😊 Azure Key Vault

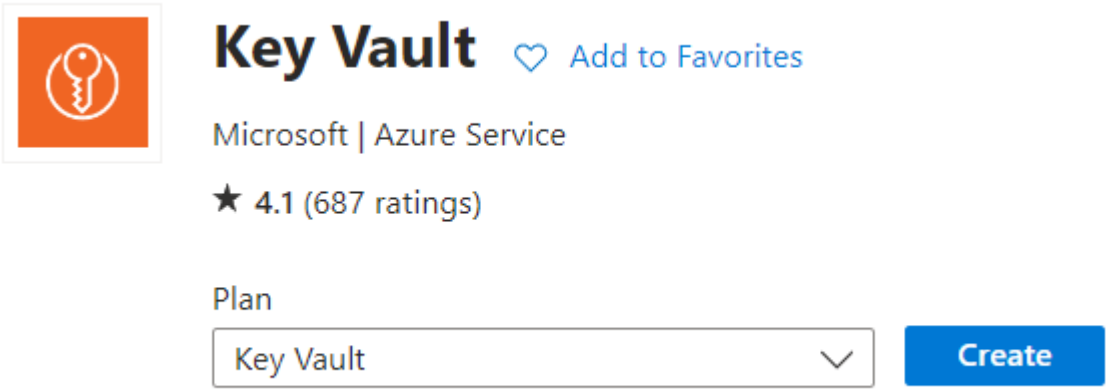
Azure Key Vault is a cloud service provided by Microsoft Azure that allows you to securely store and manage sensitive information such as keys, secrets, certificates, and connection strings. Here's a brief overview of Azure Key Vault:

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.

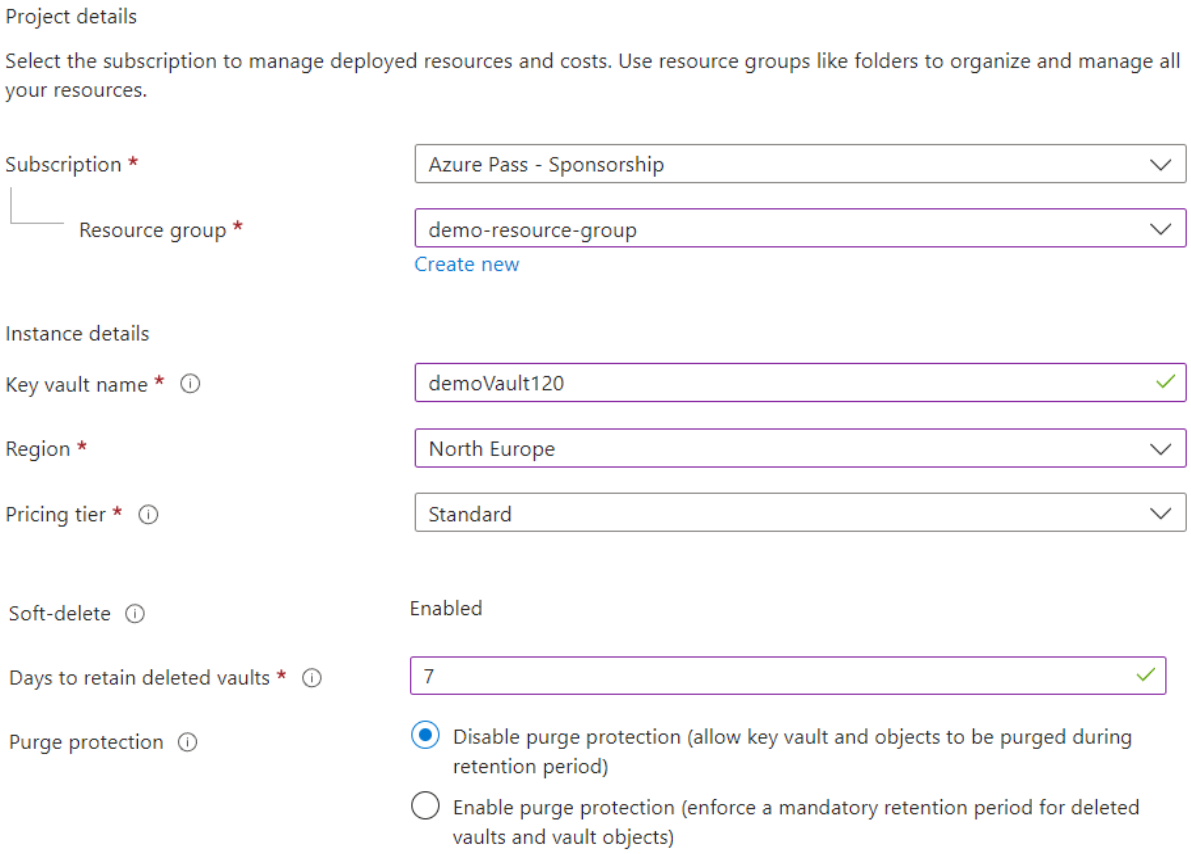
The end goal of this lab is to demonstrate how to securely access and retrieve secrets from Azure Key Vault using an application registered in Azure Active Directory (Azure AD). This setup ensures that sensitive information, such as database passwords, is securely managed and accessed, adhering to best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

