Service Principle Not able to connect

Last updated by | Vitor Tomaz | Feb 18, 2021 at 2:30 AM PST

Service Principle Not able to connect

Contents

Issue

Service principal or application is not able to connect to SQL DB

Mitigation

The C# code below allows you to troubleshoot this problem in two steps:

1. Obtain an Azure AD token
2. Pass this token to SQL DB

If needed, the encrypted user token can also be available to the support team (see the blog below)

Before building and running the code sample, perform the following steps:

1. Create a Service Principal in Azure AD for your service and obtained the following information required to execute the code sample below
   1. Application ID of the Service Principal (SP)
      clientId = "<appId>"; // Application ID of the SP (e.g. string clientId = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" 😊
   2. Copy the “Display Name” of your application which will be used in step 3)
      (e.g.”debugapp” as a “Display Name” for the app above)
   3. Azure AD tenant ID
      aadTenantId = "<tenantId>"; //Azure AD tenant ID (e.g. string aadTenantId = "xxxxxxx-xxxxx-xxxx-xxxx-xxxxxxxxxxxx" 😊
   4. Client (SP) secret key clientSecretKey = "secretKey"; // Application secret key (e.g. string clientSecretKey = "xxxxxx/xxxxxxxxxxxxxxxxxxxxxxxxxx/xxx" 😊 To obtain above information follow the steps indicated in the link below https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-porta...
2. Find the Azure SQL Server name and Database name
   serverName = "<serverName>"; // server name: myserver.database.windows.net
   (e.g. string serverName = "myserver.database.windows.net" 😊

databaseName = "<databaseName>" // database name: test;
(e.g. string databaseName = "test" 😊

3. Using SSMS to connect to SQL DB (e.g. “test”) as an Azure AD user with proper Azure AD permissions (e.g. Azure AD admin for SQL DB), create an application user from step 1 above. Execute the T-SQL statement create user command `create user [app display name] from external provider .`
   Example using “debugapp” as a display name form step1
   ```
   create user [debugapp] from external provider
   ```
   Note that the create user command grants this user a connect permission to the database, which is sufficient enough to execute the sample program below.
4. Copy and execute the program indicated below

Below is an example of the program output.

```
PS C:\0.0.debug.code\consoleapplication2\bin\debug>
PS C:\0.0.debug.code\consoleapplication2\bin\debug> ./ConsoleApplication2.exe
Time 49:57.331
Got token at 49:58.045
Total time to get token in milliseconds 714.392
Starting to open connection at 49:58.081
Got connection at 49:58.286
Total time to establish connection in milliseconds 204.9834
Starting to run query at 49:58.287
1
Completing running query at 49:58.312
Total time to execute query in milliseconds 25.0024
```

Please note that the token information displaying the access token was commented out in the program output
//Display a token
//Console.WriteLine ("This is your token: " + authenticationResult.AccessToken);

If a token display is enabled (as it is in the program below), it can be copied and decoded into a readable form with claims, using https://jwt.ms/

Below is a C# version of the application called Program.cs
To obtain the nuget package “Microsoft.IdentityModel.Clients.ActiveDirectory”, use the link below
https://www.nuget.org/api/v2/package/Microsoft.IdentityModel.Clients.ActiveDirectory/4.5.1

```
using Microsoft.IdentityModel.Clients.ActiveDirectory;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AADTest
{
    class Program
    {
        static void Main(string[] args)
        {

            // Examples for the input parameters
            // string serverName = "<serverName>"; // server name i.e. sqlxx.database.windows.net
            // string databaseName = "<databaseName>"; //Database name  i.e. test
            // string clientId = "<appId>"; // application id of the service principal
            // string aadTenantId = "<tenantId>"; //AAD tenant id
            // string clientSecretKey = "secretKey"; // AAD app secret key

            string serverName = "myserver.database.windows.net";
            string databaseName = "test";
            string clientId = "xxxxxx-xxxxx-xxxxx-xxxx-xxxx";
            string aadTenantId = "xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxxxx";
            string clientSecretKey = "xxxxx/xxxxxx/xxxxx";

            string sqlConnectionString = String.Format("Data Source=tcp:{0},1433;Initial Catalog={1};Persist S

            string AadInstance = "https://login.windows.net/{0}";
            string ResourceId = "https://database.windows.net/";

            AuthenticationContext authenticationContext = new AuthenticationContext(string.Format(AadInstance,
            ClientCredential clientCredential = new ClientCredential(clientId, clientSecretKey);

            DateTime startTime = DateTime.Now;
            Console.WriteLine("Time " + String.Format("{0:mm:ss.fff}", startTime));

            AuthenticationResult authenticationResult = authenticationContext.AcquireTokenAsync(ResourceId, cl

            DateTime endTime = DateTime.Now;
            Console.WriteLine("Got token at " + String.Format("{0:mm:ss.fff}", endTime));

            Console.WriteLine("Total time to get token in milliseconds " + (endTime - startTime).TotalMillisec

            using (var conn = new SqlConnection(sqlConnectionString))
            {
                conn.AccessToken = authenticationResult.AccessToken;

                startTime = DateTime.Now;
                Console.WriteLine("Starting to open connection at " + String.Format("{0:mm:ss.fff}", startTime

                //Display a token
                Console.WriteLine("This is your token: " + authenticationResult.AccessToken);

                conn.Open();

                endTime = DateTime.Now;
                Console.WriteLine("Got connection at " + String.Format("{0:mm:ss.fff}", endTime));

                Console.WriteLine("Total time to establish connection in milliseconds " + (endTime - startTime

                startTime = DateTime.Now;
                Console.WriteLine("Starting to run query at " + String.Format("{0:mm:ss.fff}", startTime));

                using (var cmd = new SqlCommand("SELECT 1", conn))
                {
```

```
                    var result = cmd.ExecuteScalar();
                    Console.WriteLine(result.ToString());
                }

                endTime = DateTime.Now;
                Console.WriteLine("Completing running query at " + String.Format("{0:mm:ss.fff}", endTime));
                Console.WriteLine("Total time to execute query in milliseconds " + (endTime - startTime).Total
            }

            Console.ReadKey();
        }
    }
}
```

Public Doc Reference

Service Principal Authentication

Classification

Root cause Tree - Connectivity/AAD Issue/Other AAD User / Service Principal errors

How good have you found this content?