Protecting Application Keys and Secrets with Azure Key Vault and MSI

Reza Salehi

MCT, MCSA(CLOUD PLATFORM), MCPD

@zaalion linkedin.com/in/rezasalehi2008

Overview

What are we trying to protect? keys vs. secrets

Understanding Azure Key Vault and how it works

Demo: Azure Key Vault

Securing Azure SQL connection string using Managed Service Identity (MSI)

Demo: Managed Service Identity (MSI)

Microsoft tools to help you manage identity

Module summary

Microsoft Azure Key Vault

What Are We Trying to Protect?

Key

Cryptographic keys used in other Microsoft Azure services such as "Always Encrypted" or "data encryption at rest"

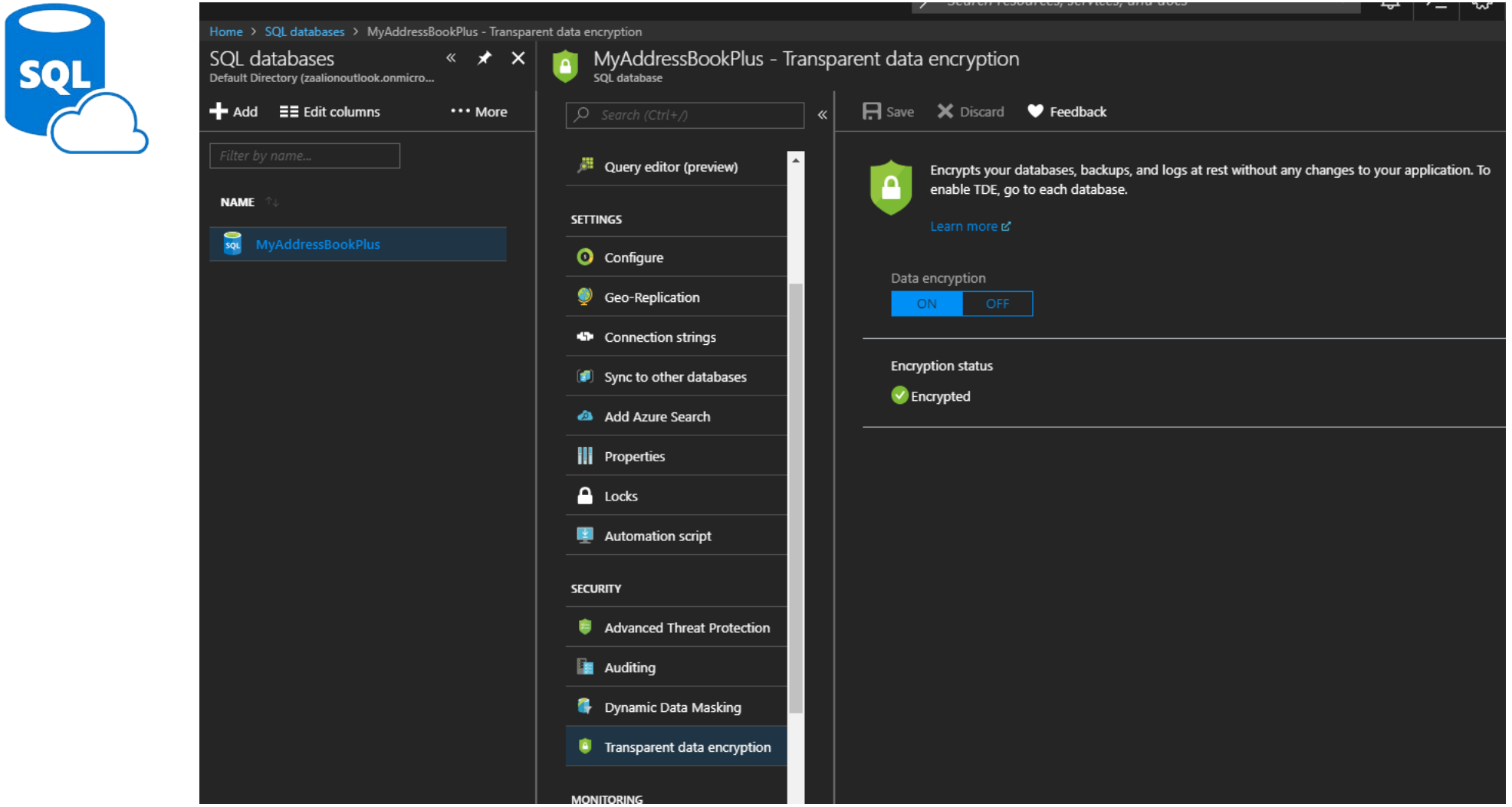
Secret

Any sensitive information including SQL server, Redis, storage connection strings or other information your application might need at runtime

