emdash on SQL AZURE

Contents

[Introduction 2](#_Toc133478387)

[Option 1 – user name and password 2](#_Toc133478388)

[Run the relevant emdash deployment script 2](#_Toc133478389)

[Common steps for Option 2 and Option 3 3](#_Toc133478390)

[Creating a Service Principal in Azure 3](#_Toc133478391)

[Adding a Service Principal to the emdash database as a user 4](#_Toc133478392)

[Option 2- authenticating using a client secret 5](#_Toc133478393)

[Create the client secret 5](#_Toc133478394)

[Set up system environment variables 5](#_Toc133478395)

[Run the emdash deployment script 6](#_Toc133478396)

[Option 3 – authenticating using a certificate 6](#_Toc133478397)

[Create the certificate 6](#_Toc133478398)

[Upload the certificate to Azure 6](#_Toc133478399)

[Set up environment variables 7](#_Toc133478400)

[Run the emdash deployment script 7](#_Toc133478401)

[Points to Note 7](#_Toc133478402)

# Introduction

emdash classic and emdash devops are capable of running on a SQL Azure (PAAS) database. In this scenario both products support three methods to login to the emdash database.

These are:

1. Authenticate with the emdash database using a user name and password in the connection string
2. Authenticate with the emdash database using a Service Principal and a client secret
3. Authenticate with the emdash database using a Service Principal and a certificate

Of the three options, option 3 is preferred as it is the most secure and does not require the management of passwords or secrets.

All three options assume that an empty shell emdash SQL Azure database has already been created in the target subscription.

# Option 1 – user name and password

## Run the relevant emdash deployment script

The emdash deployment scripts for deploying emdash to a SQL Azure database using username and password are checked into the EMDASHSCRIPTS Git repository.

Use ***emdash\_Deploy\_SQLAzureUsernamePassword.ps1*** or ***emdashDevops\_Deploy\_SQLAzureWithUserNamePassword.ps1*** depending on which product you are deploying.

Graphical user interface, text, chat or text message

Description automatically generated

# Common steps for Option 2 and Option 3

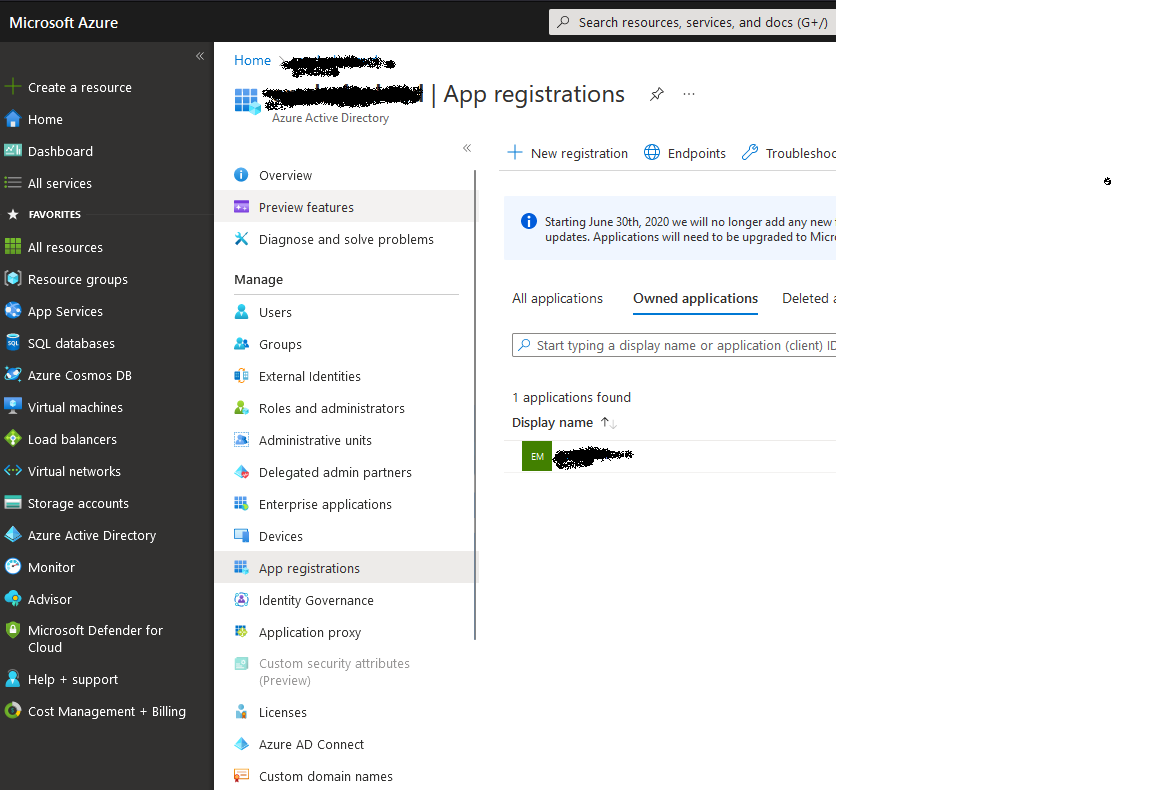
Option 2 and Option 3 share the need for an Azure service principal to be created. The next two sections describe how this is done prior to describing option 2 and option 3 in detail.

