Database Login Workflow in Sterling

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:34 PM PDT

Contents

Introduction

One of the defining characteristics of SAWA v2 is the new login workflows. In SAWA v2 there is no Gateway service that control the entire login process for the entire cluster, performing authentication, server location, TDS redirection and T-SQL parsing to complete database CRUD operations.

In SAWA v2 diverse independent stateless services perform the necessary steps to locate and redirect client connections to the SQL Server Instances serving databases. There will be no T-SQL parsing on these components, all T-SQL parsing occurs in SQL Server Engine.

Newer TDS clients will be redirected to the SQL Server Instance serving the database while older TDS client will be go thru a SNI-Level proxy.

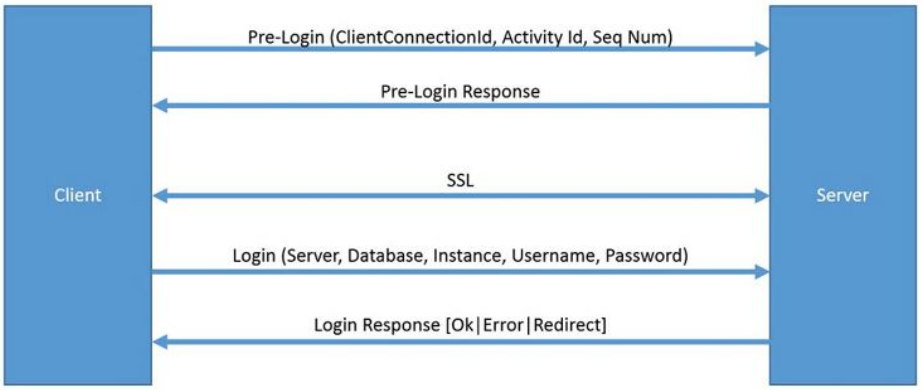
Components Involved in the Connectivity Workflow

- Control Ring
- TDS Redirector: redirects clients running TDS 7.4 or above, serves as a proxy for downward clients. Also supports UCS protocol for GeoDR.
  - Alias Service: serves as a directory to target DbSvc node.
- Tenant Ring
  - TDS socket duplicator: duplicate socket and SSL context to the appropriate SQL Instance.
  - CloudExtensions.dll (SQlClr): runs inside sqlservr.exe and performs firewall, and UserDb to logical master communication.

Connectivity Workflow

The connectivity workflow for SAWA v2 involves the Control Ring Alias Service and Tenant Nodes.