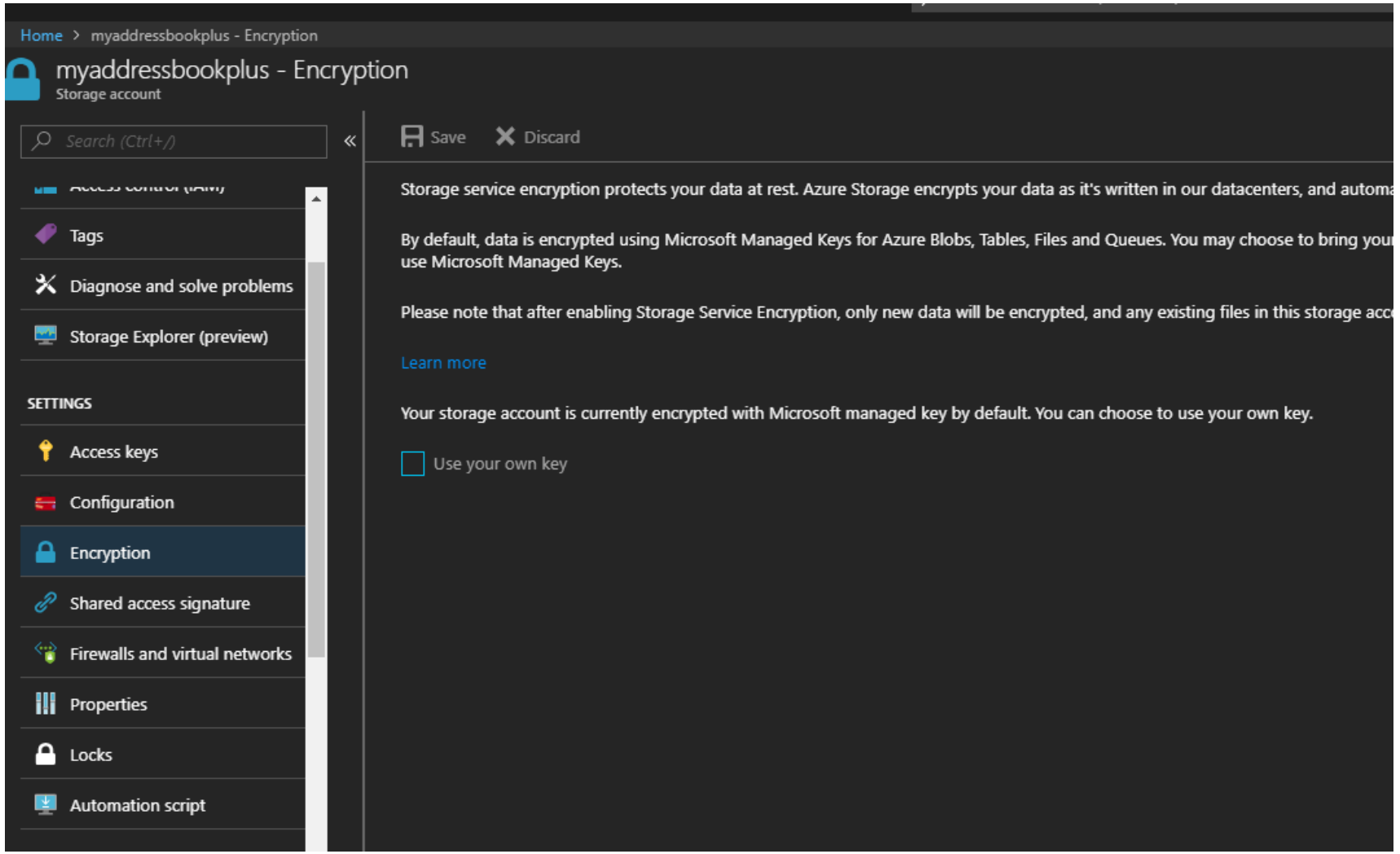
Certificate

x509 certificates being used in HTTPS/SSL communications

Keys


Home > SQL databases > MyAddressBookPlus - Transparent data encryption
SQL databases
Default Directory (zaalionoutlook.onmicro...
Add
Edit columns
More
Filter by name...
NAME
MyAddressBookPlus
MyAddressBookPlus - Transparent data encryption
SQL database
Search (Ctrl+/)
Query editor (preview)
SETTINGS
Configure
Geo-Replication
Connection strings
Sync to other databases
Add Azure Search
Properties
Locks
Automation script
SECURITY
Advanced Threat Protection
Auditing
Dynamic Data Masking
Transparent data encryption
MONITORING
Save
Discard
Feedback
Encrypts your databases, backups, and logs at rest without any changes to your application. To enable TDE, go to each database.
Learn more
Data encryption
ON
OFF
Encryption status
Encrypted

Keys


Home > myaddressbookplus - Encryption
myaddressbookplus - Encryption
Storage account
Search (Ctrl+/)
Save
Discard
Tags
Diagnose and solve problems
Storage Explorer (preview)
SETTINGS
Access keys
Configuration
Encryption
Shared access signature
Firewalls and virtual networks
Properties
Locks
Automation script
Storage service encryption protects your data at rest. Azure Storage encrypts your data as it's written in our datacenters, and automa
By default, data is encrypted using Microsoft Managed Keys for Azure Blobs, Tables, Files and Queues. You may choose to bring you
use Microsoft Managed Keys.
Please note that after enabling Storage Service Encryption, only new data will be encrypted, and any existing files in this storage acc
Learn more
Your storage account is currently encrypted with Microsoft managed key by default. You can choose to use your own key.
