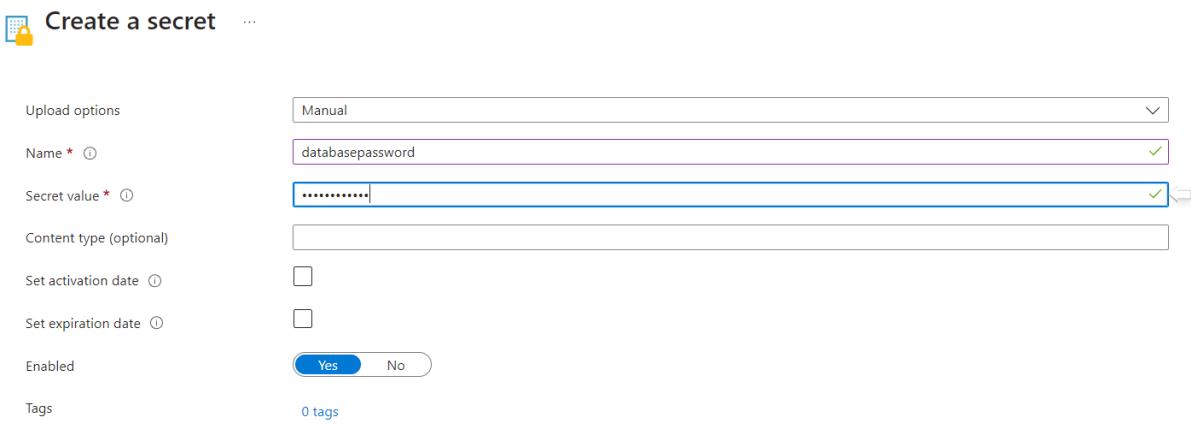
- Azure Virtual Machines for deployment [?](#)
 Azure Resource Manager for template deployment [?](#)
 Azure Disk Encryption for volume encryption [?](#)

5. In your Vault go to secrets and generate a secret.



The screenshot shows the 'demoVault120 | Secrets' page in the Azure portal. The left sidebar has 'Objects' expanded, showing 'Keys', 'Secrets' (which is selected), and 'Certificates'. The main area has a search bar and several action buttons: '+ Generate/Import', 'Refresh', 'Restore Backup', 'View sample code', and 'Manage deleted secrets'. A table below shows a single row: 'Name' (empty), 'Type' (empty), and 'Status' (empty). A message says 'There are no secrets available.'

6. Let's suppose you are storing a Database password here, so give it a name and give your secret value as your password and then hit on create button.



The screenshot shows the 'Create a secret' form. The 'Name' field is set to 'databasepassword' and the 'Secret value' field contains '.....'. Other fields include 'Content type (optional)', 'Set activation date', 'Set expiration date', 'Enabled' (set to 'Yes'), and 'Tags' (0 tags).

7. Now from GitHub you need to download a zip file named secret app and open it in Visual Studio 2022.
8. In this application we are going to make some changes, the first is to change the key vault URL and the second is to change the secret name.

```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help | Search | secretapp
secretapp Program.cs
1  using Azure.Identity;
2  using Azure.Security.KeyVault.Secrets;
3
4  string tenantId = "79c0f6d9-7f3b-4425-a6b6-09b47643ec58";
5  string clientId = "8b34a56b-84b1-4117-a852-86ec90b7a403";
6  string clientSecret = "EaM8Q-eqZ5M0n6hEUtQ.vBJVf-G5p0Yea-gbc9v";
7
8  // Given the application only the Get permission on the secret
9
10 string keyvaultUrl = "https://appvault55343.vault.azure.net/";
11 string secretName = "datapassword";
12
13 ClientSecretCredential clientSecretCredential = new ClientSecretCredential(tenantId, clientId, clientSecret);
14 SecretClient secretClient = new SecretClient(new Uri(keyvaultUrl), clientSecretCredential);
15
16 var secret = secretClient.GetSecret(secretName);
17
18 string dbpassword = secret.Value.Value;
19 Console.WriteLine(dbpassword);
20

```

9. To get this you need to go to an overview of your key vault service you can see the Vault URL copy it and paste in the application.

	Delete		Move		Refresh		Open in mobile
<hr/>							
^ Essentials							
Resource group (move) : demo-resource-group Location : North Europe Subscription (move) : Azure Pass - Sponsorship Subscription ID : 6e13e5d6-4287-42a8-b80f-91d6b14e3aec				Vault URI : https://demovault120.vault.azure.net/ Sku (Pricing tier) : Standard Directory ID : bc45c375-f8b5-420b-9bae-325d48b59d33 Directory Name : CloudFreaks Soft-delete : Enabled Purge protection : Disabled			

10. Now to get the tenant ID and client ID we can make use of our previous app registration in Microsoft Entra ID. Below you can see the client ID and the tenant ID just copy and paste them into your application.

