**ELASTIC JOBS PRIVATE PREVIEW PROGRAM**

USER GUIDE V1.5

Debra Dove

03/31/2017

Microsoft Confidential – Shared Under NDA Only

Table of Contents

[Preview audience 6](#_Toc463346303)

[Elastic jobs concepts 6](#_Toc463346304)

[Job account 6](#_Toc463346305)

[Job account database 6](#_Toc463346306)

[Jobs 6](#_Toc463346307)

[Job steps 7](#_Toc463346308)

[Target group 7](#_Toc463346309)

[Target group members 7](#_Toc463346310)

[Target group behavior 7](#_Toc463346311)

[Credentials 8](#_Toc463346312)

[Permissions 8](#_Toc463346313)

[Prerequisites 9](#_Toc463346314)

[Download and install an updated Azure PowerShell package 9](#_Toc463346315)

[Create a job account 9](#_Toc463346316)

[Preview scenarios 10](#_Toc463346317)

[Create a credential for job execution across databases in the target group 10](#_Toc463346318)

[Create a target group including a server(s) 11](#_Toc463346319)

[Create a target group including a server(s) while excluding a single database 11](#_Toc463346320)

[Create a target group including a pool(s) 12](#_Toc463346321)

[Create a job to deploy new schema to all databases in the target group 12](#_Toc463346322)

[Create a job to monitor database performance 13](#_Toc463346323)

[Create a job to monitor pool performance 13](#_Toc463346324)

[View job definitions 14](#_Toc463346325)

[Begin ad-hoc execution of a job 15](#_Toc463346326)

[Enable scheduled execution of a job 15](#_Toc463346327)

[Monitor the status of job executions 15](#_Toc463346328)

[Cancel a job execution 16](#_Toc463346329)

[Delete job history since a point in time 16](#_Toc463346330)

[Delete a job and all its job history 17](#_Toc463346331)

[Job Account API (PowerShell) 17](#_Toc463346332)

[New-AzureRmSqlJobAccount 17](#_Toc463346333)

[Syntax 17](#_Toc463346334)

[Detailed Description 17](#_Toc463346335)

[Parameters 17](#_Toc463346336)

[Semantics 18](#_Toc463346337)

[Outputs 18](#_Toc463346338)

[Example 18](#_Toc463346339)

[Get-AzureRmSqlJobAccount 18](#_Toc463346340)

[Syntax 18](#_Toc463346341)

[Detailed Description 18](#_Toc463346342)

[Parameters 19](#_Toc463346343)

[Outputs 19](#_Toc463346344)

[Example 1: Get all accounts 19](#_Toc463346345)

[Example 2: Get a specific job account 19](#_Toc463346346)

[Remove-AzureRmSqlJobAccount 19](#_Toc463346347)

[Syntax 19](#_Toc463346348)

[Detailed Description 19](#_Toc463346349)

[Parameters 19](#_Toc463346350)

[Outputs 20](#_Toc463346351)

[Example 1: Removes job account 20](#_Toc463346352)

[Job APIs & Views (Transact-SQL) 20](#_Toc463346353)

[Job APIs 21](#_Toc463346354)

[sp\_add\_job 21](#_Toc463346355)

[sp\_update\_job 23](#_Toc463346356)

[sp\_delete\_job 24](#_Toc463346357)

[sp\_add\_jobstep 25](#_Toc463346358)

[sp\_update\_jobstep 28](#_Toc463346359)

[sp\_delete\_jobstep 31](#_Toc463346360)

[sp\_start\_job 32](#_Toc463346361)

[sp\_stop\_job 32](#_Toc463346362)

[sp\_delete\_job 33](#_Toc463346363)

[sp\_delete\_jobstep 34](#_Toc463346364)

[sp\_add\_target\_group 34](#_Toc463346365)

[sp\_delete\_target\_group 35](#_Toc463346366)

[sp\_add\_target\_group\_member 36](#_Toc463346367)

[sp\_delete\_target\_group\_member 38](#_Toc463346368)

[sp\_purge\_jobhistory 39](#_Toc463346369)

[Job Views 41](#_Toc463346370)

[[jobs].[jobs] 41](#_Toc463346371)

[[jobs].[jobs\_executions] 42](#_Toc463346372)

[[jobs].[jobsteps] 43](#_Toc463346373)

[[jobs].[target\_groups] 44](#_Toc463346374)

[[jobs].[target\_groups\_members] 44](#_Toc463346375)

# Preview audience

The private preview user guide is intended for the customers participating in the private preview. The preview duration is limited so the members must commit to allocating the necessary resources for validating the feature and providing feedback to the product team. We want to thank you for your participation.

# Elastic jobs concepts

## Job account

A job account is a container for job resources that is isolated from the resources of other job accounts. Job resources include job definitions, target groups and job execution status and history.  A job account is created within a server in a specific region and jobs are defined within the job account. The jobs and job execution information, scripts, schedules, and execution history, are stored in system-defined schema and tables that are created in the user-hosted database specified when creating the job account. The job account itself has no cost associated with it at this time, only the job account database.

|  |
| --- |
| **Private preview behavior** |
| The Job account is created using PowerShell API. |

## Job account database

The job account database is used by Elastic jobs for defining jobs and tracking the status and history of job execution. The job account database is specified when a job account is created. The database must already exist at the time of job account creation. The job account database must be a stand-alone database of service tier S2 or higher. The job account database is charged at normal rates based on the resources assigned to it by the system which are based primarily on the concurrent job execution rate required across all jobs defined in the job account.  This rate is set by the customer based on several factors: the number of databases that need to be targeted, the frequency with which jobs must be executed and how long execution should take across all databases (a job that must execute on many databases in a short period of time needs more resources). If the user has a scheduled job, the user should ensure the database is a high enough service tier to complete all executions within the scheduled period, independent of query execution time at the database target.

## Jobs

The job executes Transact-SQL (T-SQL) scripts across a group of databases. When a job is submitted to be executed across a group of databases, the job *expands* the into execution instances for each database in the target group where each performs script execution against a single database in the group.

|  |
| --- |
| **Private preview behavior** |
| Jobs can only be created and monitored using the T-SQL API. The Portal, PowerShell and REST APIs for creating and monitoring jobs will be released as part of future updates. |

## Job steps

A job contains job steps. The job step specifies the T-SQL script, target group, credentials, retry semantics and output parameters.

|  |
| --- |
| **Private preview behavior** |
| Jobs can only contain up to one job step. |

## Target group

A target group is a resource containing databases that a job step can execute on, namely the job step executes independently on each database in the target group. A target group can include or exclude databases as specified when creating the target group and declaring the target group members.

### Target group members

There are three types of target group members that you can specify when adding members to a target group:

* **Server:** Specify server(s) to include all databases located within the logical server(s). When a server target group member is specified, you must include the master database credential in order to support refreshing the group of databases prior to job execution.
* **Pool:** Specify Elastic pool(s) to include or exclude all databases located within the Elastic pool(s). When a pool target group member is specified, you must include the master database credential in order to support refreshing the group of databases prior to job execution.
* **Database:** Specify database(s) to include or exclude database(s).

### Target group behavior

A target group is dynamic, namely if a target group member of type server or pool is specified, the expansion of all the databases located in the server and/or pool are evaluated at execution time of the job. Therefore, if a new database is added to the server and/or pool, the subsequent execution of the job would include the newly added database. Likewise, if a database is deleted or moved from the server or pool, the next execution of the job would not execute against that database because it is no longer in the target group. Target group members (aka, target databases) can reside in any server under any subscription in any region (jobs running across regions may experience higher latency).



Diagram 1: Example of job executions against target databases.

## Credentials

Elastic jobs use **database scoped credentials** to connect to the databases specified by the target group upon execution. These **database scoped credentials** are also used to connect to the master database of a server to enumerate all the databases in server and/or an Elastic pool, when either of these are used as the target group member type. You must create the database scoped credential(s) in the job account database that will be used doing job execution. This same credential must also be created on all target databases with sufficient permissions for the job to complete successfully. For more information about database scoped credentials, see <https://msdn.microsoft.com/en-us/library/mt270260.aspx>. Credentials are expected to be reused across jobs. The credential passwords are encrypted and secure from users who have read-only access to Elastic Database jobs objects.

Security best practices for Elastic Database jobs include:

1. Limit usage of the APIs to trusted individuals.
2. Credentials should have the least privileges necessary to perform the job task. More information can be seen within this [Authorization and Permissions](https://msdn.microsoft.com/library/bb669084.aspx) SQL Server MSDN article.