Use your own key

```
<add key="CacheConnection"
value="myaddressbookplus.redis.cache.windows.net:6380,password=hQwiwqd+jij2nZZHzyW5AtawOTq71P4DkNn3n5BFPrw=,ssl=True,abortConnect=False"/>
```

Secrets

Azure Redis cache connection string

```
<add key=“StorageConnectionString“
value="DefaultEndpointsProtocol=https;AccountName=myaddress
bookplus;AccountKey=BgAVowM+oErfnie9myvJ5XiBU0RAXtYlmyqMwEZ
ptz+pUaK2ERqZI1PJW1WL5vHofijj2SIYJq0eF7DE17OPVg==;EndpointS
uffix=core.windows.net"/>
```

Secrets

Azure Storage connection string

```
-----BEGIN CERTIFICATE-----
MIIC+TCCAeGgAwIBAgIJAMZAdG2sFLm0MA0GCSqGSIb3DQEBBQUAMBMxETAPBgNVBAMMCHRlc3QuY29tMB4XDTE4MDgxNTE5NTkyN
1oXDTI4MDgxMjE5NTkyN1owEzERMA8GA1UEAwwIdGVzdC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDGuoqtgI
qVA68A+SzkAC5cidGPbc1Lb/vsyScTDUP6cN5G4KYdPqPEDSv65rHbjl+DOCAsEd/sQpKseT1F8lhnbatBoGnyHPKwUSiiWt4gESM
PCOZSzaMz257Xqw9XjspVKYNExiJv40/iDrrFaGveYEaCLsM6iPQ9fhXOhwD+Tf08pSRq53C37jPcu/+ouvuSc3m6s7aJqcffZjeS
r3eiSpEeueu6GLLTBmRYIp2aQ6qnW9YzT+UpRck0EupTkLij1qFcsmxchrdq2zuj1p5pIgTWi60q3zsZEWdA/2MC0aOeiP+/lPgeW
DlpYf4PEO7dXSLC8ym+XfD1gd7xNyHFAgMBAAGjUDBOMB0GA1UdDgQWBBSgW4gFn9Log0HkN/A2HfyRzyJsHTAfBgNVHSMEGDAWgB
SgW4gFn9Log0HkN/A2HfyRzyJsHTAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQC/tG4TBh7DBPy4qdG4S5RpRCxa89C
OCVF+x0QZWyTGa/1fgglgvmRY1ENZKWOcLCiZ8Gb2F8yzh+tRUzP7b/8AmqK1Hv+Ap9lYWTs8PJT2vg4IZTOiwGcEWntQF15BBP/C
BMeBIcgDOb+TnJLHKCnJGxLK+EhkbCwVPPnlkZHLjmvXrIwvQxNxQm8I7wqKX/TNmg3ekzr9ZCkNafY6SkqjlKit/i15aKj5j9zd7
ynJ2wQmGbKs2BvO95lQ8UGXzFOOCQ3J/ibWs2qjUvv/QdhbY9eZntilwoCFVuBWPx5lBlm5WEHoTqBoUDD4ayUos8LU0zFx5KM2lz
ilr5z+mGVG-----END CERTIFICATE-----
```