😊 To begin with the Lab:

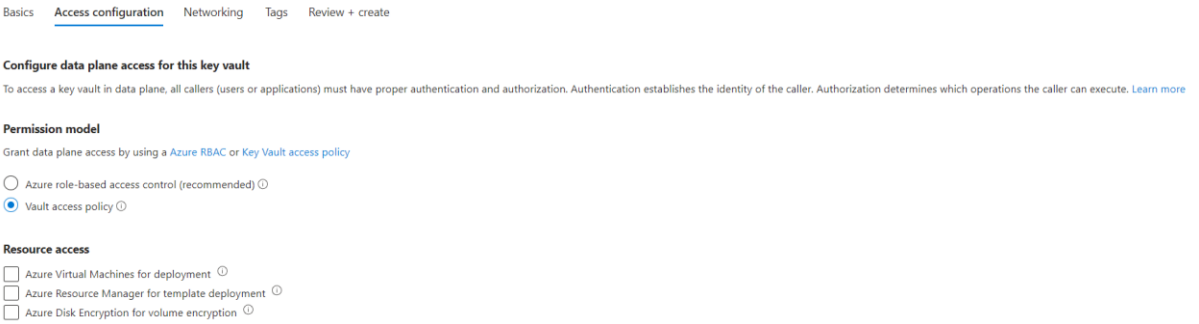
1. Now, in this lab we will go through a program that can be used to fetch a secret from the Azure Key Vault Service.
2. But first we need to create a Key Vault service. In the marketplace search for Key vault and choose this service accordingly.



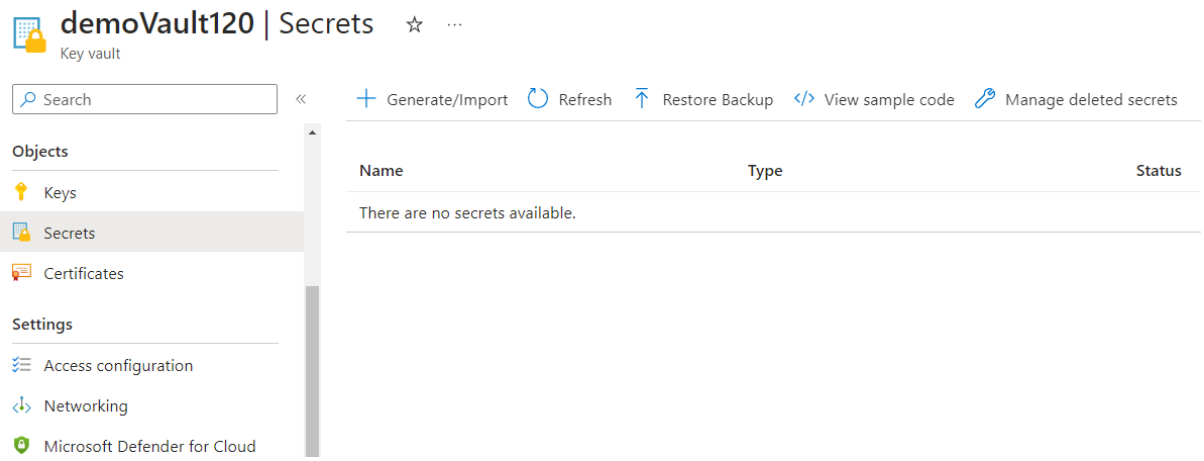
3. Now you need to choose your resource group and then give your vault a name after that choose your region.