### Create a credential for job execution across databases in the target group

The credential is used by the Elastic jobs to connect to your target databases for script execution. The credential should have appropriate permissions on the databases specified by the target group to successfully execute the script. When using a server and/or pool target group member, it is highly suggested to create server principal credential for use to refresh the database lists of the server(s) and/or pool(s) prior to expansion of the server and/or pool at time of job execution. The database scoped credential is created on the job account database. The same credential must be used to [‘Create a Login’](https://msdn.microsoft.com/en-us/library/aa337562.aspx) and [‘Create a User from Login to grant the Login Database Permissions’](https://msdn.microsoft.com/en-us/library/aa337545.aspx) on the target databases.

--Connect to the Elastic jobs account database specified when creating the job account

-- Create a db master key if one does not already exist, using your own password.

CREATE MASTER KEY ENCRYPTION BY PASSWORD='<EnterStrongPasswordHere>';

-- Create a database scoped credential.

CREATE DATABASE SCOPED CREDENTIAL myjobcred WITH IDENTITY = 'jobcred',

SECRET = '<EnterStrongPasswordHere>';

GO

-- Create a database scoped credential for the master database of server1.

CREATE DATABASE SCOPED CREDENTIAL mymastercred WITH IDENTITY = 'mastercred',

SECRET = '<EnterStrongPasswordHere>';

GO

## Permissions

At the time a job account is created and linked to a job account database, the job schema is created including the following fixed database roles, which give the administrators finer control access.

* jobs\_reader

The role are created with the following permissions within the schemas specified:

|  |  |  |
| --- | --- | --- |
| Role name | ‘jobs’ schema permissions | ‘jobs\_internal’ schema permissions |
| **jobs\_reader** | SELECT | None |

The jobs\_reader role allows the administrator to provide job monitoring only access to a group of individuals is desired.

|  |
| --- |
| **Important** |
| Consider the security implications before granting access to the job account database as a database administrator. A malicious user who has permissions to create or edit jobs could create or edit a job that uses a stored credential to connect to a database under the malicious user's control, which could allow the malicious user to determine the credential’s password. |

## Job History Cleanup

A system cleanup job purges all execution history of >45 days. The sp\_purge\_history stored procedure can be called to remove history less than 45 days old, but the automatic cleanup will always remove history older than 45 days without having to call the stored procedure.

## Results Collection Scenarios

For results collection returning scenarios, it can be useful to include scripting variables to enable grouping on the destination table, for example grouping by job execution id. The supported scriptiong variable are as follows:

• $(job\_execution\_id)

• $(job\_name)

• $(job\_id)

• $(job\_version)

• $(step\_id)

• $(step\_name) – coming soon

The column internal\_execution\_id of type unique identifier is created and added onto every result-set. This is a unique value for every row in destination table and matches the job execution in the job account database,

# Prerequisites

* An Azure subscription. For a free trial, see [Free one-month trial](https://azure.microsoft.com/pricing/free-trial/).
* An Azure SQL Database server and database for hosting the job account.
* Validate your subscription has been registered for the new feature
  1. Open Azure PowerShell
  2. If you have multiple subscriptions you can run the Get-AzureRmSubscription cmdlet and copy the desired subscription information from the result set.
  3. Select-AzureRmSubscription -SubscriptionId {SubscriptionID}
  4. Get-AzureRmProviderFeature -ProviderName Microsoft.Sql -FeatureName sqldb-JobAccounts
     + Returns:

FeatureName ProviderName RegistrationState

sqldb-JobAccounts Microsoft.Sql Registered

# Download and install an updated Azure PowerShell package

We have included the Elastic job cmdlets in an updated version of the Azure PowerShell package. The cmdlets are part of the standard Azure PowerShell API. In order to include the updated cmdlets for this Private Preview, you must download and install the ‘updated’ latest AzurePowerShell.msi located here:

* <https://github.com/jaredmoo/azure-powershell/releases>

This Azure PowerShell release is based on the latest release of Azure PowerShell and the difference is limited to the addition of the Elastic jobs APIs within the SQL Database module.

For more information about the June release and how to install it, see [How to install and configure Azure PowerShell](https://azure.microsoft.com/en-us/documentation/articles/powershell-install-configure/).

# Create a job account

To create a job account, you must already have an Azure SQL Database server and a database created to be used as the job account database. The steps below create a new Azure resource group, a new server and a new database. In step 3, these values are then specified to create the new job account.

1. Create a new server (optional, an existing server can be provided when creating job account):

New-AzureRmSqlServer -ResourceGroupName "resourcegroup1" -Location "server\_data\_center\_location" -ServerName "server1" -ServerVersion 12.0

When prompted, supply an administrator username and password for the new server (not your Azure credentials).

1. Create a new database to use as the job control database (recommended):

New-AzureRmSqlDatabase -ServerName "server1" -ResourceGroupName "resourcegroup1" -DatabaseName "ElasticsJobs" -RequestedServiceObjectiveName S2

It is recommended that the job database only be used for job account. You can specify any name for the job account database.

1. Create a new job account:

New-AzureRmSqlJobAccount -JobAccountName "jobaccount1" -ResourceGroupName "resourcegroup1" -ServerName "server1" -DatabaseName "ElasticsJobs"

*Note: the syntax of the Elastic Jobs cmdlets is not yet published. It is documented in the API section below.*

# Preview scenarios

The goal of the private preview is to validate the core features of Elastic Jobs. The following scenarios are in scope.

## Create a credential for job execution across databases in the target group

The credential is used by the Elastic jobs to connect to your target databases for script execution. The credential should have appropriate permissions on the databases specified by the target group to successfully execute the script. When using a server and/or pool target group member, it is highly suggested to create a master credential for use to refresh the credential prior to expansion of the server and/or pool at time of job execution. The database scoped credential is created on the job account database. The same credential must be used to [‘Create a Login’](https://msdn.microsoft.com/en-us/library/aa337562.aspx) and [‘Create a User from Login to grant the Login Database Permissions’](https://msdn.microsoft.com/en-us/library/aa337545.aspx) on the target databases.

--Connect to the Elastic jobs account database specified when creating the job account

-- Create a db master key if one does not already exist, using your own password.

CREATE MASTER KEY ENCRYPTION BY PASSWORD='<EnterStrongPasswordHere>';

-- Create a database scoped credential.

CREATE DATABASE SCOPED CREDENTIAL myjobcred WITH IDENTITY = 'jobcred',

SECRET = '<EnterStrongPasswordHere>';

GO

-- Create a database scoped credential for the master database of server1.

CREATE DATABASE SCOPED CREDENTIAL mymastercred WITH IDENTITY = 'mastercred',

SECRET = '<EnterStrongPasswordHere>';

GO

## Create a target group including a server(s)

“I want to execute a job against all databases in my server.”

--Connect to the Elastic jobs account database specified when creating the job account

-- Add a target group containing server(s)

EXEC jobs.sp\_add\_target\_group 'ServerGroup1'

-- Add a server target member

EXEC jobs.sp\_add\_target\_group\_member

'ServerGroup1',

@target\_type = 'SqlServer',

@refresh\_credential\_name='mymastercred', --credential required to refresh the databases in server

@server\_name='server1.database.windows.net'

--View the recently created target group and target group members

SELECT \* FROM jobs.target\_groups WHERE target\_group\_name='ServerGroup1';

SELECT \* FROM jobs.target\_group\_members WHERE target\_group\_name='ServerGroup1';

## Create a target group including a server(s) while excluding a single database

“I want to execute a job against all databases in my server(s) except the database named ‘MappingDB’.”

--Connect to the Elastic jobs account database specified when creating the job account

-- Add a target group containing server(s)

EXEC [jobs].sp\_add\_target\_group = N'ServerGroup'

GO

-- Add a server target member

EXEC [jobs].sp\_add\_target\_group\_member

@target\_group\_name = N'ServerGroup',

@target\_type = N'SqlServer',

@refresh\_credential\_name=N'mymastercred', --credential required to refresh the databases in server

@server\_name=N'London.database.windows.net'

GO

-- Add a server target member

EXEC [jobs].sp\_add\_target\_group\_member

@target\_group\_name = N'ServerGroup',

@target\_type = N'SqlServer',

@refresh\_credential\_name=N'mymastercred', --credential required to refresh the databases in server

@server\_name='server2.database.windows.net'

GO

--Excude a database target member from the server target group

EXEC [jobs].sp\_add\_target\_group\_member

@target\_group\_name = N'ServerGroup',

@membership\_type = N'Exclude',

@target\_type = N'SqlDatabase',

@server\_name = N'server1.database.windows.net',

@database\_name =N'MappingDB'

GO

--View the recently created target group and target group members

SELECT \* FROM [jobs].target\_groups WHERE target\_group\_name = N'ServerGroup';

SELECT \* FROM [jobs].target\_group\_members WHERE target\_group\_name = N'ServerGroup';

## Create a target group including a pool(s)

“I have one or more elastic pools and I want to target all the databases in my pools.”