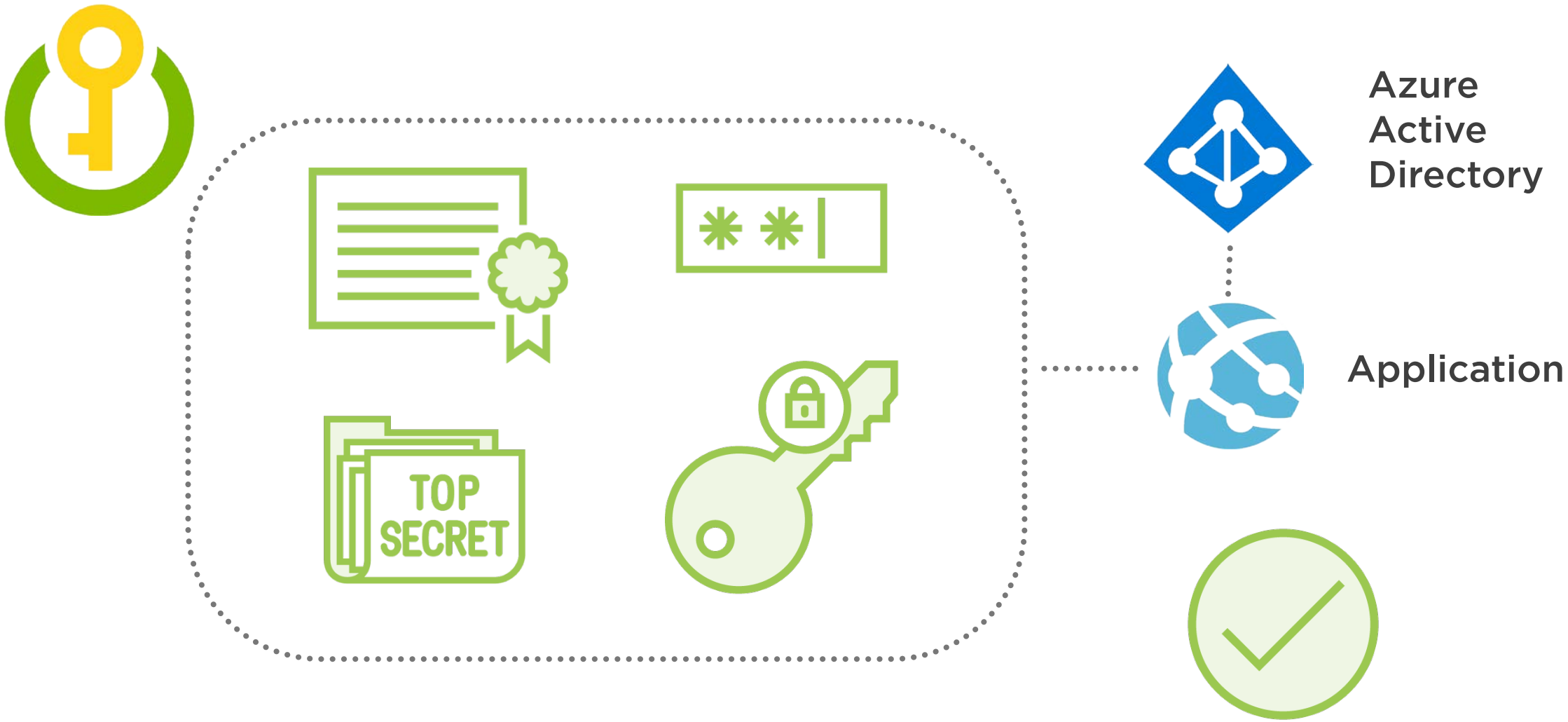
Certificates

x509 certificates being used in HTTPS/SSL communications

What is Azure Key Vault?

What is Azure Key Vault?

What is Azure Key Vault?



Why Microsoft Azure Key Vault?

Centralized key management

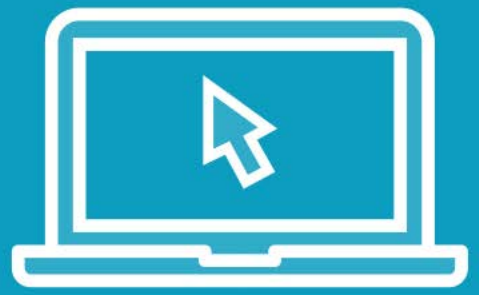
- Grant/Revoke access to people and application
- Auditing/logging
- Key rotation and versioning
- Safer than having the secrets in the code, source control or VMs

Hardware Security Modules (HSM)

Support for PowerShell, Azure CLI and RESTful API

Native support by other Microsoft Azure services

Demo



Create a new Vault in Azure Key Vault

Move the Redis cache connection string to the new Vault (as a new secret)

Register *MyAddressBook+* with Azure Active Directory

Configure *MyAdressBook+* code

- Remove Redis cache connection string from configuration
- Add support to load the connection string from Azure Key Vault

Confirm that *MyAddressBook+* can use the cache

```
<!- web.config -->
<add key="ClientId" value="686251ec-01b5-4877-9663-7bacf2d23bcc" />
<add key="ClientSecret“value="cQY9G2kEXr3/+8oqUYMT0IWYTXB9UWBJt8Ro0WMKJ48=" />
```

Code Changes

Active directory Client Id, Client Secret

```
<!– web.config -->

<add key="ClientId" value="686251ec-01b5-4877-9663-7bacf2d23bcc" />

<add key="ClientSecret“value="cQY9G2kEXr3/+8oqUYMT0IWYTXB9UWBJt8Ro0WMKJ48=" />

