# Gateway reset connections idle for 30m

Last updated by | Holger Linke | Mar 1, 2023 at 4:39 AM PST

#### **Contents**

- Issue
- Investigation / Analysis
  - Telemetry
  - Network Trace
- Mitigation
- Public Doc Reference

### Issue

Azure SQL Databases are deployed in multi-tenant clusters. To keep the resource consumption low, idle session are killed by the Azure gateway to cleanup the connections which have been idle for 30 minutes. This connection reset occurs when the client is using proxy connection policy instead of redirect. Redirect connections bypass the gateway and its proxy, and the gateway is not managing the connection lifetime then.

Clients are able to reconnect after a connection reset if they have enabled the retry feature:

- For ODBC, it is controlled through the ODBC connection string keywords connectRetryCount and connectRetryInterval. See Connection Resiliency in the Windows ODBC Driver ☑ for further details and a code example.
- For JDBC, it is controlled through the connection options connectRetryCount and connectRetryInterval . See <a href="Connection resiliency">Connection resiliency (JDBC)</a> <a href="DBC">DBC</a> for further details.

# Investigation / Analysis

# **Telemetry**

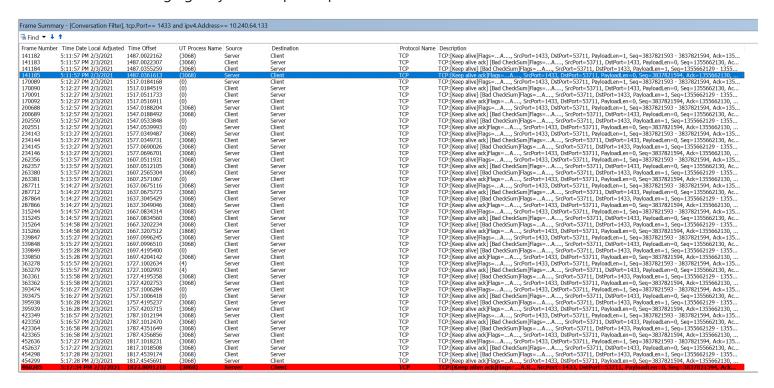
ASC will provide you with information about idle connection reset on page "Connectivity -> Disconnects".

In the MonLogin Kusto table, the idle connection reset is logged as "idle\_session\_execution\_stats" event:

```
let srv = "servername";
let startTime = datetime(2023-02-15 04:00:00Z);
let endTime = datetime(2023-02-16 05:00:00Z);
let timeRange = ago(1d);
MonLogin
 where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
//| where TIMESTAMP >= timeRange
 where LogicalServerName contains srv
 where event == "idle session execution stats"
 project TIMESTAMP, LogicalServerName, package, event, killed, remaining, remaining dac connections
Sample output:
TIMESTAMP
                             LogicalServerName package
                                                                                        killed remaining
2023-02-15 04:01:37.4129542 servername
                                               sqlserver idle session execution stats 15
2023-02-15 04:02:37.4126643 servername
                                               sqlserver idle session execution stats 11
                                                                                                674
2023-02-15 04:03:37.4280076 servername
                                               sqlserver idle session execution stats 10
                                                                                                677
2023-02-15 04:04:37.3964649 servername
                                               sqlserver idle session execution stats 16
                                                                                                670
2023-02-15 04:05:37.3961980 servername
                                               sqlserver idle session execution stats 13
                                                                                                665
```

#### **Network Trace**

If you capture a network trace on the client side, you'll see that the server/gateway sends a **RST** reset packet to the client after exchanging only TCP KeepAlive packets for some time:



# Mitigation

One way to avoid idle connection connection resets would be switching the connection policy from "proxy" to "redirect". The customer can check the current setting and change to redirect through the Azure portal, PowerShell, or Azure CLI. See <u>Change the connection policy</u>  $\square$  for the steps.

Connection policy I lists some prerequisites and limitations for using redirect:

- Clients need to allow outbound communication from the client to all Azure SQL IP addresses in the region on ports in the range of 11000 to 11999. Use the Service Tags for SQL to make this easier to manage.
- Clients need to allow outbound communication from the client to Azure SQL Database gateway IP addresses on port 1433.
- If the client is an Azure Virtual Machine, you can accomplish this using Network Security Groups (NSG) with service tags ☑.
- If the client is connecting from a workstation on-premises then you may need to work with the customer's network admin to allow network traffic through your corporate firewall.
- Connections to private endpoint only support Proxy as the connection policy.

If the customer is using Private Endpoint connections, switching to redirect is not possible though, due to the limitation that private endpoints only support proxy connections currently. In this case, they can avoid the idle reset issue by sending dummy queries to server periodically, e.g. a SELECT 1 query every 5 - 10 minutes.

## **Public Doc Reference**

- Connection Resiliency in the Windows ODBC Driver [2]
- Microsoft ODBC Driver for SQL Server ☑

These articles apply to all platform environments: Linux, Mac, Windows.

#### How good have you found this content?