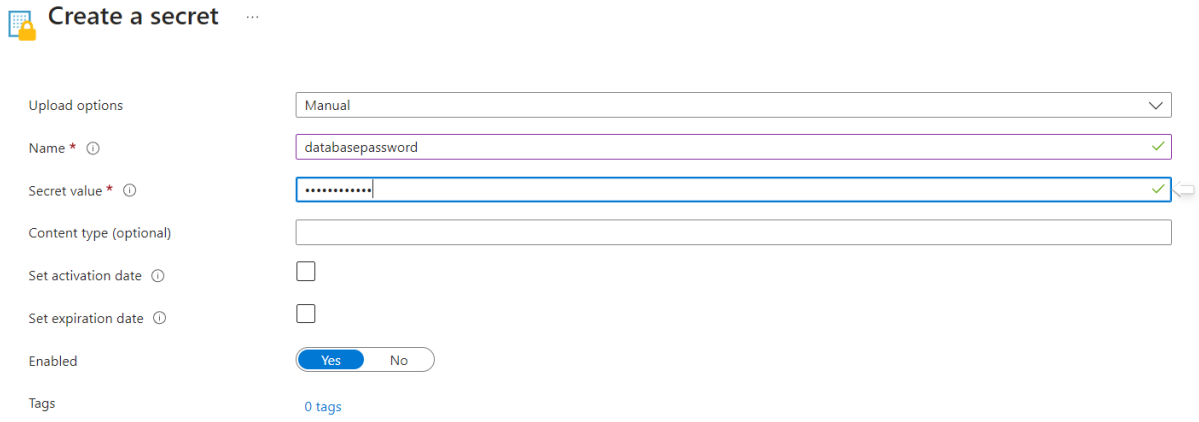
4. Then in the access configuration choose vault access policy and move to review page and create you vault.



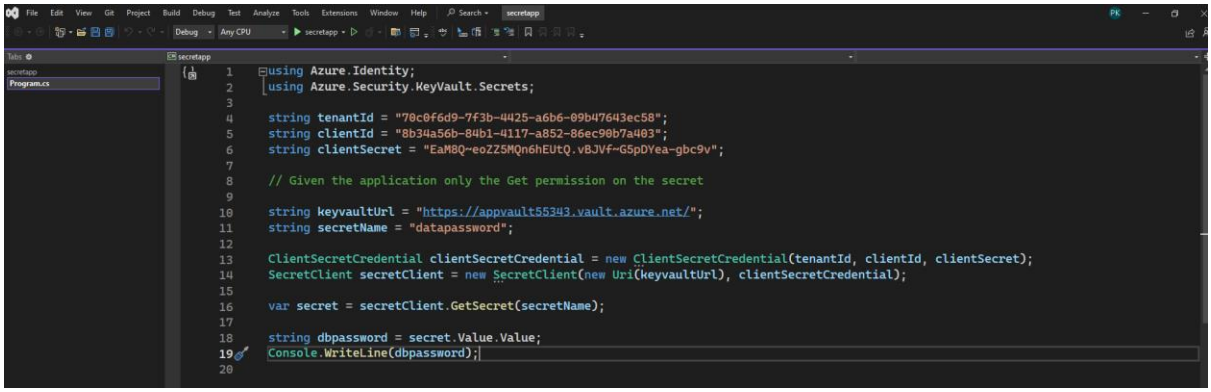
5. In your Vault go to secrets and generate a secret.



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.

