**AAD Service-to-Service Authentication and .NET SDK for Streaming files to ADLS**

In this article we present the steps to create a Service-to-Service authentication in Azure Active Directory. The authentication will then be used by .Net SDK in a Console app to stream Windows’ local files to Azure Data Lake Store. The app is built in Visual Studio 7 with .Net Framework of version 4.6.

1. Create a Data Lake Store

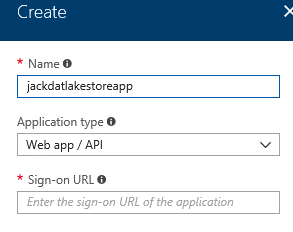
Log into Azure Portal and create a Data Lake Store with name jackdatlakestore

1. Create service-to-service authentication with ADLS using AAD

<https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-service-to-service-authenticate-using-active-directory#create-an-active-directory-application>

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal>

Azure Active Directory > Register Applications > Create



URL: <http://www.cummins.com/jackdatlakestoreapp>

From Properties to get Application ID:

354d9d70-4993-4be6-aef0-8fae7dbf700a

Keys

csharpapp: 9S5JUuKBm4xlL6aSer1paFoGikO3Gn374xz2fzbqBmw=

Directory ID (aka Subscription ID): 72f988bf-86f1-41af-91ab-2d7cd011db47

Then add Required Permissions of Azure Data Lake to this app.

1. Grant this app permission in the subscription

Log into Azure Portal > Subscriptions > Select your subscription > Access Control > Add App with name of jackdatlakestoreapp and Role of READER

1. Grant access to the app in Data Lake

Log into Azure Portal, navigate to the Data Lake Store jackdatlakestore > Access > Add jackdatlakestoreapp READ WRITE EXECUTE current and sub dir.

Create folder mytempdir

1. Create a C# Console App

The following packages are in use:

<?xml version="1.0" encoding="utf-8"?>

<packages>

<package id="Microsoft.Azure.DataLake.Store" version="1.0.4" targetFramework="net461" />

<package id="Microsoft.Azure.Management.DataLake.Store" version="2.3.3-preview" targetFramework="net461" />

<package id="Microsoft.IdentityModel.Clients.ActiveDirectory" version="2.28.3" targetFramework="net461" />

<package id="Microsoft.Rest.ClientRuntime" version="2.3.10" targetFramework="net461" />

<package id="Microsoft.Rest.ClientRuntime.Azure" version="3.3.10" targetFramework="net461" />

<package id="Microsoft.Rest.ClientRuntime.Azure.Authentication" version="2.3.1" targetFramework="net461" />

<package id="Newtonsoft.Json" version="6.0.8" targetFramework="net461" />

<package id="NLog" version="4.4.12" targetFramework="net461" />

</packages>

The Program.cs is the following:

using System;

using System.IO;

using System.Threading;

using Microsoft.Rest;

using Microsoft.Rest.Azure.Authentication;

using Microsoft.Azure.DataLake.Store;

namespace JackDataLakeStoreUploading

{

class Program

{

private static ServiceClientCredentials GetCreds\_SPI\_SecretKey(

string tenant,

Uri tokenAudience,

string clientId,

string secretKey)

{

SynchronizationContext.SetSynchronizationContext(new SynchronizationContext());

var serviceSettings = ActiveDirectoryServiceSettings.Azure;

serviceSettings.TokenAudience = tokenAudience;

var creds = ApplicationTokenProvider.LoginSilentAsync(

tenant,

clientId,

secretKey,

serviceSettings).GetAwaiter().GetResult();

return creds;

}

static void Main(string[] args)

{

try

{

Console.WriteLine("Starting...");

// Service principal / appplication authentication with client secret / key

// Use the client ID of an existing AAD "Web App" application.

string TENANT = "72f988bf-86f1-41af-91ab-2d7cd011db47";

string CLIENTID = "354d9d70-4993-4be6-aef0-8fae7dbf700a";

System.Uri ARM\_TOKEN\_AUDIENCE = new System.Uri(@"https://management.core.windows.net/");

System.Uri ADL\_TOKEN\_AUDIENCE = new System.Uri(@"https://datalake.azure.net/");

string secret\_key = "9S5JUuKBm4xlL6aSer1paFoGikO3Gn374xz2fzbqBmw=";

var armCreds = GetCreds\_SPI\_SecretKey(TENANT, ARM\_TOKEN\_AUDIENCE, CLIENTID, secret\_key);

var adlCreds = GetCreds\_SPI\_SecretKey(TENANT, ADL\_TOKEN\_AUDIENCE, CLIENTID, secret\_key);

var \_adlsAccountName = "jackdatlakestore.azuredatalakestore.net";

var client = AdlsClient.CreateClient(\_adlsAccountName, adlCreds);

// Get file properties

var destPath = "/mytempdir/test.txt";

var directoryEntry = client.GetDirectoryEntry(destPath);

Console.WriteLine(directoryEntry.ToString());

// Create a file - automatically creates any parent directories that don't exist

string fileName = "/mytempdir/test1.txt";

using (var streamWriter = new StreamWriter(client.CreateFile(fileName, IfExists.Overwrite)))

{}

// Append to existing file

var srcPath = @"c:\tmp\test.txt";

using (var streamWriter = new StreamWriter(client.GetAppendStream(fileName)))

{

streamWriter.Write(File.ReadAllText(srcPath));

}

Console.WriteLine("Ending...");

Console.ReadKey();

}

catch (Exception e)

{

Console.WriteLine("{0} Exception caught.", e);

}

}

}

}