<add key="CacheConnectionSecretUri"
value="https://myaddressbookplusvault.vault.azure.net:443/secrets/CacheConnection
/8233eacf6f97446a89aea139172dc616" />
```

Code Changes

Azure Active directory (AAD) Client Id, Client Secret

Redis cache connection string secret Key Vault URL

```
<!- Global.asax.cs -->

var kv = new KeyVaultClient(new
KeyVaultClient.AuthenticationCallback(KeyVaultService.GetToken));

var sec =
kv.GetSecretAsync(WebConfigurationManager.AppSettings["CacheConnectionSecretUri"])
.Result;

KeyVaultService.CacheConnection = sec.Value;
```

Code Changes

Read the secret from Azure Key Vault and save in memory.

Enable Key Vault Soft-delete

Soft-delete Allows recovery of deleted vaults and vault objects including keys, secrets, and certificates

```
# Existing key vault

($resource = Get-AzureRmResource -ResourceId (Get-AzureRmKeyVault
-VaultName “MyAddressBookVault").ResourceId).Properties | Add-
Member -MemberType "NoteProperty" -Name "enableSoftDelete“ -Value
"true"

Set-AzureRmResource -ResourceId $resource.ResourceId -Properties
$resource.Properties
```

Enable Key Vault Soft-delete

Soft-delete Allows recovery of deleted vaults and vault objects including keys, secrets, and certificates

```
# New key vault

New-AzureRmKeyVault -VaultName "MyAddressBookVault" -ResourceGroupName
“MyRG" -Location "westus" -EnableSoftDelete
```