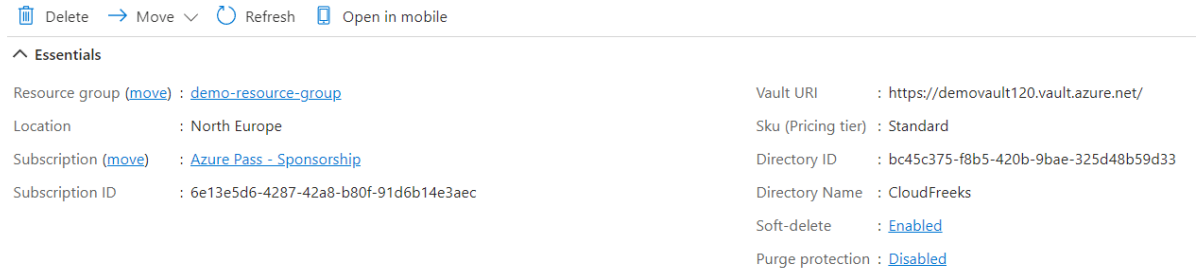


7. Now from GitHub you need to download a zip file named secret app and open it in Visual Studio 2022.
8. In this application we are going to make some changes, the first is to change the key vault URL and the second is to change the secret name.

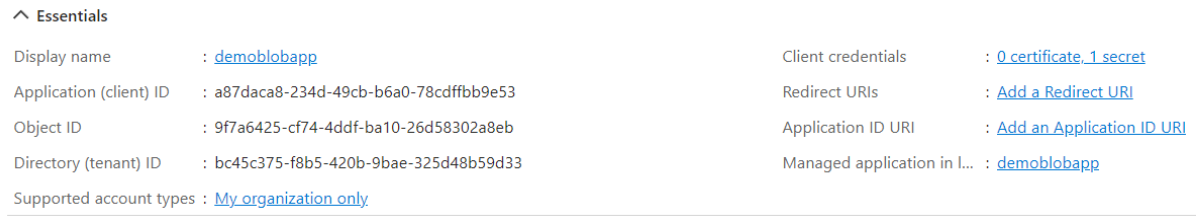


```
using Azure.Identity;
using Azure.Security.KeyVault.Secrets;

string tenantId = "70c0f6d9-7f3b-4425-a6b6-09b47643ec58";
string clientId = "8b34a56b-84b1-4117-a852-86ec90b7a403";
string clientSecret = "EaM8Q~eoZZ5MQn6hEUtQ.vBJVf~G5pDYea~gbc9v";

// Given the application only the Get permission on the secret

string keyvaultUrl = "https://appvault55343.vault.azure.net/";
string secretName = "datapassword";

ClientSecretCredential clientSecretCredential = new ClientSecretCredential(tenantId, clientId, clientSecret);
SecretClient secretClient = new SecretClient(new Uri(keyvaultUrl), clientSecretCredential);

var secret = secretClient.GetSecret(secretName);

string dbpassword = secret.Value.Value;
Console.WriteLine(dbpassword);
```

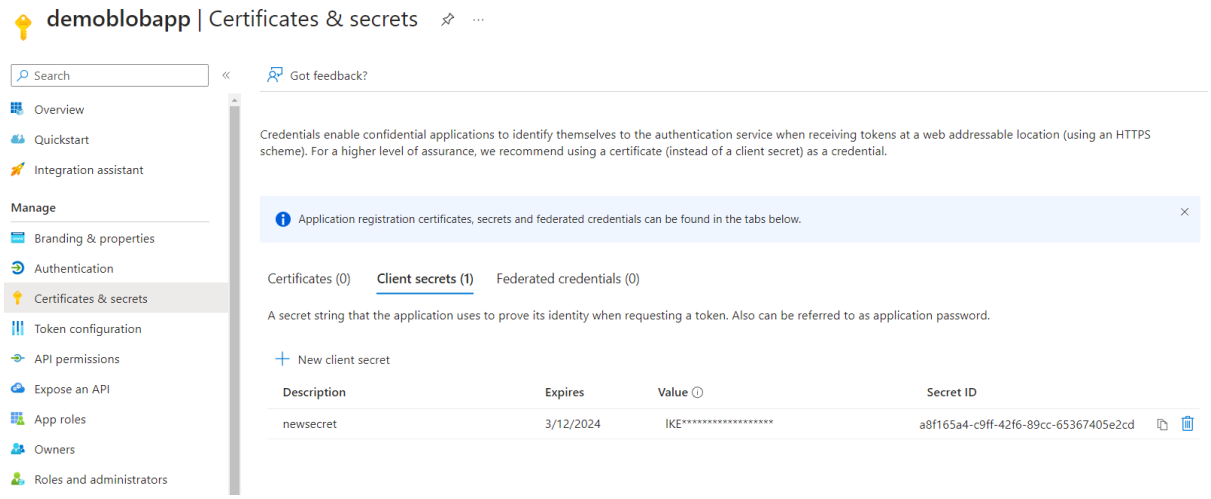
9. To get this you need to go to an overview of your key vault service you can see the Vault URL copy it and paste in the application.



