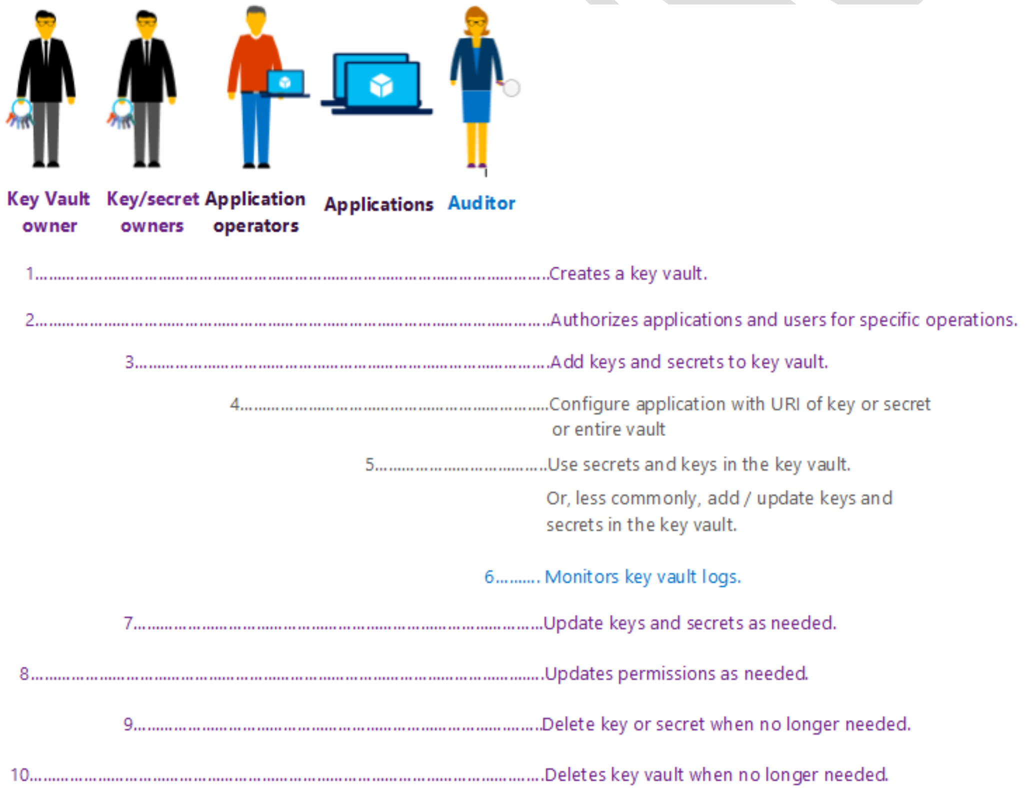
Agenda: Key Vault

- Introduction to Key Vault
- Secrets vs Keys
- How it Works
- Key Vault in CI and CD Pipeline
- Replace Token Task

About Key Vault

- Key Vault serves as a store of cryptographic keys and secrets, such as authentication keys, storage account keys, data encryption keys, .PFX files, and passwords.
- Developers can create keys for development and testing in minutes, and then seamlessly migrate them to production keys.
- Security administrators can grant (and revoke) permission to keys, as needed.



1……Creates a key vault.

2……Authorizes applications and users for specific operations.

3……Add keys and secrets to key vault.

4……Configure application with URI of key or secret or entire vault

5……Use secrets and keys in the key vault.
Or, less commonly, add / update keys and secrets in the key vault.

6……Monitors key vault logs.

7……Update keys and secrets as needed.

8……Updates permissions as needed.

9……Delete key or secret when no longer needed.

