

# Azure Meetup Hamburg

Praxisnahes Arbeiten mit Azure Functions, Graph und Managed Identities

Press Space for next page →



# Agenda?

- What is a Microsoft managed identity?
- Prerequisites
- Prerequisites VSCode
- Storage Account Explorer & Azurite
- Service Principal Local Deveolpment
- Azure Function App
  - Configure Managed Identity
  - Set Azure Function Graph Permissions
- Start local Function Development
  - Manage local.settings.json
  - Configure requiremnts
  - Run and deploy Azure Function local

# Compare Microsoft Azure Managed Identity & Service Principal

Feature	Managed Identity	Service Principal
<b>Description</b>	An Azure service that automatically manages Azure AD identities.	An Azure AD object used to authenticate and authorize applications.
<b>Authentication Mode</b>	Automatically managed by Azure.	Must be manually configured and managed.
<b>Lifecycle Management</b>	Azure automatically manages the lifecycle.	Developers need to manually manage the lifecycle.
<b>Security</b>	Reduced risk as there's no need to store credentials in code or configurations.	Higher risk as credentials need to be managed and securely stored.
<b>Complexity</b>	Easier to use as it requires less manual configuration.	More complex in setup and management.

# Prerequisites

To use Azure Function local Development, some requirements must be fulfilled

- **Storage Explorer** <https://azure.microsoft.com/de-de/products/storage/storage-explorer>
- **Node JS** <https://nodejs.org/en>
- **PowerShell 7** <https://github.com/PowerShell/PowerShell/releases/download/v7.4.1/PowerShell-7.4.1-win-x64.msi>
- **Azurite** 'npm install -g azurite'
- **Azure Function Core Tools** <https://go.microsoft.com/fwlink/?linkid=2174087>
- VSCode Extensions
  - Azure Tools
  - Azure Functions

# Storage Explorer & Azurite

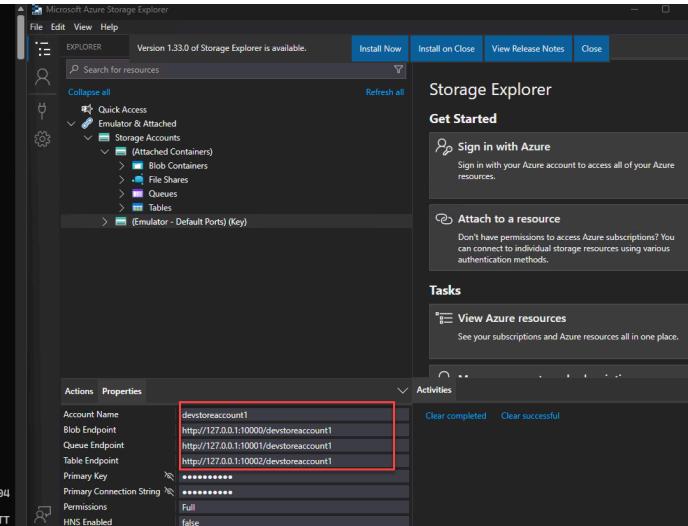
Starting the storage emulator using 'Azurite' in the console or in VSCode