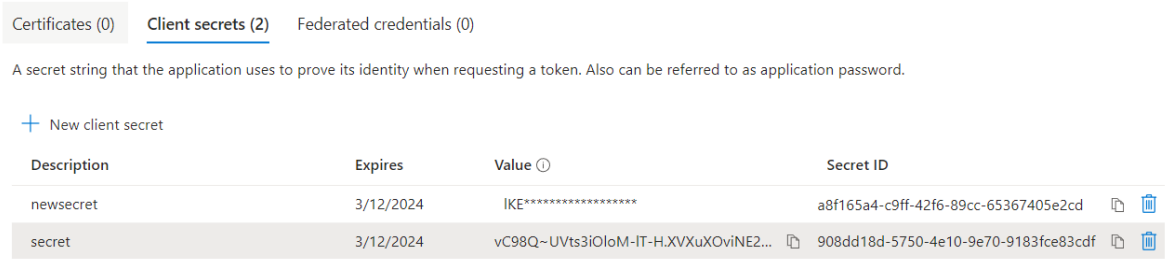
10. Now to get the tenant ID and client ID we can make use of our previous app registration in Microsoft Entra ID. Below you can see the client ID and the tenant ID just copy and paste them into your application.



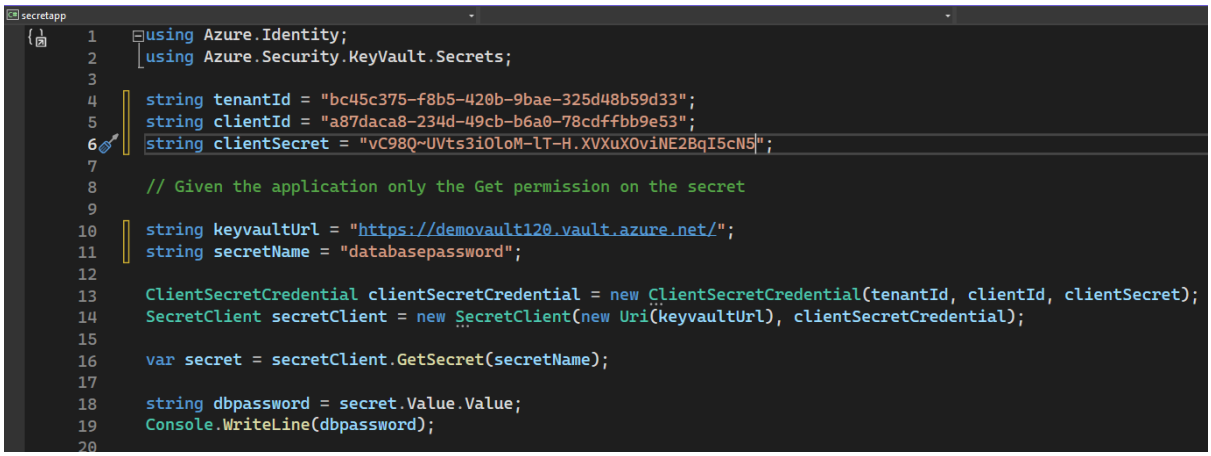
11. But for the client certificate we need to create a new one, for that in you app registration go to certificate and secrets and choose to create a new client secret.



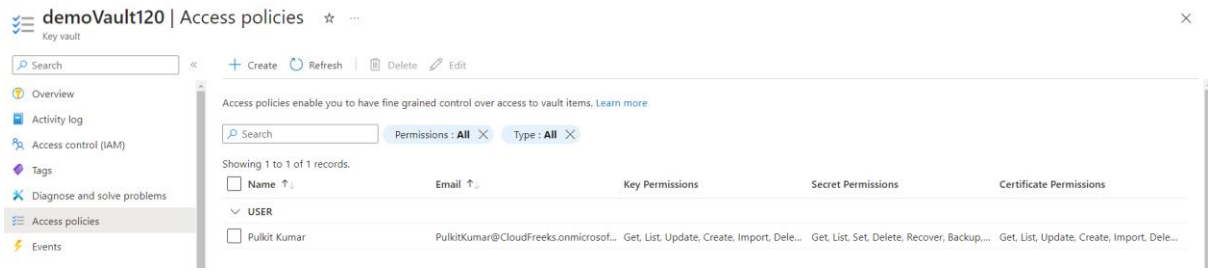
12. Below you can see your secret copy its value and paste it into the application.



