Create External Table As Select (CETAS)

Last updated by | Kinye Yu | Mar 3, 2023 at 7:04 AM PST

Contents

Summary

The CETAS functionality allows for the exportation of SELECT query results to an external storage location, specifically an Azure blob storage. This offers users the ability to query data from the SQL MI or other sources without the need for exportation via third-party tools such as SSIS or Data Factory. Additionally, CETAS offers the benefit of data archiving, thereby optimizing disk space utilization.

Prerequisites

To enable the feature on SQL MI, the customer can utilize the following PowerShell script. However, it is important to note that SQLMI will also require write permissions to the designated storage account, specifically via the Azure Blob Storage Contributor role.

To run the PowerShell command below the Azure AD user must have **Contributor** or SQL Security Manager Azure RBAC roles on the **managed instance**.

1. Open CLI with PowerShell.

```
#Run in Azure Cloud Shell

#Enter your Azure subscription ID
$SubscriptionID = "<SubscriptionID>"

#Enter your managed instance name – for example, "sqlmi1"
$ManagedInstanceName = "<managedInstanceName>"


#================================================================================
#INVOKING THE API CALL -- REST OF THE SCRIPT IS NOT USER CONFIGURABLE
#================================================================================
#Log in and select a subscription if needed.
//
if ((Get-AzContext ) -eq $null)
{
    echo "Logging to Azure subscription"
    Login-AzAccount
}
Select-AzSubscription -SubscriptionName $SubscriptionID

#Build the URI for the API call.
//
$miRG = (Get-AzSqlInstance -InstanceName $ManagedInstanceName).ResourceGroupName
$uriFull = "https://management.azure.com/subscriptions/" + $SubscriptionID + "/resourceGroups/" + $miRG + "/pr

//echo $uriFull

#Build the API request body.
//
$properties = @{serverConfigurationOptionValue = 1}

$bodyFull = @{properties = $properties}

//echo $bodyFull

$jsonBody = $bodyFull | ConvertTo-Json

#Get auth token and build the HTTP request header.
//
$azProfile = [Microsoft.Azure.Commands.Common.Authentication.Abstractions.AzureRmProfileProvider]::Instance.Pr
$currentAzureContext = Get-AzContext
$profileClient = New-Object Microsoft.Azure.Commands.ResourceManager.Common.RMProfileClient($azProfile)
$token = $profileClient.AcquireAccessToken($currentAzureContext.Tenant.TenantId)
$authToken = $token.AccessToken
$headers = @{}
$headers.Add("Authorization", "Bearer "+"$authToken")

#Invoke API call. API call creates an Asynchronous operation that will change the configuration setting. In ra
#You can check the status of the Async operation by using the callback URL you will get in the response header
//
Invoke-WebRequest -Method PUT -Headers $headers -Uri $uriFull -ContentType "application/json" -Body $jsonBody


#================================================================================================
#OPTIONAL CODE TO RUN TO GET THE STATUS OF YOUR API CALL AND CONFIRM IF IT SUCCEEDED
#================================================================================================


#If you got the Azure-AsyncOperation URL from response header, use the below to query the operation for status
//
//$AzAsyncOpURL = "https://management.azure.com/subscriptions/c9c0410b-703b-42d0-a5f8-050a93ec2562/resourceGro
//Invoke-WebRequest -Method GET -Headers $headers -Uri $AzAsyncOpURL -ContentType "application/json"

#Alternatively, instead of querying the Azure-AsyncOperation for status, you can query the configuration optio
#Get the configuration status using the call below. This will return JSON with serverConfigurationOptionValue
//
```

```
Invoke-WebRequest -Method GET -Headers $headers -Uri $uriFull -ContentType "application/json"
```