--Connect to the Elastic jobs account database specified when creating the job account

-- Add a target group containing pool(s)

EXEC jobs.sp\_add\_target\_group 'PoolGroup'

-- Add an elastic pool(s) target member

EXEC jobs.sp\_add\_target\_group\_member

'PoolGroup',

@target\_type = 'SqlElasticPool',

@refresh\_credential\_name='mymastercred', --credential required to refresh the databases in server

@server\_name='server1.database.windows.net',

@elastic\_pool\_name='ElasticPool-1'

-- View the recently created target group and target group members

SELECT \* FROM jobs.target\_groups WHERE target\_group\_name = N'PoolGroup';

SELECT \* FROM jobs.target\_group\_members WHERE target\_group\_name = N'PoolGroup';

## Create a job to deploy new schema to all databases in the target group

“I need to deploy new schema to all databases.”

--Connect to the Elastic jobs account database specified when creating the job account

--Add job for create table

EXEC jobs.sp\_add\_job @job\_name='CreateTableTest', @description='Create Table Test'

-- Add job step for create table

EXEC jobs.sp\_add\_jobstep @job\_name='CreateTableTest',

@command=N'IF NOT EXISTS (SELECT \* FROM sys.tables

WHERE object\_id = object\_id(''Test''))

CREATE TABLE [dbo].[Test]([TestId] [int] NOT NULL);',

@credential\_name=’myjobcred’,

@target\_group\_name='PoolGroup'

## Data collection Scenarios

In many data collection scenarios when the execution of the script on the tenant databases is going to return a result set which you specify to be stored in an Azure SQL Database table, it can be useful to include some of these scripting variables to help post-process the results of the job.

• $(job\_execution\_id)

• $(job\_name)

• $(job\_id)

• $(Job\_version)

• $(step\_id)

• $(step\_name)

There is a column,

## Create a job to monitor database performance

--Connect to the Elastic jobs account database specified when creating the job account

-- Add a job to collect perf results

EXEC jobs.sp\_add\_job @job\_name ='ResultsJob', @description='Collection Performance data from all customers'

-- Add a job step w/ schedule to collect results

EXEC jobs.sp\_add\_jobstep

@job\_name='ResultsJob',

@command= N' SELECT DB\_NAME() DatabaseName, $(job\_execution\_id) AS job\_execution\_id, \* FROM sys.dm\_db\_resource\_stats WHERE end\_time > DATEADD(mi, -20, GETDATE());',

@credential\_name='myjobcred',

@target\_group\_name='PoolGroup',

@output\_type='SqlDatabase',

@output\_credential\_name=’myjobcred’,

@output\_server\_name=’server1.database.windows.net',

@output\_database\_name=’<resultsdb>',

@output\_table\_name='<resutlstable>'

## Create a job to monitor pool performance

--Connect to the Elastic jobs account database specified when creating the job account

-- Add a target group containing master database

EXEC jobs.sp\_add\_target\_group 'MasterGroup'

-- Add a server target member

EXEC jobs.sp\_add\_target\_group\_member

@target\_group\_name='MasterGroup',

@target\_type='SqlDatabase',

@server\_name='server1.database.windows.net',

@database\_name='master'

-- Add a job to collect perf results

EXEC jobs.sp\_add\_job

@job\_name='ResultsPoolsJob',

@description='Demo: Collection Performance data from all pools',

@schedule\_interval\_type='Minutes',

@schedule\_interval\_count=15

-- Add a job step w/ schedule to collect results

EXEC jobs.sp\_add\_jobstep

@job\_name='ResultsPoolsJob',

@command=N'declare @now datetime

declare @startTime datetime

declare @endTime datetime

declare @poolLagMinutes datetime

declare @poolStartTime datetime

declare @poolEndTime datetime

SELECT @now = getutcdate ()

SELECT @startTime = dateadd(minute, -15, @now)

SELECT @endTime = @now

SELECT @poolStartTime = dateadd(minute, -30, @startTime)

SELECT @poolEndTime = dateadd(minute, -30, @endTime)

SELECT elastic\_pool\_name , end\_time, elastic\_pool\_dtu\_limit, avg\_cpu\_percent, avg\_data\_io\_percent, avg\_log\_write\_percent, max\_worker\_percent, max\_session\_percent,

avg\_storage\_percent, elastic\_pool\_storage\_limit\_mb FROM sys.elastic\_pool\_resource\_stats

WHERE end\_time > @poolStartTime and end\_time <= @poolEndTime;

'),

@credential\_name='myjobcred',

@target\_group\_name='MasterGroup',

@output\_type='SqlDatabase',

@output\_credential\_name='myjobcred',

@output\_server\_name=’server1.database.windows.net',

@output\_database\_name=’resultsdb',

@output\_table\_name='resutlstable'

## View job definitions

--Connect to the Elastic jobs account database specified when creating the job account

-- View all jobs

select \* from jobs.jobs

-- View the steps of the current version of all jobs

select js.\* from jobs.jobsteps js

join jobs.jobs j

on j.job\_id = js.job\_id

and j.current\_job\_version\_number = js.job\_version\_number

-- View the steps of all versions of all jobs

select \* from jobs.jobsteps

## Begin ad-hoc execution of a job

--Connect to the Elastic jobs account database specified when creating the job account

-- Execute the latest version of a job

EXEC jobs.sp\_start\_job 'CreateTableTest'

-- Execute the latest version of a job and receive the execution id

declare @je uniqueidentifier

exec jobs.sp\_start\_job 'CreateTableTest', @job\_execution\_id = @je output

select @je

select \* from jobs.job\_executions where job\_execution\_id = @je

-- Execute a specific version of a job (e.g. version 1)

exec jobs.sp\_start\_job 'CreateTableTest', 1

## Enable scheduled execution of a job

--Connect to the Elastic jobs account database specified when creating the job account

EXEC jobs.sp\_update\_job

@job\_name='ResultsJob',

@enabled=1,

@schedule\_interval\_type='Minutes',

@schedule\_interval\_count=15

## Monitor the status of job executions

--Connect to the Elastic jobs account database specified when creating the job account

--View top-level execution status for the job named ‘ResultsPoolJob’

SELECT \* FROM jobs.job\_executions

WHERE job\_name = 'ResultsPoolsJob' and step\_id IS NULL

ORDER BY start\_time DESC

--View all top-level execution status for all jobs

SELECT \* FROM jobs.job\_executions WHERE step\_id IS NULL

ORDER BY start\_time DESC

--View all execution statuses for job named ‘ResultsPoolsJob’

SELECT \* FROM jobs.job\_executions

WHERE job\_name = 'ResultsPoolsJob'

ORDER BY start\_time DESC

-- View all active executions

SELECT \* FROM jobs.job\_executions

WHERE is\_active = 1

ORDER BY start\_time DESC

## Cancel a job execution

--Connect to the Elastic jobs account database specified when creating the job account

-- View all active executions to determine job execution id

SELECT \* FROM jobs.job\_executions

WHERE is\_active = 1 AND job\_name = 'ResultPoolsJob'

ORDER BY start\_time DESC

GO

-- Cancel job execution with the specified job execution id

EXEC jobs.sp\_stop\_job '01234567-89ab-cdef-0123-456789abcdef'

## Delete job history since a point in time

--Connect to the Elastic jobs account database specified when creating the job account

-- Delete history of a specific job’s executions older than the specified date

EXEC jobs.sp\_purge\_jobhistory @job\_name='ResultPoolsJob', @oldest\_date='2016-07-01 00:00:00'

--Note: job history is automatically deleted if it is >45 days old

## Delete a job and all its job history

--Connect to the Elastic jobs account database specified when creating the job account

EXEC jobs.sp\_delete\_job @job\_name='ResultsPoolsJob'

--Note: job history is automatically deleted if it is >45 days old

For details of the individual cmdlet syntax refer to [MSDN documentation.](https://msdn.microsoft.com/en-us/library/mt574084.aspx) Note, the syntax of the job cmdlets and T-SQL APIs are not yet published. It is documented in the API section below.

# Job Account API (PowerShell)

The private preview is provided to the select customers for testing the feature using PowerShell. It is our goal to maintain the API compatibility after the feature goes to the public preview. However, it is possible that your feedback will necessitate the API changes so the full compatibility cannot be guaranteed. The following cmdlets are included in the preview.

## New-AzureRmSqlJobAccount

Creates a new Elastic job account.