Use Key Vault Soft-delete

Soft-delete Allows recovery of deleted vaults and vault objects including keys, secrets, and certificates

Enable Key Vault ”Do Not Purge”

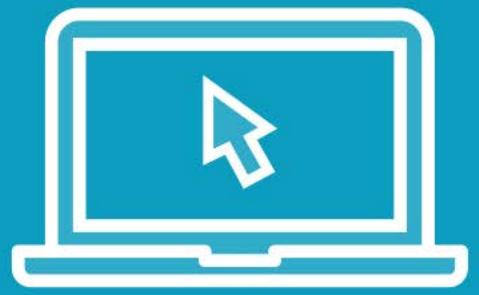
“Do Not Purge” prevents accidental purging of deleted vaults and vault objects including keys, secrets, and certificates

```
# Existing key vault

($resource = Get-AzureRmResource -ResourceId (Get-AzureRmKeyVault
-VaultName “MyAddressBookVault").ResourceId).Properties | Add-
Member -MemberType NoteProperty -Name enablePurgeProtection  -
Value "true"

Set-AzureRmResource -ResourceId $resource.ResourceId -Properties
$resource.Properties
```

Enable Key Vault ”Do Not Purge”

“Do Not Purge” prevents accidental purging of deleted vaults and vault objects including keys, secrets, and certificates