## Azurite

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\JacobMeissner> azurite
[node:12300] [DEP0000] DeprecationWarning: The 'punycode' module is deprecated.
Please use a userland alternative instead.
[Use 'node --trace-deprecation ...` to show where the warning was created]
Azurite Blob service is starting at http://127.0.0.1:10000
Azurite Blob service is successfully listening at http://127.0.0.1:10000
Azurite Queue service is starting at http://127.0.0.1:10001
Azurite Queue service is successfully listening at http://127.0.0.1:10001
Azurite Table service is starting at http://127.0.0.1:10002
Azurite Table service is successfully listening at http://127.0.0.1:10002
Azurite Blob service is closing...
Azurite Queue service is closing...
Azurite Table service is closing...
Azurite Table service successfully closed
Azurite Blob service successfully closed
Azurite Queue service successfully closed
PS C:\Users\JacobMeissner> azurite
```

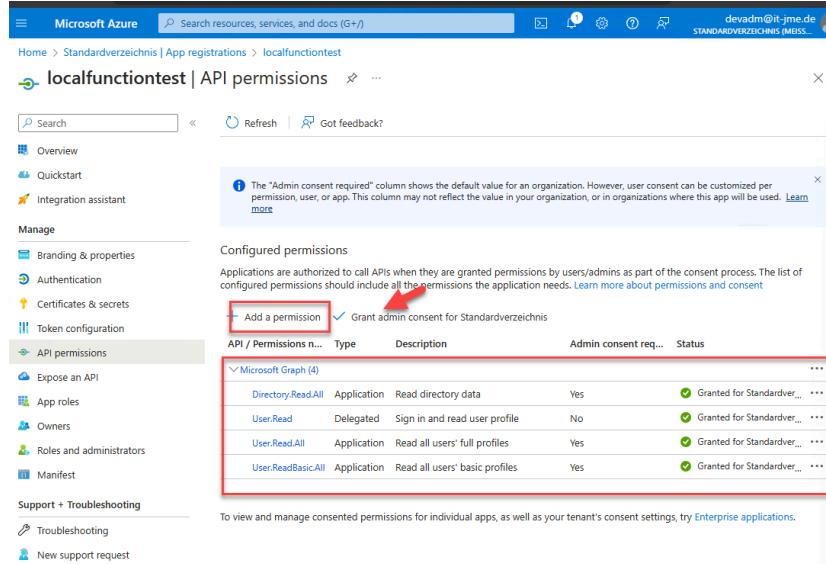
```
(node:19228) [DEP0000] DeprecationWarning: The 'punycode' module is deprecated.
Please use a userland alternative instead.
[Use 'node --trace-deprecation ...` to show where the warning was created]
Azurite Blob service is starting at http://127.0.0.1:10000
Azurite Blob service is successfully listening at http://127.0.0.1:10000
Azurite Queue service is starting at http://127.0.0.1:10001
Azurite Queue service is successfully listening at http://127.0.0.1:10001
Azurite Table service is starting at http://127.0.0.1:10002
Azurite Table service is successfully listening at http://127.0.0.1:10002
127.0.0.1 - [26/Mar/2024:13:08:30 +0000] "GET /devstoreaccount1?comp=properties&restype=account HTTP/1.1" 200 -
127.0.0.1 - [26/Mar/2024:13:08:31 +0000] "GET /devstoreaccount1?comp=properties&restype=account HTTP/1.1" 200 -
127.0.0.1 - [26/Mar/2024:13:08:34 +0000] "GET /devstoreaccount1?comp=list&listStyle=include&maxResults=50 HTTP/1.1" 200 -
127.0.0.1 - [26/Mar/2024:13:08:35 +0000] "GET /devstoreaccount1?comp=list&listStyle=include&maxResults=50 HTTP/1.1" 200 -
127.0.0.1 - [26/Mar/2024:13:08:35 +0000] "GET /devstoreaccount1?comp=logs&restype=container HTTP/1.1" 404 -
127.0.0.1 - [26/Mar/2024:13:08:35 +0000] "GET /devstoreaccount1/?204lobchange feed&restype=container HTTP/1.1" 404 -
127.0.0.1 - [26/Mar/2024:13:09:05 +0000] "GET /devstoreaccount1?restype=service&comp=properties HTTP/1.1" 200 -
127.0.0.1 - [26/Mar/2024:13:09:14 +0000] "HEAD /devstoreaccount1/azure-webjobs-hosts/locks/cpcjmeisimtjk-1545554877/Host.Functions.Function02.Listener HTTP/1.1" 404 -
127.0.0.1 - [26/Mar/2024:13:09:14 +0000] "PUT /devstoreaccount1/azure-webjobs-hosts/locks/cpcjmeisimtjk-1545554877/Host.Functions.Function02.Listener?comp=lease HTTP/1.1" 204
```



# Azure Service Principal

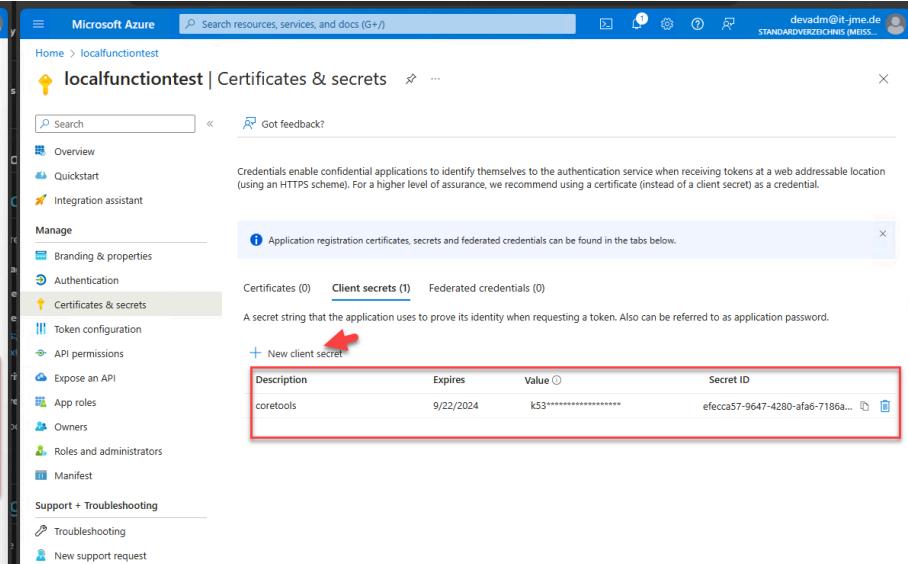
Creating an Azure Service Principal for the local environment

