

AAD Authentication Provisioning

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:34 AM PDT

Contents

- [Options and limitations](#)
- [Prerequisites](#)
 - [AAD CSS Team](#)
 - [Associate your Azure AD Directory with your Azure SQL DB](#)
- [Create Contained Database Users in your Database mappe...](#)
- [Connect to DB using AAD](#)

Options and limitations

We can provision using Azure Portal & PowerShell

Native members: A member created in Azure AD in the initial domain or in a customer domain. For more information, see [Add your own domain name to Azure AD](#). Federated domain members: A member created in Azure AD with a federated domain. For more information, see [Windows Azure now supports federation with Windows Server Active Directory](#). Imported members from other Azure Active Directories who are native or federated domain members. Active Directory groups created as security groups (Groups created via Forefront Identity Manager).

Additional considerations

To enhance manageability, we recommended you provision a dedicated Azure Active Directory group as an administrator. At any given time only one administrator can be provisioned.

Only the SQL Azure Active Directory admin can connect to the master database using an Azure Active Directory account. All other Azure Active Directory users must connect to a user database where their contained database user account was created. Connection timeouts may occur.

To work around this problem, increase the connection timeout to 30 sec. Connection from BI stack (SSRS, SSAS, SSIS), Entity Framework, Visual Studio and Excel using AAD Login authentication are not supported, use SQL Server authentication instead. Azure Active Directory authentication only supports the .NET Framework Data Provider for Sqlserver (at least version .NET Framework 4.6). Hence Management Studio (available with SQL Server 2016) and data-tier applications (DAC and. bacpac) can connect.

sqlcmd.exe, bcp.exe, import/export wizard cannot connect using AAD logins because they don't use .Net framework data provider for SQL Server (for such tools use SQL server authentication instead, support will be added in later versions).

Two-factor authentication is not supported.

Does not support multi-tenant applications that require users from multiple Azure AD directories to authenticate to a single database or different databases in the same server.

All databases within a server can be associated with only one Azure AD directory is not available for client applications running in Azure Websites or Azure Web Jobs




Prerequisites

- Create and Populate an Azure Active Directory
- Ensure your Database is in Azure SQL DB V12
- Associate your Azure AD Directory with your Azure SQL DB Subscription
- Create an Azure Active Directory Administrator for Azure SQL Server
- Create Contained Database Users in your Database mapped to Azure AD Identities
- Configure your Client Computer
- Connect to your database by using Azure Active Directory Identities
- Create and Populate an Azure Active Directory.

Note : This step needs collaboration with AAD CSS Team

Create an Azure AD directory and populate it with users. This includes:

- Create the initial domain Azure AD.
- Federate an on-premises Active Directory Domain Services with Azure Active Directory.

For more information, see [Add your own domain name to Azure AD](#) , [Administering your Azure AD directory](#) , and Manage [Azure AD using Windows PowerShell](#) .


AAD CSS Team

AAD team will help on the following -

- If customer has not created an Azure AD or has not configured Federation for an on-Premise AD, AAD Team will help customer do that (Request Customer to create a new case, our case scope is just to get AADA enabled for Azure SQL DB).
- If customer has already an AAD or has already configured Federation for an on-Premise AD, we can cut a subcase to AAD Team to validate if current configuration is as per the recommendations and best practices. (Dispatch a subcase to AAD CSS Team)

Associate your Azure AD Directory with your Azure SQL DB

(Take help from Azure AD CSS Team if you are not sure of impact)

To associate your database with the Azure AD directory for your organization, make the directory a trusted directory for the Azure subscription hosting the database. For more information, see [How Azure subscriptions are associated with Azure AD](#) .

The following procedures create an Azure AD based administrator for Azure SQL Server in two ways: By using the Azure portal and by using PowerShell commands.

- Connect to your Azure portal by using an Azure subscription administrator.
- On the left banner, select SETTINGS.

- Click your subscription, and then click the DIRECTORY.
- In the DIRECTORY box, select the Azure Active Directory that is associated with your SQL Server, and then click the arrow for next.
- In the CONFIRM directory Mapping dialog box, confirm that "All co-administrators will be removed."
- Click OK to reload the portal.