<hr/>							
^ Essentials							
Display name : demoblobapp Application (client) ID : a87daca8-234d-49cb-b6a0-78cdffbb9e53 Object ID : 9f7a6425-cf74-4ddf-ba10-26d58302a8eb Directory (tenant) ID : bc45c375-f8b5-420b-9bae-325d48b59d33 Supported account types : My organization only				Client credentials : 0 certificate, 1 secret Redirect URIs : Add a Redirect URI Application ID URI : Add an Application ID URI Managed application in I... : demoblobapp			

11. But for the client certificate we need to create a new one, for that in you app registration go to certificate and secrets and choose to create a new client secret.

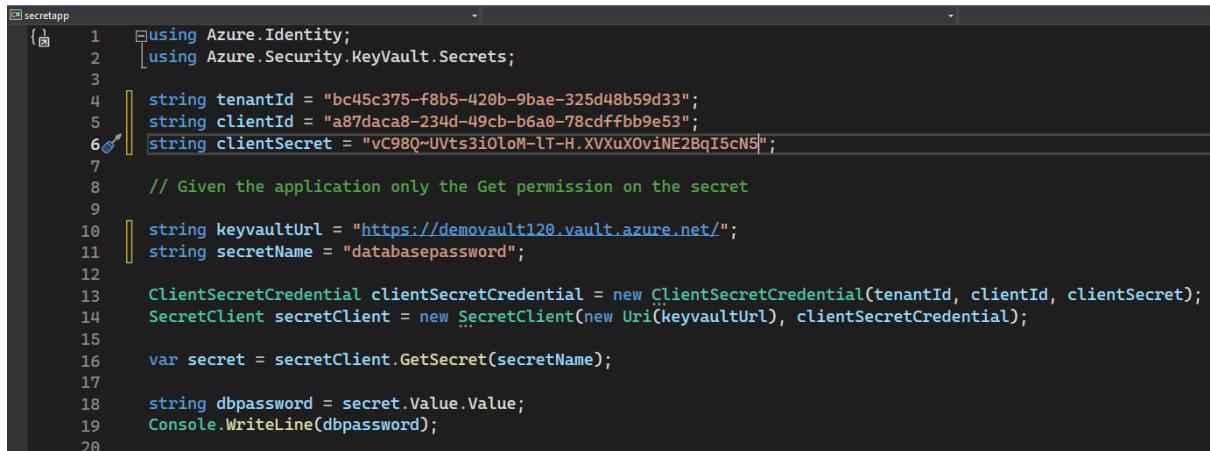
demoblobapp | Certificates & secrets

	Search		Got feedback?											
<hr/>														
Overview Quickstart Integration assistant														
Manage <ul style="list-style-type: none"> Branding & properties Authentication Certificates & secrets (selected) Token configuration API permissions Expose an API App roles Owners Roles and administrators 														
<p>Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.</p> <p>Application registration certificates, secrets and federated credentials can be found in the tabs below.</p> <table border="0"> <tr> <td>Certificates (0)</td> <td>Client secrets (1) (selected)</td> <td>Federated credentials (0)</td> </tr> </table> <p>A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.</p> <p>New client secret</p> <table border="0"> <thead> <tr> <th>Description</th> <th>Expires</th> <th>Value</th> <th>Secret ID</th> </tr> </thead> <tbody> <tr> <td>newsecret</td> <td>3/12/2024</td> <td>IE*****</td> <td>a8f165a4-c9ff-42f6-89cc-65367405e2cd</td> </tr> </tbody> </table>				Certificates (0)	Client secrets (1) (selected)	Federated credentials (0)	Description	Expires	Value	Secret ID	newsecret	3/12/2024	IE*****	a8f165a4-c9ff-42f6-89cc-65367405e2cd
Certificates (0)	Client secrets (1) (selected)	Federated credentials (0)												
Description	Expires	Value	Secret ID											
newsecret	3/12/2024	IE*****	a8f165a4-c9ff-42f6-89cc-65367405e2cd											

12. Below you can see your secret copy its value and paste it into the application.

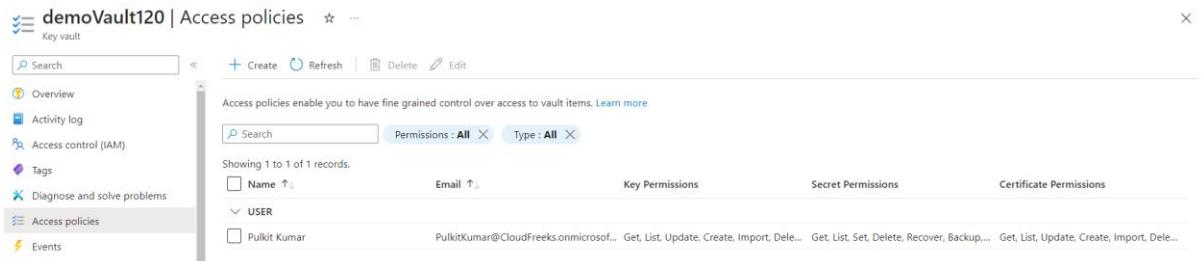
Certificates (0)	Client secrets (2)	Federated credentials (0)
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.		
+ New client secret		
Description	Expires	Value ⓘ
newsecret	3/12/2024	lKE*****
secret	3/12/2024	vC98Q~UVts3iOloM-IT-H.XVXuXOviNE2... Copy Delete