Login Packets



Picture 1

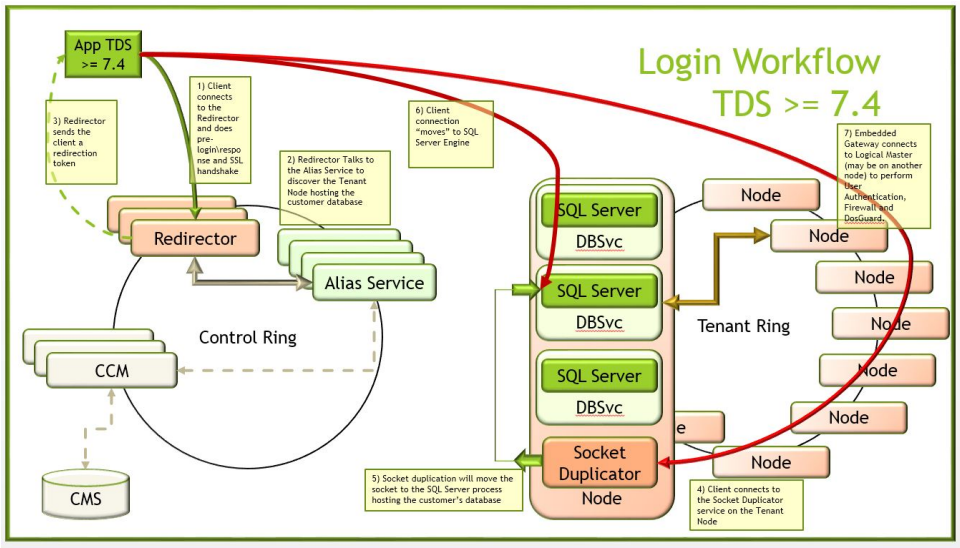
1. Application connects to cluster VIP port 1433 using the DNS name of the service
2. The connection is forwarded to one of the Redirector Nodes
    1. Client Connects to the Redirector and does pre-login\response, SSL and login
        1. Client connect
        2. Client pre-login
        3. SSL Handshake
        4. Receive client login request
3. The TDS Redirector retrieves the DNS+port of the tenant ring that hosts the logical server (WinFab Service).
    1. Redirector performs lookups in WinFab to find the customer's database's location
    2. This is done using the Alias Service of the Control Ring, which provides a mapping between a logical URL of an logical server/database to an DNS and Port address of the instance.
4. TDS Redirector checks the version of TDS and do following
    1. TDS 7.4 or above: redirects the client to the DNS/port number of the target instance host service
    2. Below TDS 7.4: TDS redirector will be a thin proxy to the target instance host service. It will also pass the client IP to the target instance host
5. TDS 7.4 or above client re-starts the connection process connecting to the VIP/Port provided by the redirector. Older clients will continue to use the same connection thru proxy (sni-level proxy).
6. Client connects to the Host Instance and does Pre-login\response, SSL and login
7. The Instance Host identifies which SQL Server instance hosts the logical server and database specified in the login request
8. The Instance Host duplicates the socket into the the SQL instance and communicates this to the instance via a dedicated named pipe. The SSL shared session key is passed to the instance
9. Login request will be forwarded to the embedded Gateway hosted inside of the SQL instance.
10. Embedded Gateway will do the following steps including talking to the SQL instance where logical master is hosted (Embedded Gateway talking to different SQL instance where logical master is hosted)

1. Authenticate login (embedded gateway might connect to Master dataset if master is running on another SQL Server Instance)

2. If the SQL Instance is not Logical Master, it creates an ODBC connection to the Logical Master. If the Master Database is running on a different DbSvc than the database we are trying to access, the Embedded Gateway will be talking to a different SQL instance where logical master is hosted. The Embedded Gateway will talk to another SQL instance using modified version of ODBC driver 11. (login sequence Xodbc)

3. Logical Master does DosGuard check, server level firewall check and authenticates credentials

11. The instance will sends the login response to the client



Picture 2

Supported Client Drivers and specific SAWA v2 impact on clients

There are some characteristics of SAWA v2 that will influence client drivers. SAWA V1 has various mechanisms to fulfill the connection request coming from clients. This includes

1. Detecting a move of primary replica and retrieving the new location

2. Making usage of secondary replica during login process when primary is not readily available.

3. After a connection is established, detecting that back end isntance is disconnected and retry to reconnect transparently.

With move to SAWA V2, our service will fail fast for the cases mentioned above. This implies that the client application will end up retrying more often than it needed to do with SAWA V1.
When ADO.Net connection pooling is enabled (http://msdn.microsoft.com/en-us/library/8xx3tyca.aspx), and if a timeout error or other login error occurs, an exception will be thrown and subsequent connection attempts will fail for the next five seconds. If the application attempts to connect within the blocking period, the first cached exception will be thrown again. Subsequent failures after a blocking period ends will result in a new blocking periods that is twice as long as the previous blocking period, up to a maximum of one minute. With the fast fail approach of SAWA V2, more existing application might observe block period.

Customers will have to enhance retry logic in their applications. Customers have complained that it is not easy to incorporate retry logic into existing applications. This is an existing issue that will get aggravated.

Drivers that do not support TDS 7.4 will have to go through redirector proxy. The expectation is that as applications are upgraded more client connection will be able to leverage redirection and avoid the proxy. Click here for a List of supported drivers and TDS versions supported.

How good have you found this content?