Unable to export Elastic Jobs Database

Last updated by | Charlene Wang | Jan 11, 2023 at 12:29 AM PST

Contents

Issue

When trying to export a database that is or was used as Elastic Jobs database you can encounter errors like:

One or more unsupported elements were found in the schema used as part of a data package.

Error SQL71501: Error validating element **[jobs_internal].[visible_targets_formatted]**: View: **[jobs_internal].[visible_targets_formatted]** contains an unresolved reference to an object. Either the object does not exist or the reference is ambiguous because it could refer to any of the following objects: **[jobs_internal].[database_credentials].[C], [jobs_internal].[database_credentials].[name] or [jobs_internal].[targets].[C]**.

Error SQL71501: Error validating element **[jobs]**: Schema: **[jobs]** has an unresolved reference to object **[##MS_JobAccount##]**.

Error SQL71501: Error validating element [jobs_internal]: Schema: [jobs_internal] has an unresolved reference to object [##MS_JobAccount##].

Investigation/Analysis

When the database is/was used as Elastic Jobs database it cannot be exported directly because there are some security-related objects that are not supported in export at the moment.

Mitigation

The supported format for Azure SQL Database export and import is .bacpac files.

This kind of files can be seen as a table by table export and are not transactionally consistent.

Independently of the chosen method to generate the .bacpac file, you must take into account this lack of transactional consistency.

The way to ensure that the backup file is consistent is to export a database that has no write activity during the export or **create a copy of the database and do the export from that copy**.

When the database is/was used as Elastic Jobs database we need to rely on the second method because there are some security-related objects that are not supported in export at the moment.

In order to successfully export the database, we need to:

1. Create a database copy.
   You can check how to create a copy of the database using the portal at https://docs.microsoft.com/en-us/azure/sql-database/sql-database-copy-portal (there is also documentation for doing this using T-SQL or PowerShell)

2. Remove the Elastic Job related objects from the copy. You can find a script to do it below. **Please make sure you are connected to the copy.**

3. Export the database from the copy.

4. Delete the database copy.

The script you can use for step 2 is the following, **please do not use this script in any other scenario**. The script will generate the drop statements, please review them and run them manually or uncomment the 'exec' statements in the bottom by removing the leading '--'

```
declare @n char(1)
set @n = char(10)

declare @triggers nvarchar(max)
declare @procedures nvarchar(max)
declare @constraints nvarchar(max)
declare @FKs nvarchar(max)
declare @views nvarchar(max)
declare @tables nvarchar(max)
declare @udt nvarchar(max)

-- procedures
select @procedures = isnull( @procedures + @n, '' ) + 'drop procedure [' + schema_name(schema_id) + '].[' + na
from sys.procedures
where schema_name(schema_id) = 'jobs' or schema_name(schema_id) = 'jobs_internal'

-- foreign keys
select @FKs = isnull( @FKs + @n, '' ) + 'alter table [' + schema_name(schema_id) + '].[' + object_name( parent
from sys.foreign_keys
where schema_name(schema_id) = 'jobs' or schema_name(schema_id) = 'jobs_internal'

-- views
select @views = isnull( @views + @n, '' ) + 'drop view [' + schema_name(schema_id) + '].[' + name + ']'
from sys.views
where schema_name(schema_id) = 'jobs' or schema_name(schema_id) = 'jobs_internal'

-- tables
select @tables = isnull( @tables + @n, '' ) + 'drop table [' + schema_name(schema_id) + '].[' + name + ']'
from sys.tables
where schema_name(schema_id) = 'jobs_internal'

-- user defined types
select @udt = isnull( @udt + @n, '' ) + 'drop type [' + schema_name(schema_id) + '].[' + name + ']'
from sys.types
where is_user_defined = 1
and schema_name(schema_id) = 'jobs_internal'
order by system_type_id desc

declare @functions nvarchar(max)

-- functions
select @functions = isnull( @functions + @n, '' ) + 'drop function [' + schema_name(schema_id) + '].[' + name
from sys.objects
where type in ( 'FN', 'IF', 'TF' )
and schema_name(schema_id) = 'jobs_internal'

declare @stmt nvarchar(max)

set @stmt = isnull(@stmt + @n, '') + 'DROP SCHEMA IF EXISTS [jobs]'
set @stmt = isnull(@stmt + @n, '') + 'DROP SCHEMA IF EXISTS [jobs_internal]'

set @stmt = isnull(@stmt + @n, '') + 'DROP USER IF EXISTS [##MS_JobAccount##]'
set @stmt = isnull(@stmt + @n, '') + 'DROP USER IF EXISTS [##MS_JobsResourceManager##]'

set @stmt = isnull(@stmt + @n, '') + 'DROP ROLE IF EXISTS [jobs_admin]'
set @stmt = isnull(@stmt + @n, '') + 'DROP ROLE IF EXISTS [jobs_reader]'
set @stmt = isnull(@stmt + @n, '') + 'DROP ROLE IF EXISTS [jobs_resource_manager]'


print @procedures
print @FKs
print @views
print @tables
print @udt
print 'GO'
print @functions
print 'GO'
print @stmt
```

```
  --exec sp_executesql @procedures
  --exec sp_executesql @FKs
  --exec sp_executesql @views
  --exec sp_executesql @tables
  --exec sp_executesql @udt
  --GO
  --exec sp_executesql @functions
  --GO
  --exec sp_executesql @stmt
  --print 'clean up finished'
```

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
<TBD>

Repair item

https://msdata.visualstudio.com/Database Systems/_workitems/edit/2140694/?triage=true

How good have you found this content?

🙂 🙁