13. Once you're done now you need to go to Key vault



```
secretapp.cs
1  using Azure.Identity;
2  using Azure.Security.KeyVault.Secrets;
3
4  string tenantId = "bc45c375-f8b5-420b-9bae-325d48b59d33";
5  string clientId = "a87daca8-234d-49cb-b6a0-78cdffbb9e53";
6  string clientSecret = "vC98Q~UVts3iOloM-IT-H.XVXuXOviNE2BqI5cN5";
7
8  // Given the application only the Get permission on the secret
9
10 string keyvaultUrl = "https://demovault120.vault.azure.net/";
11 string secretName = "databasepassword";
12
13 ClientSecretCredential clientSecretCredential = new ClientSecretCredential(tenantId, clientId, clientSecret);
14 SecretClient secretClient = new SecretClient(new Uri(keyvaultUrl), clientSecretCredential);
15
16 var secret = secretClient.GetSecret(secretName);
17
18 string dbpassword = secret.Value.Value;
19 Console.WriteLine(dbpassword);
20
```

14. Now in your key vault navigate to access policies and choose to create a new one.



The screenshot shows the 'Access policies' section of the Azure Key Vault interface. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Access policies (which is currently selected). The main area displays a table of access policies:

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions
Pulkit Kumar	PulkitKumar@CloudFreaks.onmicrosoft.com	Get, List, Update, Create, Import, Delete...	Get, List, Set, Delete, Recover, Backup...	Get, List, Update, Create, Import, Delete...

15. Here we just need to give the Get permission and move to next.

Create an access policy

demoVault120

1 Permissions **2** Principal **3** Application (optional) **4** Review + create

Configure from a template

Select a template

Key permissions

Key Management Operations

- Select all
- Get
- List
- Update
- Create
- Import
- Delete
- Recover
- Backup
- Restore

Secret permissions

Secret Management Operations

- Select all
- Get
- List
- Set
- Delete
- Recover
- Backup
- Restore

Certificate permissions

- Certificate Management Operations

Select all
- Get
 - List
 - Update
 - Create
 - Import
 - Delete
 - Recover
 - Backup
 - Restore

Previous

Next

16. Then we need to choose our app registration object. After that just move to review page and create your access policy.

1 Permissions **2** Principal **3** Application (optional) **4** Review + create

Only 1 principal can be assigned per access policy.

Use the new embedded experience to select a principal. The previous popup experience can be accessed here. [Select a principal](#)

demob



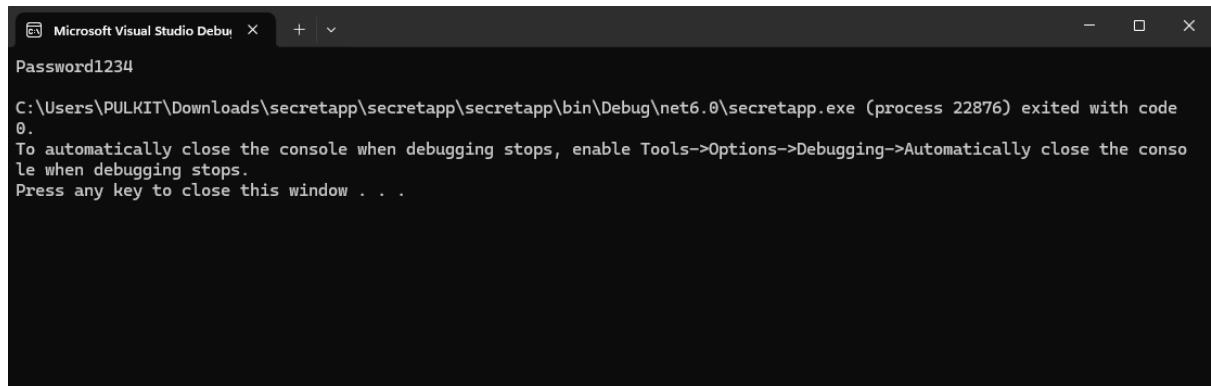
demoblobapp
a87daca8-234d-49cb-b6a0-78cdffbb9e53

Selected item



demoblobapp
a87daca8-234d-49cb-b6a0-78cdffbb9e53

17. Now if you go to Visual Studio and run your program locally then you will see your secret accordingly.



A screenshot of a Microsoft Visual Studio Debug console window. The title bar reads "Microsoft Visual Studio Debug". The main area of the window contains the following text:

```
Password1234
C:\Users\PULKIT\Downloads\secretapp\secretapp\secretapp\bin\Debug\net6.0\secretapp.exe (process 22876) exited with code
0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```