# Creating a Service Principal in Azure

Note it is recommended to use separate service principals for the emdash database user and for emdash access to Azure functions e.g. creating servers or databases. This keeps the identities and permissions distinct.

The following steps are performed in the Azure portal to create the service principal (SPN). This assumes we are creating an SPN called ***emdash-app***

1. Navigate to Azure Active Directory 🡪 App registrations

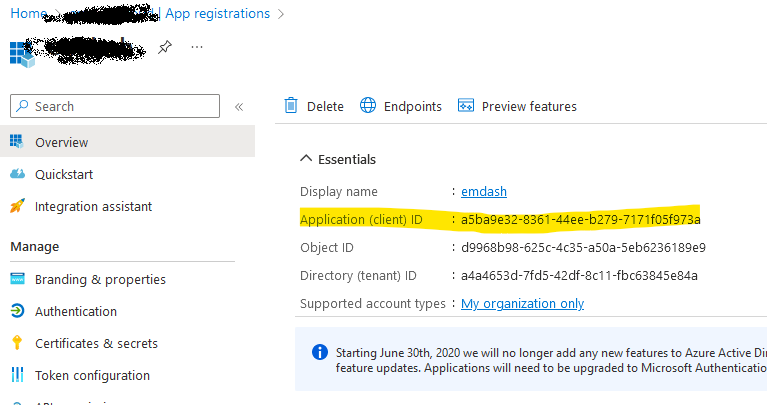


1. Select New registration and enter the following details then click Register.

Graphical user interface, text, application, email

Description automatically generated

1. Navigate to the app registration you just created and note down the ***Application (client) id***:



The service principal creation process is now complete.

# Adding a Service Principal to the emdash database as a user

Connect to SQL Server using an Azure AD SQL administrator account (not a SQL account) and run the following commands (this assumes that the emdash database is called EMDASH

Using MASTER

* CREATE LOGIN [emdash-app] FROM EXTERNAL PROVIDER

Using EMDASH

* CREATE USER [emdash-app] FROM LOGIN [emdash-app]
* EXEC sp\_addrolemember db\_datareader, [emdash-app]
* EXEC sp\_addrolemember db\_datawriter, [emdash-app]
* EXEC sp\_addrolemember db\_owner, [emdash-app]
* GRANT EXECUTE TO [emdash-app]

# Option 2- authenticating using a client secret

In order for emdash to authenticate using a client secret the following additional steps need to be performed.

## Create the client secret

Login to the Azure portal and navigate to the app registration set up previously (emdash-app)

Graphical user interface, text, application, email

Description automatically generated

Click ***New client secret*** and enter the secret details with a recommended expiry date based on your organizations policies.

Graphical user interface, text, application

Description automatically generated

You must copy the secret Value immediately after creation as once you have navigated away from the page it can no longer be retrieved.

## Set up system environment variables

The following system environment variables need to be set up on **each emdash server, including workflow servers**

* ***$server*** should be set to the emdash database server name
* ***$database*** should be set to the emdash database name.
* EmdashTenantId should be set to the Azure tenant id where the emdash SPN was created.
* EmdashClientId should be set to the Application (client) id of the emdash-app SPN
* EmdashClientSecret should be set to the client secret Value created in the previous step.