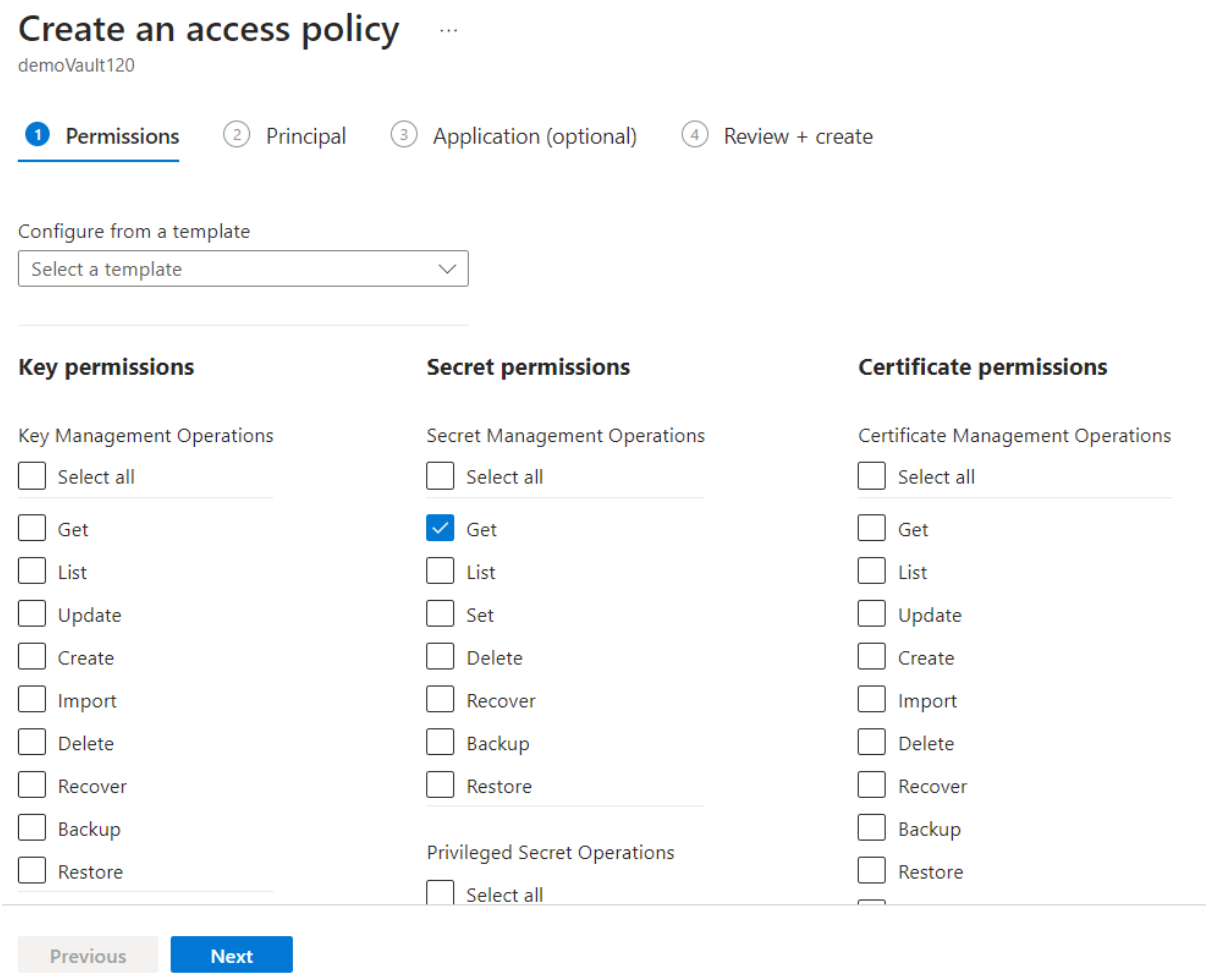
13. Once you're done now you need to go to Key vault