```
New-AzADServicePrincipal -DisplayName "MeinServicePrincipal"
```



The screenshot shows the 'API permissions' section of the Azure portal for the 'localfunctiontest' app registration. The 'Add a permission' button is highlighted with a red box and a red arrow. The table below lists several permissions:

API / Permissions n...	Type	Description	Admin consent req...	Status
Microsoft Graph (4)				
Directory.Read.All	Application	Read directory data	Yes	Granted for Standardver...
User.Read	Delegated	Sign in and read user profile	No	Granted for Standardver...
User.Read.All	Application	Read all users' full profiles	Yes	Granted for Standardver...
User.ReadBasic.All	Application	Read all users' basic profiles	Yes	Granted for Standardver...



The screenshot shows the 'Certificates & secrets' section of the Azure portal for the 'localfunctiontest' app registration. The 'Client secrets' tab is selected. A red box highlights the 'New client secret' button, and a red arrow points to it. The table shows one existing client secret:

Description	Expires	Value	Secret ID
coretools	9/22/2024	k53*****	efeca57-9e47-4280-afa6-7186a...

# Azure Function App & Enable Managed Identity

Create an Azure Function App (e.g. Powershell) and activate the Managed Identity Configuration

The screenshot shows the Azure portal interface for a Function App named 'fateeuw001'. The left sidebar lists various settings like Functions, Deployment, and Settings. The main content area shows the 'Identity' configuration for a 'System assigned' managed identity. A red box highlights the 'Status' section where the toggle switch is set to 'On'. Another red box highlights the 'Permissions' section which lists 'Azure role assignments'. A note at the bottom states: 'This resource is registered with Microsoft Entra ID. The managed identity can be configured to allow access to other resources. Be careful when making changes to the access settings for the managed identity because it can result in failures.'

fateeuw001 | Identity

Function App

Search

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Events (preview)

Log stream

Functions

App keys

App files

Proxies

Deployment

Deployment slots

Deployment Center

Settings

Configuration

Authentication

System assigned User assigned

A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. You can grant permissions to the managed identity by using Azure role-based access control (Azure RBAC). The managed identity is authenticated with Microsoft Entra ID, so you don't have to store any credentials in code.

Status ⓘ

Off On

Object (principal) ID ⓘ

1b105bf7-37e9-48aa-9534-62b939d

Permissions ⓘ

Azure role assignments

This resource is registered with Microsoft Entra ID. The managed identity can be configured to allow access to other resources. Be careful when making changes to the access settings for the managed identity because it can result in failures.

# Managed Identity Graph Permission

Following the creation of the Managed Identity, the Graph Endpoint must be authorised so that it can access the requested data in the Managed Identity Context.

```
$tenantId = "00000000-0000-0000-0000-000000000000" # Replace with your tenant ID
$graphApiAppId = "00000003-0000-0000-c000-000000000000" # Well known ID
$msiName = "MSINAME" # Name of your managed identity e.g. name of Function or Logic App
$graphPermissions = @("Directory.Read.All", "User.Read.All") # Add or remove permissions

Connect-AzureAD -TenantId $tenantId
$msi = Get-AzureADServicePrincipal -Filter "displayName eq '$msiName'" # Can take a few seconds, add a sleep if necessary
$graphApiAppRegistration = Get-AzureADServicePrincipal -Filter "appId eq '$graphApiAppId'"
$appRoles = $graphApiAppRegistration.AppRoles | Where-Object { $graphPermissions -contains $_.Value -and $_.AllowedMembers.Count -gt 0 }
foreach ($appRole in $appRoles) {
    New-AzureAdServiceAppRoleAssignment -ObjectId $msi.ObjectId -PrincipalId $msi.ObjectId -ResourceId $graphApiAppRegistration.ObjectId
}
```

# Function Requirements

```
@{

# For latest supported version, go to 'https://www.powershellgallery.com/packages/Az'. Uncomment the next line and : 
# 'Az' = 'MAJOR_VERSION.*'
'Az.Accounts' = '2.13.2'
'PSReadLine' = '2.3.4'
'Microsoft.Graph.Authentication' = '2.9.0'

}
```