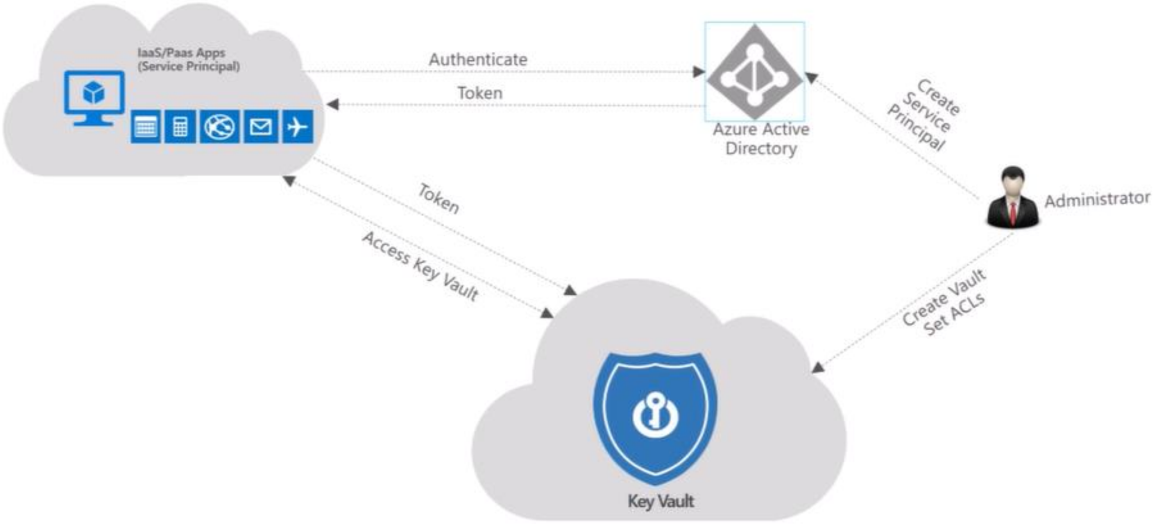
10……Deletes key vault when no longer needed.

How to access keys and secrets:

- To access keys and secrets, users and applications must possess valid Azure Active Directory tokens representing security principal with sufficient permissions to the target vault.
- You can use a REST-based API or Azure PowerShell to retrieve secrets and public parts of keys (in JSON format) from Key Vault.

How it works:

1. Applications that need access to Key Vault are registered with Azure Active Directory as Service Principals.
2. Administrator will then create a Key Vault and sets Access Control Lists on the vault so that applications can access it.
3. Applications then authenticates with Azure Active Directory and gets the Token.
4. The application then presents this token to Azure Key Vault
5. Azure Key vault then grants access based on Access Control List (ACL)



Walkthrough: Steps to create Hello Key Vault Application

1. Create Service Principal (Azure AD Application)
2. Create Key Vaults and Set ACL's
3. Add Keys and Secrets to the Key Vault.
4. Get Sample Code and Edit config configuration.
5. Build and Run the application.

Step 1: Create Azure Active Directory Application (Service Principal)

1. Azure Portal → Azure Active Directory → App registrations → **+ New application registration**
2. Provide Name="**KeyVaultDemoApp**"
3. KeyVaultDemoApp → Properties → **Copy App ID URI**
4. KeyVaultDemoApp → Certificates and Secrets → Click on **+ New client secret** and copy the same.

Step 2: Create Key Vaults

5. Azure Portal → All Services → Key Vault → +Add
6. Enter Name = DssDemoKeyVault
7. Add **Access Policy** → +Add
   a. Select Secret Persmissions (Get / List)
   b. Select Principals = "***KeyVaultDemoApp***" (AAD Application Created before)
   c. Authorized Application is disabled
8. Create
9. Go to Created Key Vault → Essentials → Copy DNS Name: https://dssdemokeyvault.vault.azure.net/

Step 3: Add a key or secret to the key vault

10. Select the DssDemoKeyVault → Secrets → +Add
11. Options = Generate, Name=MyFirstSec → Create
12. Click on Key → Current Version → Copy Identifier