### Syntax

|  |  |
| --- | --- |
| **Cmdlet** | New- AzureRmSqlJobAccount |
| **Parameters** | [-JobAccountName <String>]  [-ResourceGroupName <String>]  [-ServerName <String>]  [-DatabaseName <String>]  [<CommonParameters>] |

### 

### Detailed Description

The **New-AzureRmSqlJobAccount** cmdlet creates an Elastic jobs account contained within a server and referenced to a database to serve as the job database. Only one job account can be created per region per subscription.

An Elastic job account is a container for jobs resources that is isolated from the resources of other job accounts. Elastic job resources include jobs and target groups.

### Parameters

#### -JobAccountName <String>

Specifies the name of the job account.

#### -ResourceGroupName <String>

Specifies the name of the resource group hosting the logical server of the job account.

*-ServerName <String>*

Specifies the name of the logical server for the job account to be create.

*-DatabaseName <String>*

Name of the database of the elastic database jobs database.

<CommonParameters>

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about\_CommonParameters](http://go.microsoft.com/fwlink/p/?LinkID=113216).

### Semantics

This command will create a new job account.

Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

### Outputs

The output type is the type of the objects that the cmdlet emits.

### Example

The following examples creates a new job account “jobaccount1” in the server “server1”. The server and hence the job account are members of the resource group “resourcegroup1”. The database “jobdb1” must be pre-created and cannot be referenced by another job account.

|  |
| --- |
| New-AzureRmSqlJobAccount -JobAccountName "jobaccount1" -ResourceGroupName "resourcegroup1" -ServerName "server1" -DatabaseName "jobdb1" |

## Get-AzureRmSqlJobAccount

Gets Elastic job accounts in the resource group.

### Syntax

|  |  |
| --- | --- |
| **Cmdlet** | Get- AzureRmSqlJobAccount |
| **Parameters** | [-JobAccountName <String>]  [-ResourceGroupName <String>]  [-ServerName <String>]  [-DatabaseName <String>]  [<CommonParameters>] |

### 

### Detailed Description

The **Get-AzureRmSqlJobAccount** cmdlet creates an Elastic jobs account contained within a server and referenced to a database to serve as the job database. Only one job account can be created per region per subscription.

An Elastic job account is a container for jobs resources that is isolated from the resources of other job accounts. Elastic job resources include jobs and target groups.

### Parameters

#### -JobAccountName <String>

Specifies the name of the job account that this cmdlet gets.

#### -ResourceGroupName <String>

Specifies the name of a resource group in which this cmdlet gets job accounts.

*-ServerName <String>*

Specifies the name of the logical server in which this cmdlet get job accounts.

#### <CommonParameters>

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about\_CommonParameters](http://go.microsoft.com/fwlink/p/?LinkID=113216).

Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

### Outputs

The output type is the type of the objects that the cmdlet emits.

### Example 1: Get all accounts

This command gets all job accounts in the resource group named myrg.

|  |
| --- |
| Get-AzureRMSqlJobAccount –ResourceGroupName "resourcegroup1" |

### Example 2: Get a specific job account

This command gets the job account named “myjobaccount” in the resource group named “myrg” in the server “myserver”.

|  |
| --- |
| Get-AzureRMSqlJobAccount -JobAccountName "jobaccount1" –ResourceGroupName "resourcegroup1" -ServerName "server1" |

## Remove-AzureRmSqlJobAccount

Removes the Elastic job accounts in a server.

### Syntax

|  |  |
| --- | --- |
| **Cmdlet** | Get- AzureRmSqlJobAccount |
| **Parameters** | [-JobAccountName <String>]  [-ResourceGroupName <String>]  [-ServerName <String>]  [<CommonParameters>] |

### Detailed Description

The **Remove-AzureRmSqlJobAccount** cmdlet removes an Elastic jobs account from a server in a resource group.

### Parameters

#### -Force

Forces the command to run without asking for user confirmation.

#### -JobAccountName <String>

Specifies the name of the job account that this cmdlet gets.

#### -ResourceGroupName <String>

Specifies the name of a resource group from which this cmdlet removes a job account.

*-ServerName <String>*

Specifies the name of the logical server in which this cmdlet removes a job account.

#### <CommonParameters>

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see [about\_CommonParameters](http://go.microsoft.com/fwlink/p/?LinkID=113216).

Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

### Outputs

The output type is the type of the objects that the cmdlet emits.

### Example 1: Removes job account

This command removes a job account named "jobaccount1" in the resource group named "resourcegroup1" in the server named "server1".

|  |
| --- |
| Remove-AzureRMSqlJobAccount –JobAccountName "jobaccount1" –ResourceGroupName  "resourcegroup1" -ServerName "server1" |

# Job APIs & Views (Transact-SQL)

Azure SQL Database supports the following stored procedures and views that are used by the job account to manage and monitor scheduled activities. It is our goal to maintain the API compatibility after the feature goes to the public preview. However, it is possible that your feedback will necessitate the API changes so the full compatibility cannot be guaranteed. The following stored procedures and views are created when the job account id created in the server and database specified.

The following table lists out all the object types of **Elastic jobs** along with its description and relevant stored procedures APIs and Views.

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Type** | **Description** | **Views** | **Stored Procedures** |
| Job | The job executes Transact-SQL (T-SQL) scripts across a group of databases. For more information, see jobs. | jobs.jobs | sp\_add\_job  sp\_update\_job  sp\_delete\_job |
| Job step | A job contains job steps. The job step specifies the T-SQL script, target group, credentials, retry semantics and output parameters. | jobs.jobsteps | sp\_add\_jobstep  sp\_update\_jobstep  sp\_delete\_jobstep |
| Target groups | A target group is a resource containing databases that a job step can execute on, namely the job step executes independently on each database in the target group. for more information, see target groups. | jobs.target\_groups  jobs.target\_group\_members | sp\_add\_target\_group  sp\_add\_target\_group\_member  sp\_remove\_target\_group\_member |
| Job executions | The instance of a job executing against every database defined within the target group. Each execution will contain the history or status of it’s execution. | jobs.job\_executions | sp\_start\_job  sp\_stop\_job  sp\_purge\_jobhistory |

## Job APIs

### sp\_add\_job

Adds a new job executed by the job account.

#### Syntax

[jobs]**.sp\_add\_job** [ @job\_name = ] 'job\_name'

[ , [ @description = ] 'description' ]

[ , [ @enabled = ] enabled ]

[ , [ @schedule\_interval\_type = ] schedule\_interval\_type ]

[ , [ @schedule\_interval\_count = ] schedule\_interval\_count ]

[ , [ @schedule\_start\_time = ] schedule\_start\_time ]

[ , [ @schedule\_end\_time = ] schedule\_end\_time ]

[ , [ @job\_id = ] job\_id OUTPUT ]

#### Arguments

[ **@job\_name =** ] **'**job\_name**'**  
The name of the job. The name must be unique and cannot contain the percent (**%**) character. job\_name is **nvarchar(128)**, with no default.

[ **@description =** ] **'**description**'**  
The description of the job. description is **nvarchar(512)**, with a default of NULL. If description is omitted, an empty string is used.

[ **@enabled =** ] enabled  
Whether the job’s schedule is enabled. Enabled is **bit**, with a default of 0 (disabled). If **0**, the job is not enabled and does not run according to its schedule; however, it can be run manually. If **1**, the job will run according to its schedule, and can also be run manually.

[ **@schedule\_interval\_type=** ] *schedule\_interval\_type*  
Value indicates when the job is to be executed. *schedule\_interval\_type* is **nvarchar(50)**, with a default of **Once**, and can be one of the following values:

* 'Once',
* 'Minutes',
* 'Hours',
* 'Days',
* 'Weeks',
* 'Months'

[ **@schedule\_interval\_count**= ] schedule\_interval\_count  
Number of schedule\_interval\_count periods to occur between each execution of the job. schedule\_interval\_count is int, with a default of 1. The value must be greater than or equal to 1.

[ **@schedule\_start\_time**= ] schedule\_start\_time  
Date on which job execution can begin. schedule\_start\_time is DATETIME2, with the default of 0001-01-01 00:00:00.0000000.

[ **@schedule\_end\_time**= ] schedule\_end\_time  
Date on which job execution can stop. schedule\_end\_time is DATETIME2, with the default of 9999-12-31 11:59:59.0000000.

[ **@job\_id =** ] *job\_id* **OUTPUT**  
The job identification number assigned to the job if created successfully. *job\_id* is an output variable of type **uniqueidentifier**.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

sp\_add\_job must be run from the job account database specified when creating the job account.

