What is Azure Key Vault?

5 minutes

Azure Key Vault is a *secret store*: a centralized cloud service for storing application secrets - configuration values like passwords and connection strings that must remain secure at all times. Key Vault helps you control your applications' secrets by keeping them in a single central location and providing secure access, permissions control, and access logging.

The main benefits of using Key Vault are:

- Separation of sensitive application information from other configuration and code, reducing risk of accidental leaks
- Restricted secret access with access policies tailored to the applications and individuals that need them
- Centralized secret storage, allowing required changes to happen in only one place
- Access logging and monitoring to help you understand how and when secrets are accessed

Secrets are stored in individual *vaults*, which are Azure resources used to group secrets together. Secret access and vault management is accomplished via a REST API, which is also supported by all of the Azure management tools as well as client libraries available for many popular languages. Every vault has a unique URL where its API is hosted.

> ⓘ **Important**
>
> **Key Vault is designed to store configuration secrets for server applications.** It's not intended for storing data belonging to your app's users, and it shouldn't be used in the client-side part of an app. This is reflected in its performance characteristics, API, and cost model.
>
> User data should be stored elsewhere, such as in an Azure SQL database with Transparent Data Encryption, or a storage account with Storage Service Encryption. Secrets used by your application to access those data stores can be kept in Key Vault.

What is a secret in Key Vault?

In Key Vault, a secret is a name-value pair of strings. Secret names must be 1-127 characters long, contain only alphanumeric characters and dashes, and must be unique within a vault. A

secret value can be any UTF-8 string up to 25 KB in size.

> **Tip**
>
> Secret names don't need to be considered especially secret themselves. You can store them in your app's configuration if your implementation calls for it. The same is true of vault names and URLs.

> **Note**
>
> Key Vault supports two additional kinds of secrets beyond strings — *keys* and *certificates* — and provides useful functionality specific to their use cases. This module does not cover these features and concentrates on secret strings like passwords and connection strings.

Vault authentication and permissions

Azure Key Vault's API uses Azure Active Directory to authenticate users and applications. Vault access policies are based on *actions*, and are applied across an entire vault. For example, an application with **Get** (read secret values), **List** (list names of all secrets), and **Set** (create or update secret values) permissions to a vault is able to create secrets, list all secret names, and get and set all secret values in that vault.

All actions performed on a vault require authentication and authorization — there is no way to grant any kind of anonymous access.

> **Tip**
>
> When granting vault access to developers and apps, grant only the minimum set of permissions needed. Permissions restrictions help avoid accidents caused by code bugs and reduce the impact of stolen credentials or malicious code injected into your app.

Developers will usually only need **Get** and **List** permissions to a development-environment vault. Some engineers will need full permissions to change and add secrets when necessary.

For apps, often only **Get** permissions are required. Some apps may require **List** depending on the way the app is implemented. The app we'll implement in this module's exercise requires the **List** permission because of the technique it uses to read secrets from the vault.

Next unit: Exercise - Create a key vault and store secrets

Continue >