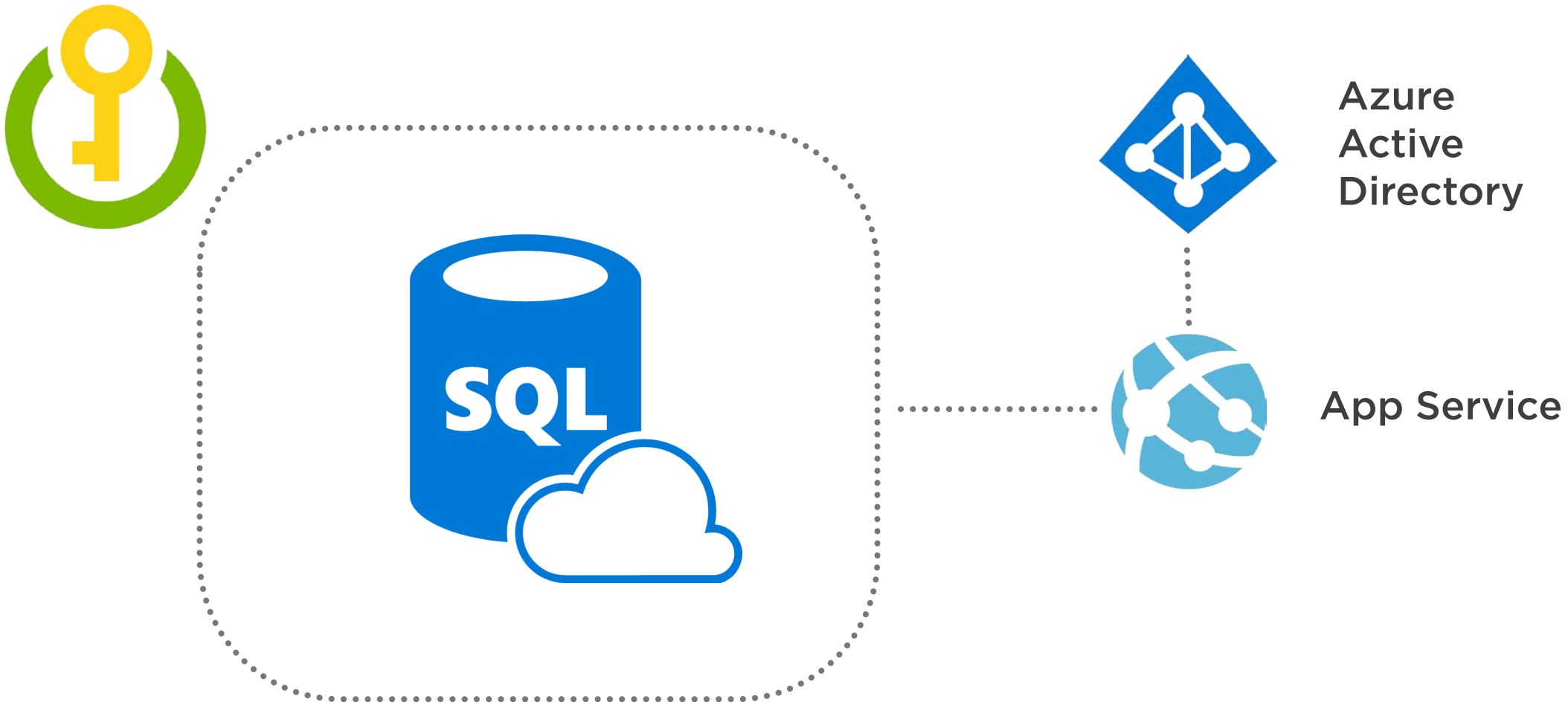
Demo



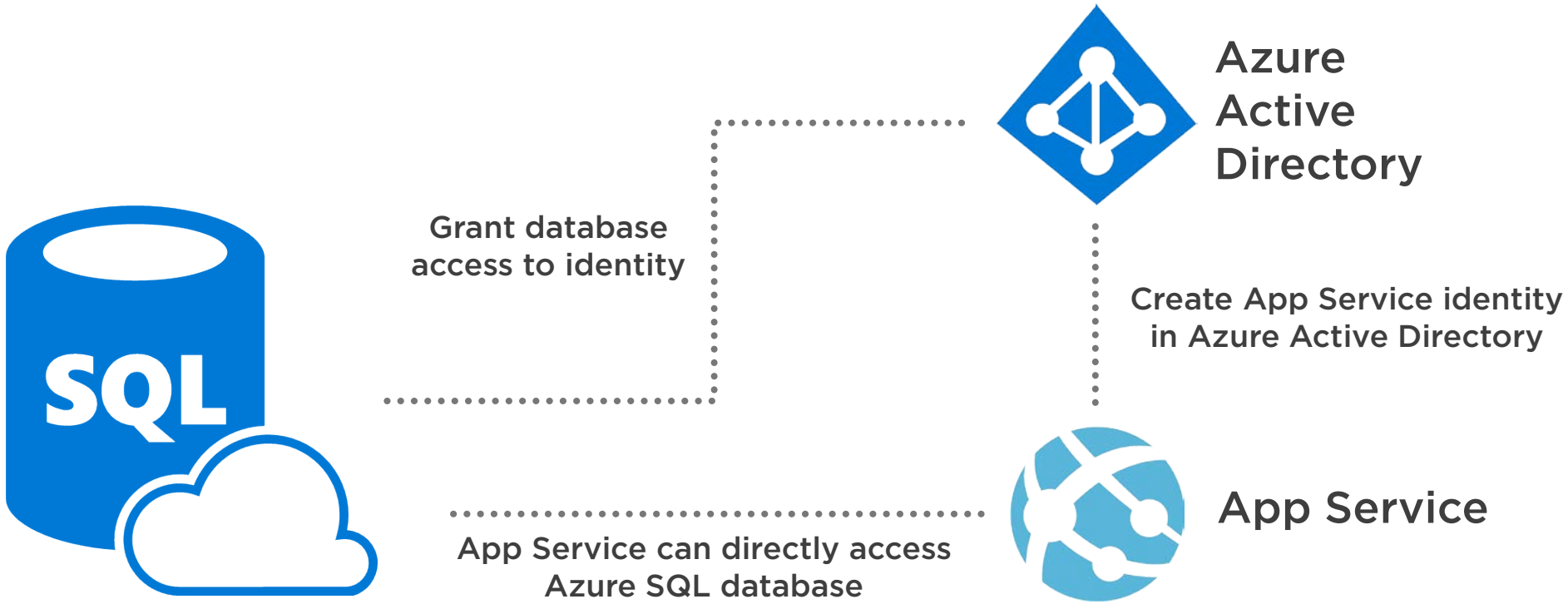
- Checking if soft-delete is enabled for our Vault
- Enabling soft-delete for the Key Vault
- Verifying that soft-delete is indeed enabled
- Deleting a vault protected by soft-delete
- Recovering the deleted key vault
- Purging a key vault

Managed Service Identity (MSI)

What is Managed Service Identity (MSI)?