Creating via PowerShell

```

`--Connecting to Azure
PS C:\> Add-AzureAccount
VERBOSE: Account "user@domain" has been added.
VERBOSE: Subscription "Microsoft Azure Internal Consumption" is selected as the default subscription.
VERBOSE: To view all the subscriptions, please use Get-AzureSubscription.
VERBOSE: To switch to a different subscription, please use Select-AzureSubscription.

```

Id	Type	Subscriptions	Tenants
user@domain.com	User	aaxx9933-282-xxxx-oooo-ytytytytytr 99999999-000e-4990-ae2e-7f51f95c2117	77779933-282-xxxx-oooo-ytytytyty 99999999-ff03-40ed-8ce6-67b6b

```

--Selecting the Subscription on which i will setup the SQL Azure DB Server
PS C:\> Select-AzureSubscription

```

```

cmdlet Select-AzureSubscription at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
SubscriptionName: Microsoft Azure Internal Consumption

```

```

--Switching the Azure CMDlet mode to Resource Manager Mode
PS C:\> Switch-AzureMode AzureResourceManager
WARNING: The Switch-AzureMode cmdlet is deprecated and will be removed in a future release.
VERBOSE: Removing Azure module...
VERBOSE: Removing the imported "aliases" variable.
VERBOSE: Removing the imported "Add-WAPackEnvironment" alias.
.
.
VERBOSE: Importing alias 'Get-AzureSqlDatabaseServerAuditingPolicy'.
VERBOSE: Importing alias 'Get-AzureStorageContainerAcl'.
VERBOSE: Importing alias 'Remove-AzureSqlDatabaseServerAuditing'.
VERBOSE: Importing alias 'Set-AzureSqlDatabaseServerAuditingPolicy'.
VERBOSE: Importing alias 'Start-CopyAzureStorageBlob'.
VERBOSE: Importing alias 'Stop-CopyAzureStorageBlob'.
VERBOSE: Importing alias 'Use-AzureSqlDatabaseServerAuditingPolicy'.

```

```

-- Setting up Azure AD administrator user named user@domain.com for Server Named pradmauses in Group-3 Resour

```

```

PS C:\> $server = Get-AzureSqlServer -ResourceGroupName Group-3 -ServerName pradmauses
PS C:\> $server | Set-AzureSqlServerActiveDirectoryAdministrator -DisplayName " user@domain.com"

```

ResourceGroupName	ServerName	DisplayName	ObjectId
Group-3	pradmauses	user@domain.com	99999999-8f2a-4906-a53f

```

--Validating who is the Azure AD administrator for Server Named pradmauses in Group-3 Resource Group
PS C:\> $server = Get-AzureSqlServer -ResourceGroupName Group-3 -ServerName pradmauses

```

```

PS C:\> $server | Get-AzureSqlServerActiveDirectoryAdministrator

```

ResourceGroupName	ServerName	DisplayName	ObjectId
Group-3	pradmauses	user@domain.com	99999999-8f2a-4906-a53f

```

PS C:\> Get-AzureSqlServerActiveDirectoryAdministrator -ResourceGroupName Group-3 -ServerName pradmauses | For

```

```

ResourceGroupName : Group-3
ServerName        : pradmauses
DisplayName       : user@domain.com
ObjectId          : 99999999-8f2a-4906-a53f-b4360ef74607

```

```

-- Using ObjectID parameter to force uniqueness , representing Azure AD ObjectID for pradm@microsoft.com
PS C:\> $server = Get-AzureSqlServer -ResourceGroupName Group-3 -ServerName pradmauses
PS C:\> $server | Set-AzureSqlServerActiveDirectoryAdministrator -DisplayName " user@domain.com" -ObjectId "99
8f2a-4906-a53f-b4360ef74607"

```

ResourceGroupName	ServerName	DisplayName	ObjectId
Group-3	pradmauses	user@domain.com	99999999-8f2a-4906-a53f-b4360ef74607