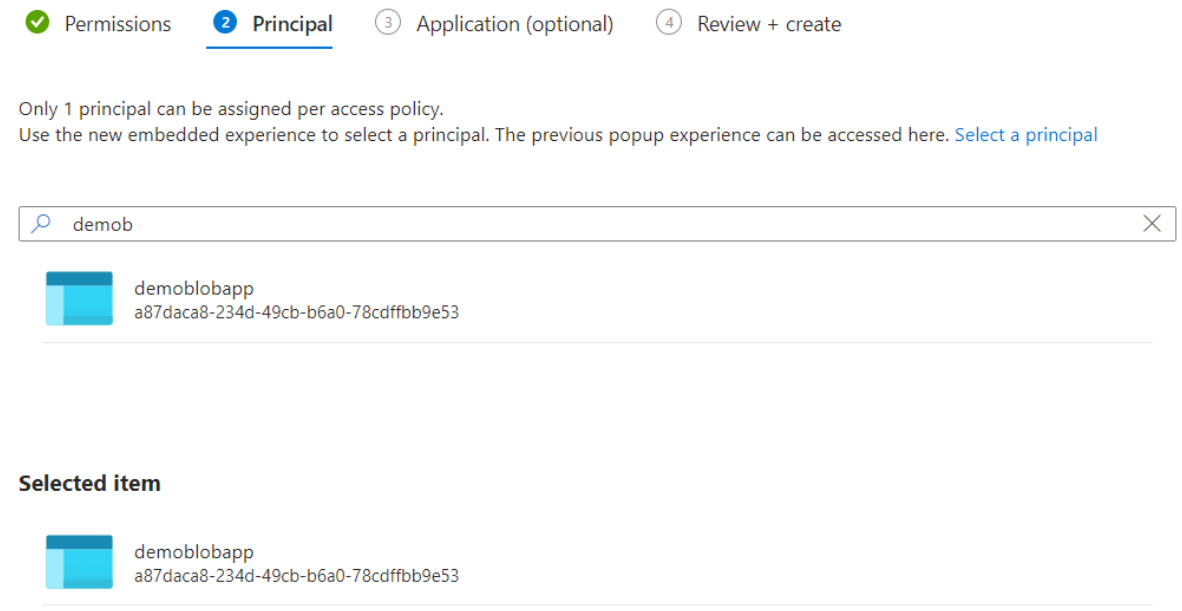
14. Now in your key vault navigate to access policies and choose to create a new one.



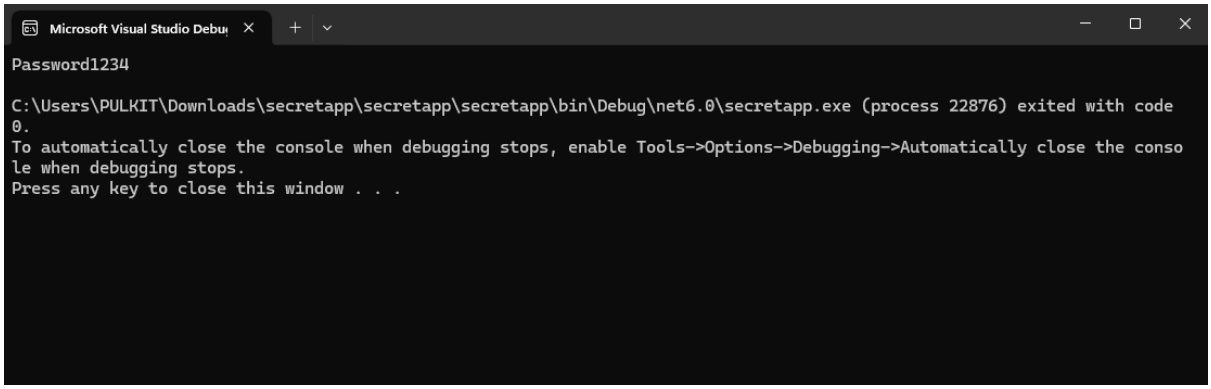
15. Here we just need to give the Get permission and move to next.



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



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


Microsoft Visual Studio Debu
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 console when debugging stops.
Press any key to close this window . . .