After sp\_add\_job has been executed to add a job, sp\_add\_jobstep can be used to add steps that perform the activities for the job. The job’s initial version number is 0, which will be incremented to 1 when the first step is added.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_update\_job

Changes the attributes of a job. within a job account.

#### Syntax

[jobs]**.sp\_update\_job** [ @job\_name = ] 'job\_name'

[ @new\_name = ] 'new\_name'

[ , [ @description = ] 'description' ]

[ , [ @enabled = ] enabled ]

[ , [ @schedule\_interval\_type = ] schedule\_interval\_type ]

[ , [ @schedule\_interval\_count = ] schedule\_interval\_count ]

[ , [ @schedule\_start\_time = ] schedule\_start\_time ]

[ , [ @schedule\_end\_time = ] schedule\_end\_time ]

[ , [ @job\_id = ] job\_id OUTPUT ]

#### Arguments

[ **@job\_name =** ] **'***job\_name***'**  
The name of the job to be updated. *job\_name* is **nvarchar(128)**.

[ **@new\_name =** ] **'***new\_name***'**  
The new name of the job to be updated. *new\_name* is **nvarchar(128)**.

[ **@description =** ] **'***description***'**  
The description of the job. *description* is **nvarchar(512)**.

[ **@enabled =** ] *enabled*  
Specifies whether the job’s schedule is enabled **(1)** or not enabled **(0)**. *Enabled* is **bit**.

[ **@schedule\_interval\_type=** ] *schedule\_interval\_type*  
Value indicates when the job is to be executed. *schedule\_interval\_type* is **nvarchar(50)** and can be one of the following values:

* 'Once',
* 'Minutes',
* 'Hours',
* 'Days',
* 'Weeks',
* 'Months'

[ **@schedule\_interval\_count**= ] schedule\_interval\_count  
Number of schedule\_interval\_count periods to occur between each execution of the job. schedule\_interval\_count is int, with a default of 1. The value must be greater than or equal to 1.

[ **@schedule\_start\_time**= ] schedule\_start\_time  
Date on which job execution can begin. schedule\_start\_time is DATETIME2, with the default of 0001-01-01 00:00:00.0000000.

[ **@schedule\_end\_time**= ] schedule\_end\_time  
Date on which job execution can stop. schedule\_end\_time is DATETIME2, with the default of 9999-12-31 11:59:59.0000000.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

After sp\_add\_job has been executed to add a job, sp\_add\_jobstep can be used to add steps that perform the activities for the job. The job’s initial version number is 0, which will be incremented to 1 when the first step is added.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_delete\_job

Changes the attributes of a job. within a job account.

#### Syntax

**[jobs].sp\_update\_job** [ @job\_name = ] 'job\_name'

#### Arguments

[ **@job\_name =** ] **'***job\_name***'**  
The name of the job to be deleted. *job\_name* is **nvarchar(128)**.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

Job history is automatically deleted when a job is deleted.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_add\_jobstep

Adds a step (operation) to a job.

#### Syntax

[jobs]**.sp\_add\_jobstep** [ @job\_name = ] 'job\_name'

[ , [ @step\_id = ] step\_id ]

[ , [ @command\_type = ] 'command\_type' ]

[ , [ @command\_source = ] 'command\_source' ]

, [ @command = ] 'command'

, [ @credential\_name = ] 'credential\_name'

, [ @target\_group\_name = ] 'target\_group\_name'

[ , [ @initial\_retry\_interval\_seconds = ] initial\_retry\_interval\_seconds ]

[ , [ @maximum\_retry\_interval\_seconds = ] maximum\_retry\_interval\_seconds ]

[ , [ @retry\_interval\_backoff\_multiplier = ] retry\_interval\_backoff\_multiplier ]

[ , [ @retry\_attempts = ] retry\_attempts ]

[ , [ @step\_timeout\_seconds = ] step\_timeout\_seconds ]

[ , [ @output\_type = ] 'output\_type' ]

[ , [ @output\_credential\_name = ] 'output\_credential\_name' ]

[ , [ @output\_server\_name = ] 'output\_server\_name' ]

[ , [ @output\_database\_name = ] 'output\_database\_name' ]

[ , [ @output\_schema\_name = ] 'output\_schema\_name' ]

[ , [ @output\_table\_name = ] 'output\_table\_name' ]

#### Arguments

[ **@job\_name =** ] **'**job\_name**'**  
The name of the job to which to add the step. *job\_name* is **nvarchar(128)**.

[ **@step\_id** = ] *step\_id*

The sequence identification number for the job step. s*tep\_id* is an **int**, with a default of **1**. Step identification numbers start at 1 and increment without gaps. If not specified, the step\_id will be automatically assigned to the last in the sequence of steps, for example if the job currently contains no steps, then this new step will be step 1; if the job currently contains step 1, then this new step will be step 2.

|  |
| --- |
| **Private preview behavior** |
| Only zero or one step per job is supported at this time. |

[ **@command\_type** = ] *'command\_type'*  
The type of command that is executed by this jobstep. *command\_type* is **nvarchar(50)**, with a default value of **TSql**, meaning that the value of the **@command\_type** parameter is a T-SQL script.

|  |
| --- |
| **Private preview behavior** |
| If specified, the value must be **TSql**. |

[ **@command\_source** = ] *'command\_source'*  
The type of location where the command is stored. *command\_source* is **nvarchar(50)**, with a default value of **Inline**, meaning that the value of the **@command\_source** parameter is the literal text of the command.

|  |
| --- |
| **Private preview behavior** |
| If specified, the value must be **Inline**. |

[ **@command** = ] *'command'*  
The command must be valid T-SQL script and is then executed by this job step. *command* is **nvarchar(max)**, with a default of **NULL**.

[ **@credential\_name** = ] *‘credential\_name’*

The name of the database scoped credential stored in this job control database that is used to connect to each of the target databases within the target group when this step is executed. *credential\_name* is **nvarchar(128)**.

[ **@target\_group\_name** = ] *‘target-group\_name'*  
The name of the target group that contains the target databases that the job step will be executed on. *target\_group\_name* is **nvarchar(128)**.

[ **@initial\_retry\_interval\_seconds** = ] *initial\_retry\_interval\_seconds*

The delay before the first retry attempt, if the job step fails on the initial execution attempt. *initial\_retry\_interval\_seconds* is **int**, with default value of **1**.

[ **@maximum\_retry\_interval\_seconds** = ] *maximum\_retry\_interval\_seconds*

The maximum delay between retry attempts. If the delay between retries would grow larger than this value, it is capped to this value instead. *maximum\_retry\_interval\_seconds* is **int**, with default value of **120**.

[ **@retry\_interval\_backoff\_multiplier** = ] *retry\_interval\_backoff\_multiplier*

The multiplier to apply to the retry delay if multiple job step execution attempts fail. For example, if the first retry had a delay of 5 second and the backoff multiplier is 2.0, then the second retry will have a delay of 10 seconds and the third retry will have a delay of 20 seconds. *retry\_interval\_backoff\_multiplier* is **real**, with default value of **2.0**.

[ **@retry\_attempts** = ] *retry\_attempts*

The number of times to retry execution if the initial attempt fails. For example, if the retry\_attempts value is 10, then there will be 1 initial attempt and 10 retry attempts, giving a total of 11 attempts. If the final retry attempt fails, then the job execution will terminate with a lifecycle of **Failed**. *retry\_attempts* is **int,** with default value of **10**.

**[ @step\_timeout\_seconds = ]** *step\_timeout\_seconds*

The maximum amount of time allowed for the step to execute. If this time is exceeded, then the job execution will terminate with a lifecycle of **TimedOut**. *step\_timeout\_seconds* is **int,** with default value of **43,200 seconds (12 hours)**.

**[ @output\_type = ]** '*output\_type*'

If not null, the type of destination that the command’s first result set is written to. *output\_type* is **nvarchar(50)**, with a default of **NULL**.

|  |
| --- |
| **Private preview behavior** |
| If specified, the value must be **SqlDatabase**. |

**[ @output\_credential\_name = ]** '*output\_credential\_name*'

If not null, the name of the database scoped credential that is used to connect to the output destination database. Must be specified if *output\_type* equals **SqlDatabase**. *output\_credential\_name* is **nvarchar(128)**, with a default value of **NULL**.

**[ @output\_server\_name = ]** '*output\_server\_name*'

If not null, the fully qualified DNS name of the server that contains the output destination database. Must be specified if *output\_type* equals **SqlDatabase**. *output\_server\_name* is **nvarchar(256)**, with a default of **NULL**.

**[ @output\_database\_name = ]** '*output\_database\_name'*

If not null, the name of the database that contains the output destination table. Must be specified if *output\_type* equals **SqlDatabase**. *output\_database\_name* is **nvarchar(128)**, with a default of **NULL**.