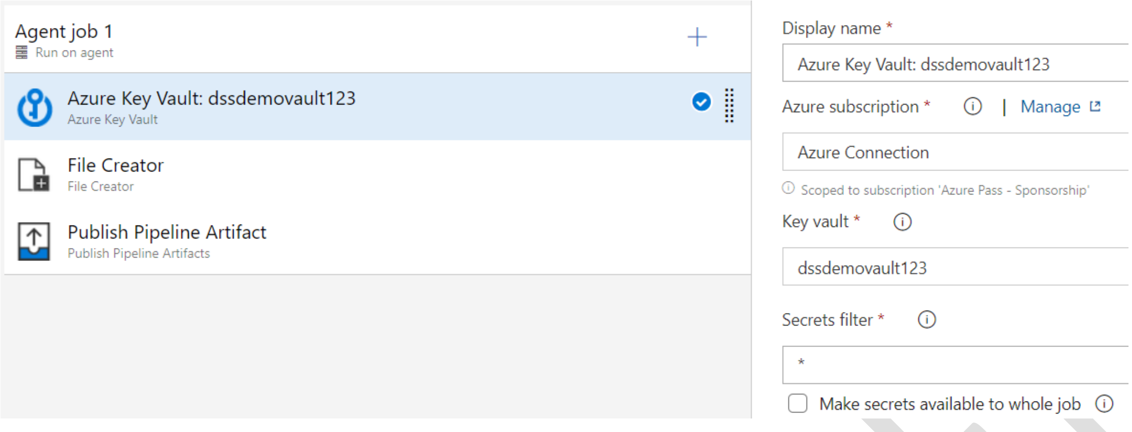
You can reference a key that you created or uploaded to Azure Key Vault by using its URI.

13. Get the current version, you can use https://dssdemokeyvalut.vault.azure.net/secrets/MyFirstSecret and use https://dssdemokeyvalut.vault.azure.net/secrets/MyFirstSecret/cgacf4f763ar42ffb0a1gca546aygd87 to get this specific version.

Key Vault in Azure DevOps CI and CD Pipeline

Azure Key Vault Task:

Azure Pipelines will fetch the latest values of the secrets and set them as task variables which can be consumed in the following tasks.



Agent job 1
Run on agent

Azure Key Vault: dssdemovault123
Azure Key Vault

File Creator
File Creator

Publish Pipeline Artifact
Publish Pipeline Artifacts

Display name *
Azure Key Vault: dssdemovault123

Azure subscription * | Manage
Azure Connection
Scoped to subscription 'Azure Pass - Sponsorship'

Key vault *
dssdemovault123

Secrets filter *
*

□ Make secrets available to whole job

```
trigger:
- master

pool:
  vmImage: ubuntu-latest

steps:
- task: AzureKeyVault@1
  displayName: 'Azure Key Vault: dssdemovault123'
  inputs:
    azureSubscription: 'Azure Subscription MPN'
    KeyVaultName: 'dss-keyvault-demo'
    SecretsFilter: '*'
    RunAsPreJob: false

- task: file-creator@6
  inputs:
    filepath: '$(build.artifactstagingdirectory)/demo.txt'
    filecontent: |
      $(Server)
      $(Password)

- task: PublishPipelineArtifact@1
  displayName: 'Publish Pipeline Artifact'
```

```
 inputs:
   targetPath: '$(build.artifactstagingdirectory)'
```

Replace Token Task

Note: Replace Token Task can be used to replace Tokens in any file (mostly configuration file) with values of Variables in the pipeline. The token in the pipelines must be enclosed in between some prefix and suffic. Example below replaces all values in appsettings.json with variables of pipeline provided they are enclosed inbetween [* and *]

```
- task: qetza.replacetokens.replacetokens-task.replacetokens@3
  displayName: 'Replace tokens in $(build.sourcedirectory)/**/appsettings.json'
  inputs:
    targetFiles: '$(build.sourcedirectory)/**/appsettings.json'
    tokenPrefix: '[*'
    tokenSuffix: '*]'
```

Example of AppSettings.json

```
{
  "AllowedHosts": "*",
  "ConnectionString": "server=[*SQLServer*];Database=[*Database*];uid=[*Login*];Password=[*Password*]"
}
```