Connection taking a long time

Last updated by | Mohammad Abu Hamdieh | Oct 20, 2022 at 10:12 AM PDT

Issue

Creating or connecting to server takes long time and sometimes it timeouts.

Troubleshooting:

PostgreSQL creates dedicate process for each connection, which is an expensive operation, having connection pooler minimize creating new connections and reuse already created ones.

Below is a list of things that needs to be checked for why connections to server is taking too long.

- Check server availability:
  If server was restarting or shutting down it will take time to connect for some drivers, as some of them have retry options, so check if server was restarting or shutting down.
- Server's performance:
  if server has high IO or CPU usage, that will slow down creating new connections, since as mentioned, creating new connections is an expensive operation.
- If customer is using connection pooler, Please check connections in Connection pooler if they are utilized and active, for instance in HikariCP you might see an error connection is not available error, but this indicate all connections in pools are used, more info this link
  Some drivers retry operations which can add latency to create new connections or get a connection from the pool.
- Network related issues:
  If All above were already checked and no issue found, you can check below points from network side, please make sure to open a collab with network team, you can do below sanity check if needed.
  - **routing**:
    - Using public route can sometime slow connecting to server, its better to use private access.
    - if customer uses public access from outside azure, its good practice to use VPN or similar solution or migrate his application to Azure.
    - If customer is using private access but in different VNET, check routing table and check if he has low number of hops.
    - You can check routing by running:
      - traceruote IP/HOSTNAME (Linux)
      - tracert IP/HOSTNAME (Windows)
  - **Slow DNS lookup**:
    You can check slow DNS lookup by running:
    - Measure-Command { nslookup POSTGRESQL_SERVER_FULL_HOSTNAME } (Windows), check Milliseconds output but not TotalMilliseconds
    - dig POSTGRESQL_SERVER_FULL_HOSTNAME (Linux), check Query time value from output.
  - something in the middle between server and application.(for instance firewall, Load balancer)
    for such cases having tcpdump or wireshark output can show what and what is traffic status between client and server.