**[ @output\_schema\_name = ]** '*output\_schema\_name*'

If not null, the name of the SQL schema that contains the output destination table. If *output\_type* equals **SqlDatabase**, the default value is **dbo**. *output\_schema\_name* is **nvarchar(128)**.

**[ @output\_table\_name = ]** *'output\_table\_name*'

If not null, the name of the table that the command’s first result set will be written to. Must be specified if *output\_type* equals **SqlDatabase**. *output\_table\_name* is **nvarchar(128)**, with a default value of **NULL**. The table in the destination database indicated through the parameter *output\_database\_name* will be created based on the schema of the returning result-set if it doesn’t already exist.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

When **sp\_add\_jobstep** succeeds, the job’s current version number is incremented. The next time the job is executed, the new version will be used. If the job is currently executing, that execution will not contain the new step.

|  |
| --- |
| **Private preview behavior** |
| Only one step per job is supported at this time. |

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_update\_jobstep

Changes the setting for a step in a job that is used to perform automated activities.

#### Syntax

[jobs].**sp\_update\_jobstep** [ @job\_name = ] 'job\_name'

[ , [ @step\_id = ] step\_id ]

[ , [ @command\_type = ] 'command\_type' ]

[ , [ @command\_source = ] 'command\_source' ]

, [ @command = ] 'command'

, [ @credential\_name = ] 'credential\_name'

, [ @target\_group\_name = ] 'target\_group\_name'

[ , [ @initial\_retry\_interval\_seconds = ] initial\_retry\_interval\_seconds ]

[ , [ @maximum\_retry\_interval\_seconds = ] maximum\_retry\_interval\_seconds ]

[ , [ @retry\_interval\_backoff\_multiplier = ] retry\_interval\_backoff\_multiplier ]

[ , [ @retry\_attempts = ] retry\_attempts ]

[ , [ @step\_timeout\_seconds = ] step\_timeout\_seconds ]

[ , [ @output\_type = ] 'output\_type' ]

[ , [ @output\_credential\_name = ] 'output\_credential\_name' ]

[ , [ @output\_server\_name = ] 'output\_server\_name' ]

[ , [ @output\_database\_name = ] 'output\_database\_name' ]

[ , [ @output\_schema\_name = ] 'output\_schema\_name' ]

[ , [ @output\_table\_name = ] 'output\_table\_name' ]

#### Arguments

[ **@job\_name =** ] **'***job\_name***'**  
The name of the job to which the step belongs. *job\_name* is **nvarchar(128)**.

[ **@step\_id** = ] *step\_id*

The identification number for the job step to be modified. This number cannot be changed. s*tep\_id* is an **int**, with a default of **1**.

|  |
| --- |
| **Private preview behavior** |
| Only zero or one step per job is supported at this time. |

[ **@command\_type** = ] *'command\_type'*  
The type of command that is executed by this jobstep. *command\_type* is **nvarchar(50)**, with a default value of **TSql**, meaning that the value of the **@command\_type** parameter is a T-SQL script.

|  |
| --- |
| **Private preview behavior** |
| If specified, the value must be **TSql**. |

[ **@command\_source** = ] *'command\_source'*  
The type of location where the command is stored. *command\_source* is **nvarchar(50)**, with a default value of **Inline**, meaning that the value of the **@command\_source** parameter is the literal text of the command.

|  |
| --- |
| **Private preview behavior** |
| If specified, the value must be **Inline**. |

[ **@command** = ] *'command'*  
The command(s) must be valid T-SQL script and is then executed by this job step. *command* is **nvarchar(max)**, with a default of **NULL**.

[ **@credential\_name** = ] *‘credential\_name’*

The name of the database scoped credential stored in this job control database that is used to connect to each of the target databases within the target group when this step is executed. *credential\_name* is **nvarchar(128)**.

[ **@target\_group\_name** = ] *‘target-group\_name'*  
The name of the target group that contains the target databases that the job step will be executed on. *target\_group\_name* is **nvarchar(128)**.

[ **@initial\_retry\_interval\_seconds** = ] *initial\_retry\_interval\_seconds*

The delay before the first retry attempt, if the job step fails on the initial execution attempt. *initial\_retry\_interval\_seconds* is **int**, with default value of **1**.

[ **@maximum\_retry\_interval\_seconds** = ] *maximum\_retry\_interval\_seconds*

The maximum delay between retry attempts. If the delay between retries would grow larger than this value, it is capped to this value instead. *maximum\_retry\_interval\_seconds* is **int**, with default value of **120**.

[ **@retry\_interval\_backoff\_multiplier** = ] *retry\_interval\_backoff\_multiplier*

The multiplier to apply to the retry delay if multiple job step execution attempts fail. For example, if the first retry had a delay of 5 second and the backoff multiplier is 2.0, then the second retry will have a delay of 10 seconds and the third retry will have a delay of 20 seconds. *retry\_interval\_backoff\_multiplier* is **real**, with default value of **2.0**.

[ **@retry\_attempts** = ] *retry\_attempts*

The number of times to retry execution if the initial attempt fails. For example, if the retry\_attempts value is 10, then there will be 1 initial attempt and 10 retry attempts, giving a total of 11 attempts. If the final retry attempt fails, then the job execution will terminate with a lifecycle of **Failed**. *retry\_attempts* is **int,** with default value of **10**.

**[ @step\_timeout\_seconds = ]** *step\_timeout\_seconds*

The maximum amount of time allowed for the step to execute. If this time is exceeded, then the job execution will terminate with a lifecycle of **TimedOut**. *step\_timeout\_seconds* is **int,** with default value of **43,200 seconds (12 hours)**.

**[ @output\_type = ]** '*output\_type*'

If not null, the type of destination that the command’s first result set is written to. *output\_type* is **nvarchar(50)**, with a default of **NULL**.

|  |
| --- |
| **Private preview behavior** |
| If specified, the value must be **SqlDatabase**. |

**[ @output\_credential\_name = ]** '*output\_credential\_name*'

If not null, the name of the database scoped credential that is used to connect to the output destination database. Must be specified if *output\_type* equals **SqlDatabase**. *output\_credential\_name* is **nvarchar(128)**, with a default value of **NULL**.

**[ @output\_server\_name = ]** '*output\_server\_name*'

If not null, the fully qualified DNS name of the server that contains the output destination database. Must be specified if *output\_type* equals **SqlDatabase**. *output\_server\_name* is **nvarchar(256)**, with a default of **NULL**.

**[ @output\_database\_name = ]** '*output\_database\_name'*

If not null, the name of the database that contains the output destination table. Must be specified if *output\_type* equals **SqlDatabase**. *output\_database\_name* is **nvarchar(128)**, with a default of **NULL**.

**[ @output\_schema\_name = ]** '*output\_schema\_name*'

If not null, the name of the SQL schema that contains the output destination table. If *output\_type* equals **SqlDatabase**, the default value is **dbo**. *output\_schema\_name* is **nvarchar(128)**.

**[ @output\_table\_name = ]** *'output\_table\_name*'

If not null, the name of the table that the command’s first result set will be written to. Must be specified if *output\_type* equals **SqlDatabase**. *output\_table\_name* is **nvarchar(128)**, with a default value of **NULL**. The table in the destination database indicated through the parameter *output\_database\_name* will be created based on the schema of the returning result-set if it doesn’t already exist.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

When **sp\_update\_jobstep** succeeds, the job’s current version number is incremented. The next time the job is executed, the new version will be used. If the job is currently executing, that execution will not contain the new step.

|  |
| --- |
| **Private preview behavior** |
| Only one step per job is supported at this time. |

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_delete\_jobstep

Removes a job step from a job.

#### Syntax

[jobs].**sp\_delete\_jobstep** [ @job\_name = ] 'job\_name'

[ , [ @step\_id = ] step\_id ]

#### Arguments

[ **@job\_name =** ] **'***job\_name***'**  
The name of the job from which the step will be removed. *job\_name* is **nvarchar(128)**, with no default.

[ **@step\_id** = ] *step\_id*

The identification number for the job step to be modified. This number cannot be changed. s*tep\_id* is an **int**, with no default.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

Since only one job step is supported at this time, **sp\_delete\_jobstep** succeeds the job has no steps.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_start\_job

Instructs the Elastic job account to execute the job immediately.

#### Syntax

[jobs].**sp\_start\_job** [ @job\_name = ] 'job\_name'

[ , [ @job\_version\_number = ] job\_version\_number ]

#### Arguments

[ **@job\_name =** ] **'***job\_name***'**  
The name of the job from which the step will be removed. *job\_name* is **nvarchar(128)**, with no default.

