About Azure Key Vault

Azure Key Vault is one of several key management solutions in Azure, and helps solve the following problems:

- **Secrets Management** - Azure Key Vault can be used to Securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets
- **Key Management** - Azure Key Vault can be used as a Key Management solution. Azure Key Vault makes it easy to create and control the encryption keys used to encrypt your data.
- **Certificate Management** - Azure Key Vault lets you easily provision, manage, and deploy public and private Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificates for use with Azure and your internal connected resources.

Azure Key Vault has two service tiers: Standard, which encrypts with a software key, and a Premium tier, which includes hardware security module(HSM)-protected keys.

Why use Azure Key Vault?

Centralize application secrets

Centralizing storage of application secrets in Azure Key Vault allows you to control their distribution. Key Vault greatly reduces the chances that secrets may be accidentally leaked.

When application developers use Key Vault, they no longer need to store security information in their application. Not having to store security information in applications eliminates the need to make this information part of the code. For example, an application may need to connect to a database. Instead of storing the connection string in the app's code, you can store it securely in Key Vault.

Your applications can securely access the information they need by using URIs. These URIs allow the applications to retrieve specific versions of a secret. There's no need to write custom code to protect any of the secret information stored in Key Vault.

Securely store secrets and keys

Access to a key vault requires proper authentication and authorization before a caller (user or application) can get access. Authentication establishes the identity of the caller, while authorization determines the operations that they're allowed to perform.

Authentication is done via Microsoft Entra ID. Authorization may be done via Azure role-based access control (Azure RBAC) or Key Vault access policy. Azure RBAC can be used for both management of the vaults and to access data stored in a vault, while key vault access policy can only be used when attempting to access data stored in a vault.

Azure key vaults are encrypted at rest with a key stored in hardware security modules (HSMs). Azure safeguards keys, secrets, and certificates using industry-standard algorithms, key lengths, and software cryptographic modules. For added assurance, you can generate or import keys in HSMs (type RSA-HSM, EC-HSM, or OCT-HSM) that never leave the HSM boundary. Azure Key Vault uses Federal Information Processing Standard 140 validated software cryptographic modules and HSMs.
Finally, Azure Key Vault is designed so that Microsoft doesn't see or extract your data.

Monitor access and use

Once you've created a couple of Key Vaults, you'll want to monitor how and when your keys and secrets are being accessed. You can monitor activity by enabling logging for your vaults. You can configure Azure Key Vault to:

- Archive to a storage account.
- Stream to an event hub.
- Send the logs to Azure Monitor logs.

You have control over your logs and you may secure them by restricting access and you may also delete logs that you no longer need.

Simplified administration of application secrets

When storing valuable data, you must take several steps. Security information must be secured, it must follow a life cycle, and it must be highly available. Azure Key Vault simplifies the process of meeting these requirements by:

- Removing the need for in-house knowledge of Hardware Security Modules.
- Scaling up on short notice to meet your organization's usage spikes.
- Replicating the contents of your Key Vault within a region and to a secondary region. Data replication ensures high availability and takes away the need of any action from the administrator to trigger the failover.
- Providing standard Azure administration options via the portal, Azure CLI and PowerShell.
- Automating certain tasks on certificates that you purchase from Public CAs, such as enrollment and renewal.

In addition, Azure Key Vaults allow you to segregate application secrets. Applications may access only the vault that they're allowed to access, and they can be limited to only perform specific operations. You can create an Azure Key Vault per application and restrict the secrets stored in a Key Vault to a specific application and team of developers.

Integrate with other Azure services

As a secure store in Azure, Key Vault has been used to simplify scenarios like:

- Azure Disk Encryption
- The always encrypted and Transparent Data Encryption functionality in SQL server and Azure SQL Database
- Azure App Service.

Key Vault itself can integrate with storage accounts, event hubs, and log analytics.

Set and retrieve a secret from Azure Key Vault using PowerShell

Connect-AzAccount

1. **Create a resource group**

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

2. **Create a key vault**

```
New-AzKeyVault -Name "<your-unique-keyvault-name>" -ResourceGroupName "myResourceGroup" -Location "EastUS"
```

3. **Give your user account permissions to manage secrets in Key Vault**

```
New-AzRoleAssignment -SignInName "<upn>" -RoleDefinitionName "Key Vault Secrets Officer" -Scope "/subscriptions/<subscription-id>/resourceGroups/<resource-group-name>/providers/Microsoft.KeyVault/vaults/<your-unique-keyvault-name>"
```

4. **Adding a secret to Key Vault**

```
$secretvalue = Read-Host -Prompt 'Enter the example password' -AsSecureString
```

5. **Retrieve a secret from Key Vault**

```
$secret = Set-AzKeyVaultSecret -VaultName "<your-unique-keyvault-name>" -Name "ExamplePassword" -SecretValue $secretvalue
```