[System.Environment]::SetEnvironmentVariable('EmdashUseAadAuthentication','true','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashDatabaseServer',$server,'Machine')

[System.Environment]::SetEnvironmentVariable('EmdashTenantId','a4a4653d-7fd5-42df-8c11-fbc63845e84a','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashAuthority','https://login.windows.net','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashClientId','a5ba9e32-8361-44ee-b279-7171f05f973a','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashClientSecret','SHZ8Q~Q~fcsiaUWfZ9QnGHEwAAk7JZ07hktZKar1','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashCertificateThumbprint','','Machine') # set this to be the emdash SPN certificate thumbprint and set EmdashClientSecret to empty string if required

[System.Environment]::SetEnvironmentVariable('EmdashScope','https://database.windows.net/.default','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashDatabase',$database,'Machine')

[System.Environment]::SetEnvironmentVariable('EmdashEncryptConnection','true','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashTrustServerCertificate','false','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashMultipleActiveResultSets','true','Machine')

[System.Environment]::SetEnvironmentVariable('EmdashConnectionTimeout','30','Machine')

## Run the emdash deployment script

The emdash deployment scripts for deploying emdash to a SQL Azure database using a client secret are checked into the EMDASHSCRIPTS Git repository.

Use ***emdash*\_Deploy\_SQLAzureWithAccessToken*.ps1*** or ***emdashDevops*\_Deploy\_SQLAzureWithAccessToken*.ps1*** depending on which product you are deploying.

# Option 3 – authenticating using a certificate

In order for emdash to authenticate using a certificate the following additional steps need to be performed.

## Create the certificate

The following PowerShell can be run on the emdash server to create a self-signed certificate. Alternatively a certificate from an organizational certificate authority or third party provider can be used. **Note – the pfx password below should be replaced with a different secure value and the pfx file and password should not be shared outside the system administration team.**

$certname = "emdashServicePrincipalCert"

$pfxPassword = "5j35hkefhisety7ewy7trretjhsjf"

$exportPfxPath = "C:\Users\EMDASH\emdashServicePrincipalCert.pfx"

$cert = New-SelfSignedCertificate -Subject "CN=emdashServicePrincipalCert" -CertStoreLocation "Cert:\CurrentUser\My" -KeyExportPolicy Exportable -KeyLength 2048 -KeyAlgorithm RSA -HashAlgorithm SHA256 -NotAfter "31/12/2035" `

-FriendlyName $certname -KeyUsageProperty All

$mypwd = ConvertTo-SecureString -String $pfxPassword -Force -AsPlainText

Export-PfxCertificate -Cert $cert -FilePath $exportPfxPath -Password $mypwd

Note that the certificate must be imported into the Personal store of the emdash service account on **each emdash server, including workflow servers.** This can be done by importing the exported pfx file on each server.

## Upload the certificate to Azure

Export the certificate from the local user Personal store to a .CER file format. Login to the Azure portal and under the emdash-app app registration upload the certificate:

Graphical user interface, text, application

Description automatically generated

Also make a note of the certificate thumbprint as this is required in the next step.

## Set up environment variables

The same environment variable set up commands for client secret can be used for certificate based authentication. The only difference is that

* EmdashCertificateThumbprint should be set to the certificate thumbprint from the previous step
* EmdashClientSecret should be set to an empty string or removed from the script completely.

## Run the emdash deployment script

The emdash deployment scripts for deploying emdash to a SQL Azure database using a certificate are checked into the EMDASHSCRIPTS Git repository.

Use ***emdash*\_Deploy\_SQLAzureWithAccessToken*.ps1*** or ***emdashDevops*\_Deploy\_SQLAzureWithAccessToken*.ps1*** depending on which product you are deploying.

After running the set up ensure that the emdash application pool in IIS is recycled and that all emdash workflow services are restarted so that the new settings are picked up.

# Points to Note

* When emdash is set to use an SPN with client secret or certificate to authenticate, the emdash connection strings in the various config files will still indicate Integrated Security. The value is not used anywhere and is simply the default that comes when emdash is deployed. All connections are made using the properties set in the environment variables.
* Reminders should be set for when the emdash SPN certificates or secrets will expire and these should be updated ahead of expiry and the relevant environment variables updated. This will also require an emdash application pool recycle and the emdash workflow services to be restarted.