Group-3

pradmauses

user@domain.com

99999999-8f2a-4906-a53f

```
-- This can be used to identify the TenantID (Azure AD ObjectID )
PS C:\> Get-AzureSubscription
```

```
SubscriptionId      : aaxx9933-282-xxxx-oooo-ytytytytytytr
SubscriptionName    : dsdazure-pradm
Environment         : AzureCloud
SupportedModes      : AzureServiceManagement,AzureResourceManager
DefaultAccount      : user@domain.com
Accounts            : { user@domain.com}
IsDefault           : False
IsCurrent           : False
CurrentStorageAccountName :
TenantId            : 77779933-282-xxxx-oooo-ytytytytytytr`
```

```
SubscriptionId      : 99999999-000e-4990-ae2e-7f51f95c2117
SubscriptionName    : Microsoft Azure Internal Consumption
Environment         : AzureCloud
SupportedModes      : AzureServiceManagement,AzureResourceManager
DefaultAccount      : user@domain.com
Accounts            : { user@domain.com}
IsDefault           : True
IsCurrent           : True
CurrentStorageAccountName :
TenantId            : 77779933-282-xxxx-oooo-ytytytytytytr`
```

Create Contained Database Users in your Database mapped to Azure AD Identities :

Using the above steps you have now successfully provisioned an Azure AD Administrator who can manage you Azure SQL DB along with the SQL Login Admin that gets created when provisioning the server. Now you need to provide permission to other AD Accounts from your domain to access specific databases based on business need. Azure Active Directory authentication requires database users to be created as contained database users. A contained database user based on an Azure AD identity is a database user that does not have a login in the master database, and which maps to an identity in the Azure AD directory that is associated with the database. The Azure AD user can be either an individual user account, a group, or an Azure AD token.

You can now use Management studio to connect to SQL Azure DB with the Credentials of Azure AD Login, there are 2 new authentication methods that are designed for this purpose which internally uses ADAL implementations to talk to Azure AD. Before you proceed ahead please review the next step which talks about the client requirements. We need to make sure all the requirements are met before we start to use a client to connect to Azure SQL DB using AAD accounts.

Authentication Methods:

- Active Directory Password Authentication
- Active Directory Integrated Authentication

If you try to choose the "Windows Authentication" and try to connect to SQL Azure DB with an AAD Login, it will fail with the below error message

TITLE: Connect to Server

 Cannot connect to servername.database.windows.net.

 ADDITIONAL INFORMATION:

Windows logins are not supported in this version of SQL Server. (Microsoft SQL Server, Error: 40607)

For help, click: <http://go.microsoft.com/fwlink?ProdName=Microsoft%20SQL%20Server&EvtSrc=MSSQLServer&EvtID=406>

 BUTTONS: OK



Use any of these 2 authentication methods and present the credential of the newly provisioned Azure AD Login, please make sure to set the context of the connection to the required user database since other AD Logins will be added a contained database users restricting their permission to the specific database as per their business need.

- Adding Federated Login as Contained User : CREATE USER [bob@contoso.com] FROM EXTERNAL PROVIDER
- Adding Managed AD Login as Contained User : CREATE USER [alice@fabrikam.onmicrosoft.com] FROM EXTERNAL PROVIDER
- Adding Managed / Federated Group as Contained User : CREATE USER [hr_group] FROM EXTERNAL PROVIDER -- For groups just the name is enough

When you create a database user, that user receives the CONNECT permission and can connect to that database as a member of the PUBLIC role. Initially the only permissions available to the user are any permissions granted to the PUBLIC role, or any permissions granted to any Windows groups that they are a member of. Once you provision an Azure AD-based contained database user, you can grant the user additional permissions, the same way as you grant permission to any other type of user. Azure AD users are marked in the database metadata with type E (EXTERNAL_USER) and for groups with type X (EXTERNAL_GROUPS).

Client computer from where applications or users connect to azure sql db using AAD, must have .Net Framework 4.6 and above, and AAD Authentication Library for SQL Server (ADALSQL.DLL) [amd64](#) or [x86](#).

Connect to DB using AAD

Once this has been done you can start using these database users to connect to SQL Azure DB using the "Active Directory Password Authentication" or "Active Directory Integrated Authentication" options from SSMS. From the application perspective you can use Authentication="Active Directory Password" or Authentication="Active Directory Integrated" keywords in your connection string and passing required credentials or by directly passing the previously acquired token.

How good have you found this content?



-