Issue

1. Customer not able to create external table due to External Data Source

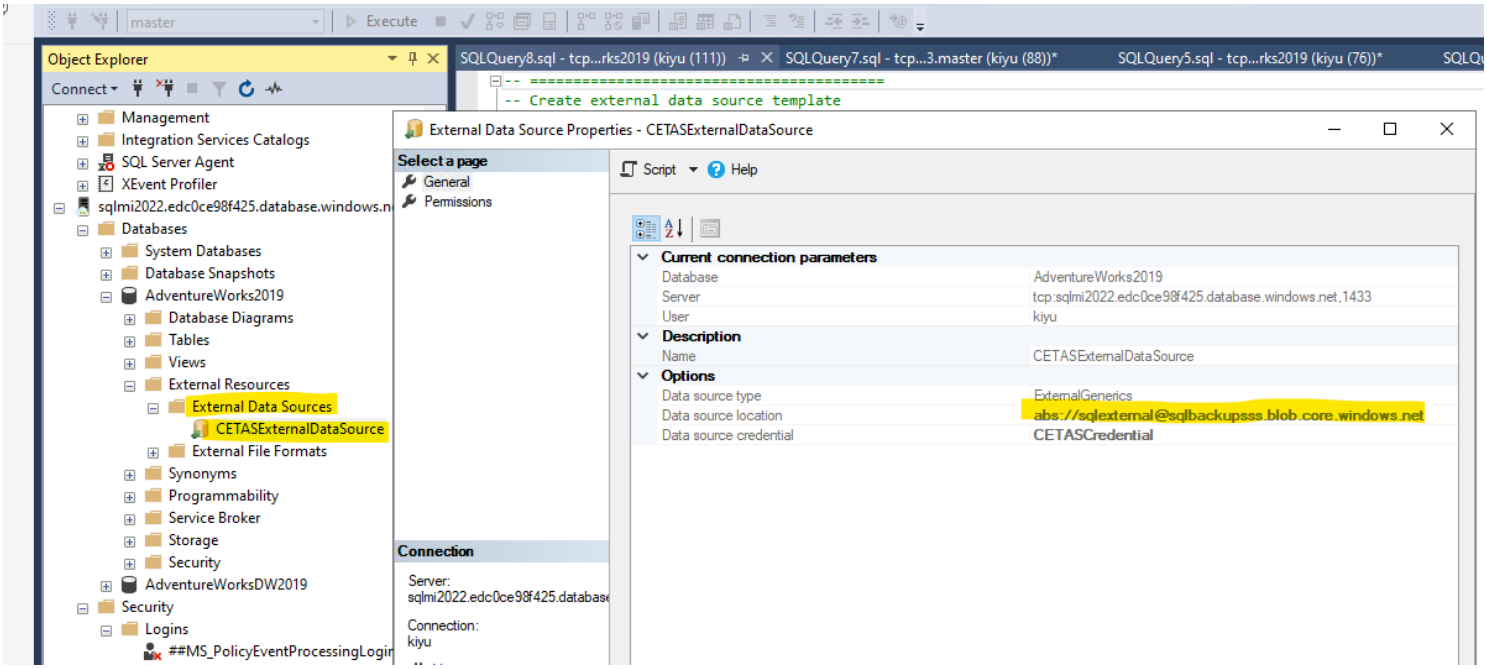
Using Azure Data Lake use URL prefix "adls"

```
CREATE EXTERNAL DATA SOURCE [SQLwriteable] WITH (
    LOCATION = 'adls://<containername>@<mystorageaccount>.dfs.core.windows.net/mybaseoutputfolderpath',
    CREDENTIAL = [WorkspaceIdentity]
);
GO
```

Blob Storage use URL prefix "abs"

```
CREATE EXTERNAL DATA SOURCE [CETASExternalDataSource]
WITH (
    LOCATION = 'abs://<containerName>@<storageaccount>.blob.core.windows.net',
    CREDENTIAL = [CETASCredential]);
GO
```

To confirm external data source:



2. Unfortunately, refreshing data that has already been created in the external storage location is not supported. In such cases, the customer must drop the external table and external storage location and re-run the script to ensure that the data is correctly refreshed.

Mitigation

What to do to resolve the issue. Maybe file an ICM, might have a canned RCA template to be used, etc.

RCA (optional)

For recurring [by design] issues where there is an RCA template for the engineer to customize with timestamps and server details, provide that here.

Product Group Video

Video Link

Public Doc Reference

Public Doc

Internal Reference (optional)

This may include links to architecture, training material, etc that is helpful for understanding the steps given above.

How good have you found this content?