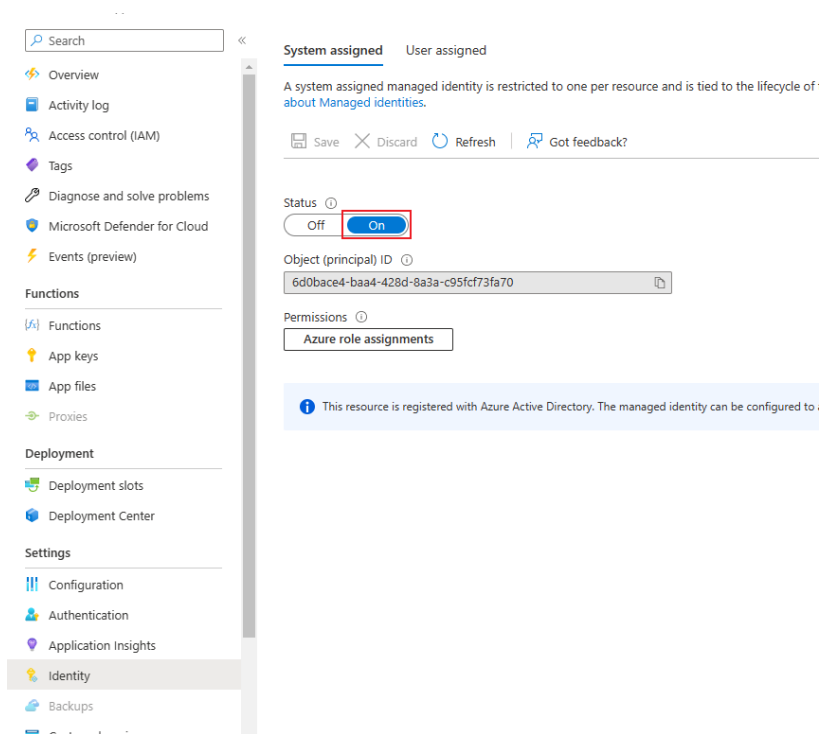


# Credential handling with Azure KeyVault

With this option, the secrets for Azure Functions are stored in an Azure Key Vault. An access policy is placed on the Key Vault which only allows the Managed Identity of an Azure Function to read the secret. On the Azure Function side, a Managed Identity is set up as well as an Environment Variable that is linked to the Secret in the Key Vault.

## Activate managed identity

To authenticate against the Key Vault you can activate the Managed Identity of your Function App. Thus, permissions can be given to the function app and used within the function.



## Create new Key Vault Secret

Create a new secret in the key vault and give this secret a unique name. This name will then be used to create a link from the Function App Variable to the Key Vault.

Search

<<

Generate/Import

Refresh

Restore Backup

View sample code

Manage deleted secrets

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Access policies

Events

Objects

Keys

Secrets

Certificates

Upload options

Manual

Name \* ⓘ

testsecret

Secret value \* ⓘ

.....

Content type (optional)

Set activation date ⓘ

Set expiration date ⓘ

Enabled

YesNo

Tags

0 tags

# Activate access policies

Go to access policies to create a new permission container for the Azure Function.

Diagnose and solve problems

Access policies

Events

Objects

Keys

Secrets

Certificates

Settings

Access configuration

Networking

Microsoft Defender for Cloud

Properties

Locks

Permission model

Grant data plane access by using a [Key Vault access policy](#) or [Azure RBAC](#)

☒ Vault access policy ⓘ

☐ Azure role-based access control ⓘ

Go to access policies

Resource access

Choose among the following options to grant access to specific resource types

☐ Azure Virtual Machines for deployment ⓘ

☐ Azure Resource Manager for template deployment ⓘ

☐ Azure Disk Encryption for volume encryption ⓘ

Select the necessary permissions for your function app.