What is Managed Service Identity (MSI)?



Why Managed Service Identity Is Recommended?

No need to authenticate to Azure Key Vault to get secrets

No client id and client secret is needed in the code

Easier to configure comparing to Azure Key Vault

You can authenticate to any service that supports Azure AD authentication

Services That Support Managed Service Identity

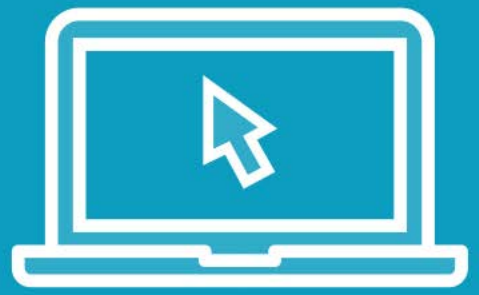
- Azure SQL
- Azure Service Bus
- Azure Storage
- Azure Key Vault
- Azure Resource Manager
- Azure Data Lake

```
<add name="SqlDataConnection" connectionString="data
source=zaalion.database.windows.net;initial
catalog=MyAddressBookPlus;persist security info=True;user
id=AppServiceLogin;password=P@$$w0rd;MultipleActiveResultSe
ts=True;" />
```

Secrets

Azure SQL database connection string

Demo



Enable Managed Service Identity for *MyAddressBook+*

Configure Azure SQL Database to grant access to the new identity

Update *MyAddressBook+* code

- Remove credentials (username, password) from connection string
- Modify ASP.NET code

Verify *MyAddressBook+* works as expected

```
<!- web.config -->

<!- before -->

<add name="SqlDataConnection" connectionString="data
source=zaalion.database.windows.net;initial catalog=MyAddressBookPlus;persist security
info=True;user id=AppServiceLogin;password=P@$$w0rd;MultipleActiveResultSets=True;" />

<!- after -->

<add name="SqlDataConnection" connectionString="data
source=zaalion.database.windows.net;initial catalog=MyAddressBookPlus;persist security
info=True;MultipleActiveResultSets=True;" />
```

Code Changes

Remove user id and password from SqlDataConnection

```
Install-Package Microsoft.Azure.Services.AppAuthentication -ProjectName
MyAddressBookPlus
```

Code Changes

Remove user id and password from SqlDataConnection

Install the Microsoft.Azure.Services.AppAuthentication Nuget package

```
<!– ContactRepository -->

var accesstoken = (new
AzureServiceTokenProvider()).GetAccessTokenAsync("https://database.windows.net/").Result;

db = new SqlConnection()            {

     AccessToken = accesstoken,

     ConnectionString = connectionstring

};
```

Code Changes

Remove user id and password from *SqlDataConnection*

Install the *Microsoft.Azure.Services.AppAuthentication* Nuget package

Update *SqlConnection* to use AAD access token for authentication

Useful Tools from Microsoft

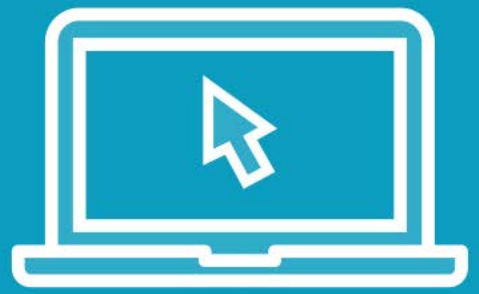
Azure Services Authentication Extension for Visual Studio 2017 update 5

Allows projects that use the Microsoft.Azure.Services.AppAuthentication library to access Azure resources using their Visual Studio accounts.

Microsoft Credential Scanner (preview)

Monitors all incoming commits on GitHub and checks for specific Azure tenant secrets such as Azure SQL connection strings.

Demo



Using Azure Services Authentication Extension

Summary

- Keys vs. Secrets
- Microsoft Azure Key Vault
- Managed Service Identity (MSI)
- Some tools to keep your eye on
- Demo
- Resources