

Export failing with GatewayTimeout due to Firewall Setting

Last updated by | Balaji Barmavat | Nov 18, 2020 at 7:15 PM PST

Contents

- [Export failing with GatewayTimeout due to Firewall Setting](#)
 - [Issue](#)
 - [Troubleshooting](#)
 - [Solution](#)
 - [Kusto Queries:](#)
 - [Classification](#)

Export failing with GatewayTimeout due to Firewall Setting

Issue

Customer is running the PowerShell command `New-AzSqlDatabaseExport` for exporting their Azure SQL Database into a BACPAC file on BLOB storage. The command is failing after exactly 1 minute with the following error:

```
New-AzSqlDatabaseExport : GatewayTimeout: The gateway did not receive a response from 'Microsoft.Sql' within the specified time period.
```

They tried the same using SSMS and it worked; if they try to import the database through `New-AzSqlDatabaseImport`, they are getting the same GatewayTimeout error.

Troubleshooting

MonManagement shows the attempts for exporting the database. They are all failing after 1 minute with the error details given above.

Same thing on **HttpIncomingRequests**: export commands are failing after 60054ms with `httpStatusCode=504` - > Gateway Timeout

The GatewayTimeout appears on **EventServiceEntries** as well but no further details.

MonLogin has a lot of errors 40615 state 82 for `application_name="SQL Database Import/Export Service"`. Each of those login attempts is taking only a few milliseconds, there are several occurring within the minute of each export attempt. The attempts within the minute follow the standard retry pattern: after 10s/20s/30s.

So it looks as if the export attempt is failing and retried 3 times, each attempt failing again with the same 40615/82 error.

82	Firewall blocked the connection	40615 error state 82 indicates that customer's IP address is not present in their list of firewall rules. Customer needs to add IP address to firewall rules to gain access to database.

Solution

The customer had set the "Allow access to Azure Services" option to OFF for the Azure SQL Database server. Once they switched it back on, the New-AzSqlDatabaseExport command ran successfully.

Kusto Queries:

```
MonManagement
| where TIMESTAMP >= datetime(2019-09-11 13:00:00Z)
| where TIMESTAMP <= datetime(2019-09-11 16:00:00Z)
| where logical_server_name == "nhsuk-i-synctesting-server-1"
//| where logical_database_name == "testsync_aaaa"
| where event contains "importexport"
| project TIMESTAMP, operation_type, operation_parameters, originalEventTimestamp, resource_group, logical_ser
| limit 10000
```

```
HttpIncomingRequests
| where TIMESTAMP >= datetime(2019-09-11 13:00:00Z)
| where TIMESTAMP <= datetime(2019-09-11 16:00:00Z)
| where subscriptionId =~ "subscriptionId"
| where targetUri contains "nhsuk-i-synctesting-server-1"
//| where targetUri contains "serviceendpointissue"
| where httpMethod != "GET"
//| where httpStatusCode > 300
//| where correlationId in ("511a8366-27e0-40c7-8eb1-49e816759a9b")
//| where commandName == "Restore-AzSqlInstanceDatabase"
//| where clientRequestId =~ "764488f0-dc3d-4deb-82d9-b4acfc09916a"
| project TIMESTAMP, PreciseTimeStamp, operationName, httpMethod, commandName, parameterSetName, targetUri, us
| limit 1000
```

```
EventServiceEntries
| where TIMESTAMP >= datetime(2019-09-11 14:00:00Z)
| where TIMESTAMP <= datetime(2019-09-11 15:00:00Z)
| where subscriptionId =~ "subscriptionId"
| where correlationId =~ "48dc64dd-e9ab-4e92-8099-ee8d79547308"
| project TIMESTAMP, Deployment, ActivityId, subscriptionId, correlationId, eventInstanceId, principalOid, pri
| limit 1000
```

```
MonLogin
| where TIMESTAMP >= datetime(2019-09-11 14:00:00Z)
| where TIMESTAMP <= datetime(2019-09-11 15:00:00Z)
| where LogicalServerName contains "nhsuk-i-synctesting-server-1" or logical_server_name contains "nhsuk-i-syn
| extend ProxyOrRedirect = iif( result == "e_crContinueSameState", "Proxy" , "Redirect")
| extend ProxyOrRedirectEmpty = iif( result == "" and ProxyOrRedirect == "Redirect", "(ignore ProxyOrRedirect)"
| project originalEventTimestamp, type, event, error, state, is_user_error, is_success, os_error, sni_error, s
| limit 10000
| order by originalEventTimestamp asc
```

Classification

Root Cause: Azure SQL DB v2\Connectivity\Login Errors\Firewall errors and misconfigurations

How good have you found this content?