## Create an access policy

kv-bkw-intune-prod-we

1 Permissions 2 Principal 3 Application (optional) 4 Review + create

Configure from a template

Select a template

### Key permissions

Key Management Operations

☐ Select all

☐ Get

☐ List

☐ Update

☐ Create

☐ Import

☐ Delete

☐ Recover

☐ Backup

☐ Restore

Cryptographic Operations

☐ Select all

☐ Decrypt

☐ Encrypt

☐ Unwrap Key

☐ Wrap Key

☐ Verify

☐ Sign

Privileged Key Operations

☐ Select all

☐ Purge

☐ Release

Rotation Policy Operations

☐ Select all

☐ Rotate

☐ Get Rotation Policy

☐ Set Rotation Policy

### Secret permissions

Secret Management Operations

☐ Select all

☒ Get

☒ List

☐ Set

☐ Delete

☐ Recover

☐ Backup

☐ Restore

Privileged Secret Operations

☐ Select all

☐ Purge

### Certificate permissions

Certificate Management Operations

☐ Select all

☐ Get

☐ List

☐ Update

☐ Create

☐ Import

☐ Delete

☐ Recover

☐ Backup

☐ Restore

☐ Manage Contacts

☐ Manage Certificate Authorities

☐ Get Certificate Authorities

☐ List Certificate Authorities

☐ Set Certificate Authorities

☐ Delete Certificate Authorities

Privileged Certificate Operations

☐ Select all

☐ Purge

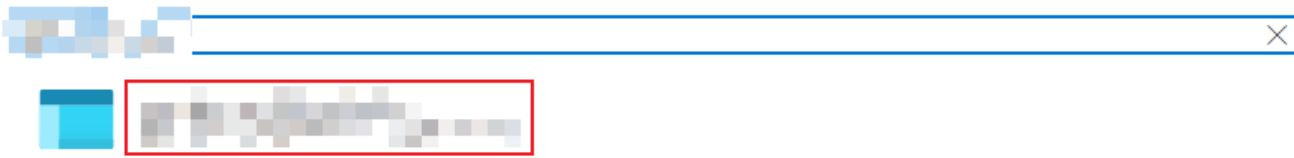
Previous Next

Select the appropriate Managed Identity from the function, which was created earlier.

1 Permissions 2 Principal 3 Application (optional) 4 Review + create

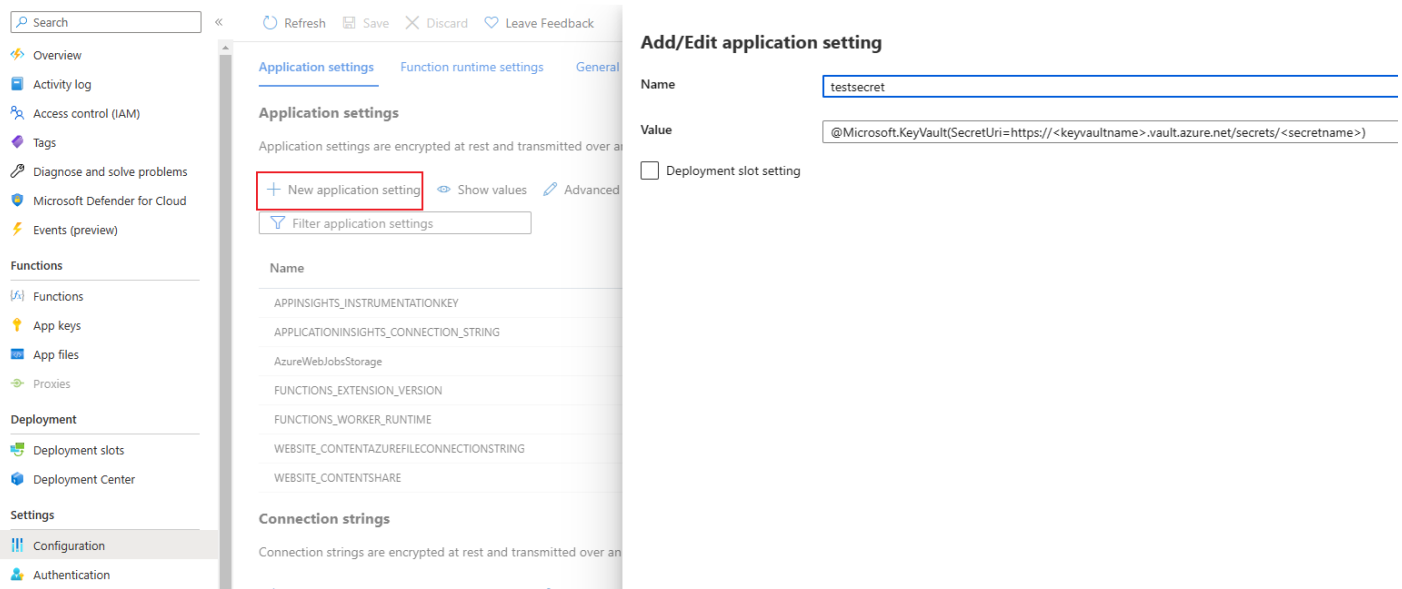
Only 1 principal can be assigned per access policy.

Use the new embedded experience to select a principal. The previous popup experience can be accessed here. [Select a principal](#)



# Add environment variable link

Then you can get the Key Vault name and secret name. With this information you can create an Application setting which will create an environment variable to use in the Azure Functions code. Set the name of the Environment Variable and create a link to the corresponding secret in the Key Vault.



@Microsoft.KeyVault(SecretUri=https://<keyvaultname>.vault.azure.net/secrets/<secretname>)

# Usage in function code

When everything is implemented as described you can use the variable accordingly:

```
$testsecret = $env:testsecret
```

Revision #2

Created 9 January 2023 14:19:02

Updated 21 July 2024 15:11:16