[ **@** **job\_version\_number** = ] *job\_version\_number*

The version of the job to be executed. *job\_version\_number* is an **int**, defaults to the latest version of the job.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

None.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_stop\_job

Instructs Elastic job account to stop the execution of a job.

#### Syntax

[jobs]**.sp\_stop\_job** [ @ job\_execution\_id = ] ' job\_execution\_id '

#### Arguments

[ @ job\_execution\_id = ] *job\_execution\_id*  
The identification number of the job to stop. *job\_execution\_id* is **uniqueidentifier**, with default of **NULL**.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

None.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_delete\_job

Deletes a job.

#### Syntax

[jobs].**sp\_delete\_job** [ @job\_name = ] ' job\_name '

#### Arguments

**[ @ job\_name = ]** ' *job\_name* '

The name of the job to delete. *job\_name* is **nvarchar(128)**, with default of **NULL**.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

When a job is deleted, all job history is deleted.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_delete\_jobstep

Removes a job step from a job.

#### Syntax

[jobs].**sp\_delete\_jobstep** [ @job\_name = ] 'job\_name'

[ , [ @step id = ] step\_id ]

#### Arguments

**[ @ job\_name = ]** ' *job\_name* '

The name of the job. *job\_name* is **nvarchar(128)**, with default of **NULL**.

**[ @step\_id = ]** *step\_id*

The identification number of the step being removed. *step\_id* is an **int**, with no default.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

When a job is deleted, all job history is deleted.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_add\_target\_group

Add a target group, namely a collection of databases.

#### Syntax

[jobs].**add\_target\_group** [ @target\_group\_name = ] 'target\_group\_name'

[ , [ @target\_group\_id = ] target\_group\_id OUTPUT ]

#### Arguments

**[ @ target\_group\_name = ]** ' *target\_group\_name* '

The name of the target group to create. *target\_group\_name* is **nvarchar(128)**, with no default.

**[ @target\_group\_id = ]** target\_group\_id OUTPUT

The target group identification number assigned to the job if created successfully. *target\_group\_id* is an output variable of type **uniqueidentifier**, with a default of **NULL**.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

Target groups provide an easy way to target a job at a collection of databases.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_delete\_target\_group

Deletes a specific target group.

#### Syntax

[jobs].**delete\_target\_group** [ @target\_group\_name = ] ' target\_group\_name '

#### Arguments

**[ @ target\_group\_name = ]** ' *target\_group\_name*'

The name of the target group to delete. *target\_group\_name* is **nvarchar(128)**, with no default.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

None.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

### sp\_add\_target\_group\_member

Adds the specified target group member, database or collection of databases, to the specified target group.

#### Syntax

[jobs].**add\_target\_group\_member** [ @target\_group\_name = ] 'target\_group\_name'

[ @membership\_type = ] ‘membership\_type’ ]

[ , [ @target\_type = ] ‘target \_type’ ]

[ , [ @refresh\_credential\_name = ] ‘refresh\_credential\_name’ ]

[ , [ @server\_name = ] ‘server\_name’ ]

[ , [ @database\_name = ] ‘database\_name’ ]

[ , [ @elastic\_pool\_name = ] ‘elastic\_pool\_name’ ]

[ , [ @target\_id = ] ‘target\_id’ OUTPUT ]

#### Arguments

**[ @ target\_group\_name = ]** '*target\_group\_name*'

The name of the target group to which the member will be added. *target\_group\_name* is **nvarchar(128)**, with no default.

**[ @ membership\_type = ]** '*membership\_type*'

Specifies if the target group member will be included or excluded. *target\_group\_name* is **nvarchar(128)**, with default of **‘Include’**. Valid values for *target\_group\_name* are **‘Include’** or **‘Exclude’.**

**[ @ target\_type = ]** '*target\_type*'

The type of target database or collection of databases including all databases in a server, all databases in an Elastic pool or a database. *target\_type* is **nvarchar(128)**, with no default. Valid values for *target\_type* are **‘SqlServer’**, **‘SqlElasticPool’** or **‘SqlDatabase’.**

**[ @ refresh\_credential\_name = ]** '*refresh\_credential\_name*'

The name of the logical server. *refresh\_credential\_name* is **nvarchar(128)**, with no default.

**[ @ server\_name = ]** '*server\_name*'

The name of the logical server that should be added to the specified target group. *server\_name* should be specified when *target\_type* is **‘SqlServer’**. *server\_name* is **nvarchar(128)**, with no default.

**[ @ database\_name = ]** '*database\_name*'

The name of the database that should be added to the specified target group. *database\_name* should be specified when *target\_type* is **‘SqlDatabase’.** *database\_name* is **nvarchar(128)**, with no default.

**[ @ elastic\_pool\_name = ]** *‘elastic\_pool\_name'*

The name of the Elastic pool that should be added to the specified target group. *elastic\_pool\_name* should be specified when *target\_type* is **‘SqlElasticPool’**. *elastic\_pool\_name* is **nvarchar(128)**, with no default.

**[ @ target\_id = ]** *target\_group\_id* OUTPUT

The target identification number assigned to the target group member if created added to the target group. *target\_id* is an output variable of type **uniqueidentifier**, with a default of **NULL**.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

A job executes on all databases within a server or Elastic pool at time of execution, when a logical server or Elastic pool is included in the target group.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

#### Examples

The following example adds all the databases in the *London and NewYork* servers to the group *Servers Maintaining Customer Information*. You must connect to the Elastic jobs account database specified when creating the job account, in this case ElasticJobs*.*

--Connect to the Elastic jobs account database specified when creating the job account

USE ElasticJobs ;

GO

-- Add a target group containing server(s)

EXEC jobs.sp\_add\_target\_group @target\_group\_name = N'Servers Maintaining Customer Information'

GO

-- Add a server target member

EXEC jobs.sp\_add\_target\_group\_member

@target\_group\_name = N'Servers Maintaining Customer Information',

@target\_type = N'SqlServer',

@refresh\_credential\_name=N'mymastercred', --credential required to refresh the databases in server

@server\_name=N'London.database.windows.net' ;

GO

-- Add a server target member

EXEC jobs.sp\_add\_target\_group\_member

@target\_group\_name = N'Servers Maintaining Customer Information',

@target\_type = N'SqlServer',

@refresh\_credential\_name=N'mymastercred', --credential required to refresh the databases in server

@server\_name=N'NewYork.database.windows.net' ;

GO

--View the recently added members to the target group

SELECT \* FROM [jobs].target\_group\_members WHERE target\_group\_name= N'Servers Maintaining Customer Information';

GO

### sp\_delete\_target\_group\_member

Removes a specified target group member from the target group.

#### Syntax

[jobs].**delete\_target\_group\_member** [ @target\_group\_name = ] 'target\_group\_name'

[ , [ @target\_id = ] ‘target\_id’]

#### Arguments

**[ @ target\_group\_name = ]** '*target\_group\_name*'

The name of the target group from which to remove the target group member. *target\_group\_name* is **nvarchar(128)**, with no default.

**[ @ target\_id = ]** *target\_id*

The target identification number assigned to the target group member to be removed. *target\_id* is an **uniqueidentifier**, with a default of **NULL**.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

Target groups provide an easy way to target a job at a collection of databases.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

#### Examples

The following example removes the *London* server from the group *Servers Maintaining Customer Information*. You must connect to the Elastic jobs account database specified when creating the job account, in this case ElasticJobs*.*

--Connect to the Elastic jobs account database specified when creating the job account

USE ElasticJobs ;

GO

-- Retrieve the target\_id for a target\_group\_members

declare @tid uniqueidentifier

SELECT @tid = target\_id FROM [jobs].target\_group\_members WHERE target\_group\_name = 'Servers Maintaining Customer Information' and server\_name = 'London.database.windows.net'

-- Remove a target group member of type server

EXEC jobs.sp\_delete\_target\_group\_member

@target\_group\_name = N'Servers Maintaining Customer Information',

@target\_id = @tid

GO

### sp\_purge\_jobhistory

Removes the history records for a job.

#### Syntax

[jobs].**sp\_pruge\_jobhistory** [ @job\_name = ] 'job\_name'

[ @job\_id = ] job\_id

[ @oldest\_date = ] oldest\_date

#### Arguments

**[ @ job\_name = ]** ' *job\_name* '

The name of the job for which to delete the history records. *job\_name* is **nvarchar(128)**, with a default of **NULL**. Either job\_id or job\_name must be specified, but both cannot be specified.

**[ @job\_id = ]** job\_id

The job identification number of the job for the records to be deleted. *job\_id* is **uniqueidentifier**, with a default of **NULL**. Either job\_id or job\_name must be specified, but both cannot be specified.

**[ @oldest\_date = ]** oldest\_date

The oldest record to retain in the history. *oldest\_date* is **DATETIME2**, with a default of **NULL**. When *oldest\_date* is specified, sp\_purge\_jobhistory only removes records that are older than the value specified.

#### Return Code Values

0 (success) or 1 (failure)

#### Remarks

Target groups provide an easy way to target a job at a collection of databases.

#### Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. The restrict a user to just be able to monitor jobs, you can grant the user to be part of the following database role in the job account database specified when creating the job account:

* jobs\_reader

For details about the permissions of these roles, see the Permission section in this document. Only members of sysadmin can use this stored procedure to edit the attributes of jobs that are owned by other users.

#### Examples

The following example adds all the databases in the *London and NewYork* servers to the group *Servers Maintaining Customer Information*. You must connect to the Elastic jobs account database specified when creating the job account, in this case *ElasticJobs.*

--Connect to the Elastic jobs account database specified when creating the job account

EXEC sp\_delete\_target\_group\_member

@target\_group\_name = N'Servers Maintaining Customer Information',

@server\_name = N'London.database.windows.net';

GO

## Job Views

### [jobs].[jobs]

Returns information about jobs that are used by Elastic jobs to perform automated activities in Azure SQL Database.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **job\_name** | **nvarchar(128)** | Name of the job. |
| **job\_id** | **uniqueidentifier** | Unique ID of the job. |
| **current\_job\_version\_number** | **int** | Version of the job (automatically updated each time the job is modified). |
| **description** | **nvarchar(512)** | Description for the job. |
| **enabled** | **bit** | Indicates whether the job is enabled or disabled. **1** indicates enabled jobs, and **0** indicates disabled jobs. |
| **schedule\_interval\_type** | **nvarchar(50)** | Value indicating when the job is to be executed:  'Once',  'Minutes',  'Hours',  'Days',  'Weeks',  'Months' |
| **schedule\_interval\_count** | **int** | Number of schedule\_interval\_type periods to occur between each execution of the job. |
| **schedule\_start\_time** | **datetime2** | Date and time the job was last started execution. |
| **schedule\_end\_time** | **datetime2** | Date and time the job was last completed execution. |

### [jobs].[jobs\_executions]

Returns information about jobs that are used by Elastic job account to perform automated activities in Azure SQL Database.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **job\_execution\_id** | **uniqueidentifier** | Unique ID of an instance of a job execution. |
| **job\_name** | **nvarchar(128)** | Name of the job. |
| **job\_id** | **uniqueidentifier** | Unique ID of the job. |
| **job\_version\_number** | **int** | Version of the job (automatically updated each time the job is modified). |
| **step\_id** | **int** | Unique (for this job) identifier for the step. |
| **is\_active** | **bit** | Indicates whether information is active or inactive. **1** indicates active jobs, and **0** indicates inactive. |
| **create\_time** | **datetime2** | Date and time the job was created. |
| **start\_time** | **datetime2** | Date and time the job started execution. **NULL** if the job has not yet been executed. |
| **end\_time** | **datetime2** | Date and time the job finished execution. **NULL** if the job has not yet been executed or has not yet completed execution. |
| **current\_attempts** | **int** | Number of times the step was retried. |
| **current\_attempt\_start\_time** | **datetime2** | Date and time the job started execution. |
| **last\_message** | **nvarchar(max)** | Job or step history message. |
| **target\_type** | **nvarchar(128)** | Type of target database or collection of databases including all databases in a server, all databases in an Elastic pool or a database. Valid values for target\_type are ‘SqlServer’, ‘SqlElasticPool’ or ‘SqlDatabase’. |
| **target\_id** | **uniqueidentifier** | Unique ID of the target group member. |
| **target\_group\_name** | **nvarchar(128)** | Name of the target group. |
| **target\_server\_name** | **nvarchar(128)** | Name of the logical server contained in the target group. Specified only if target\_type is ‘SqlServer’**.** |
| **target\_database\_name** | **nvarchar(128)** | Name of the database contained in the target group. Specified only when target\_type is ‘SqlDatabase’. |

### [jobs].[jobsteps]

Returns information for the steps in a job used by Elastic job account to perform automated activities.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **job\_name** | **nvarchar(128)** | Name of the job. |
| **job\_id** | **uniqueidentifier** | Unique ID of the job. |
| **job\_version\_number** | **int** | Version of the job (automatically updated each time the job is modified). |
| **step\_id** | **int** | Unique (for this job) identifier for the step. |
| **command\_type** | **nvarchar(50)** | Type of command to execute in the job step. For v1, value must equal to and defaults to **TSql**. |
| **command\_source** | **nvarchar(6)** | Location of the command. For v1, **Inline** is the default and only accepted value.  Note: Column size will be changed to nvarchar(50) in future release. |
| **command** | **nvarchar(max)** | The commands to be executed by Elastic jobs through command\_type. |
| **credential\_name** | **nvarchar(128)** |  |
| **target\_group\_name** | **nvarchar(128)** | Name of the target group. |
| **target\_group\_id** | **uniqueidentifier** | Unique ID of the target group. |
| **initial\_retry\_interval\_seconds** | **int** | The delay before the first retry attempt. Default value is 1. |
| **maximum\_retry\_interval\_seconds** | **int** | The maximum delay between retry attempts. If the delay between retries would grow larger than this value, it is capped to this value instead. Default value is 120. |
| **retry\_interval\_backoff\_multiplier** | **real** | The multiplier to apply to the retry delay if multiple job step execution attempts fail. Default value is 2.0. |
| **retry\_attempts** | **int** | The number of retry attempts to use if this step fails. Default of 0, which indicates no retry attempts. |
| **step\_timeout\_seconds** | **int** | The amount of time in minutes between retry attempts. The default is 0, which indicates a 0-minute interval. |
| **output\_type** | **nvarchar(11)** | Location of the command. For v1, Inline is the default and only accepted value.  Note: Column size will be changed to nvarchar(50) in future release. |
| **output\_credential\_name** | **nvarchar(128)** | Name of the credentials to be used to connect to the destination server to store the results set. |
| **output\_subscription\_id** | **uniqueidentifier** | Unique ID of the subscription of the destination server\database for the results set from the query execution. |
| **output\_resource\_group\_name** | **nvarchar(128)** | Resource group name where the destination server resides. |
| **output\_server\_name** | **nvarchar(256)** | Name of the destination server for the results set. |
| **output\_database\_name** | **nvarchar(128)** | Name of the destination database for the results set. |
| **output\_schema\_name** | **nvarchar(max)** | Name of the destination schema. Defaults to dbo, if not specified. |
| **output\_table\_name** | **nvarchar(max)** | Name of the table to store the results set from the query results. Table will be created automatically based on the schema of the results set if it doesn’t already exist. Schema must match the schema of the results set.  Note: Column size will be changed to nvarchar(128) in future release. |

### [jobs].[target\_groups]

Lists all target groups.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **target\_group\_name** | **nvarchar(128)** | The name of the target group, a collection of databases. |
| **target\_group\_id** | **uniqueidentifier** | Unique ID of the target group. |

### [jobs].[target\_groups\_members]

Lists all target group members in the specified group. If no group is specified, Elastic job account returns information about all target group members.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **target\_group\_name** | **nvarchar(128)** | The name of the target group, a collection of databases. |
| **target\_group\_id** | **uniqueidentifier** | Unique ID of the target group. |
| **membership\_type** | **int** | Specifies if the target group member is included or excluded in the target group. Valid values for target\_group\_name are ‘Include’ or ‘Exclude’. |
| **target\_type** | **nvarchar(128)** | Type of target database or collection of databases including all databases in a server, all databases in an Elastic pool or a database. Valid values for target\_type are ‘SqlServer’, ‘SqlElasticPool’ or ‘SqlDatabase’. |
| **target\_id** | **uniqueidentifier** | Unique ID of the target group member. |
| **refresh\_credential\_name** | **nvarchar(128)** | Name of the credential used to connect to the target group member. |
| **subscription\_id** | **uniqueidentifier** | Unique ID of the subscription. |
| **resource\_group\_name** | **nvarchar(128)** | Name of the resource group in which the target group member resides. |
| **server\_name** | **nvarchar(128)** | Name of the logical server contained in the target group. Specified only if target\_type is ‘SqlServer’**.** |
| **database\_name** | **nvarchar(128)** | Name of the database contained in the target group. Specified only when target\_type is ‘SqlDatabase’. |
| **elastic\_pool\_name** | **nvarchar(128)** | Name of the Elastic pool contained in the target group. Specified only when target\_type is ‘SqlElasticPool’. |