Error 18456 state 127

Last updated by | Shital Modi | Jun 23, 2021 at 9:11 AM PDT

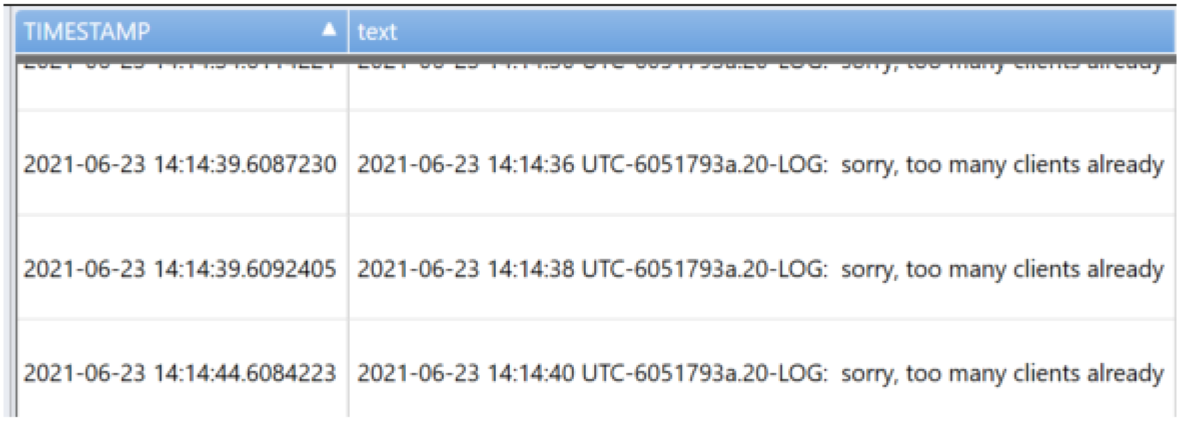
Issue: Intermittently, unable to connect Azure database for PostgreSQL server and seeing below error.

Customer side Error: 'too many connections' while connecting to Azure database for PostgreSQL.

Investigation/Analysis : CSS telemetry level

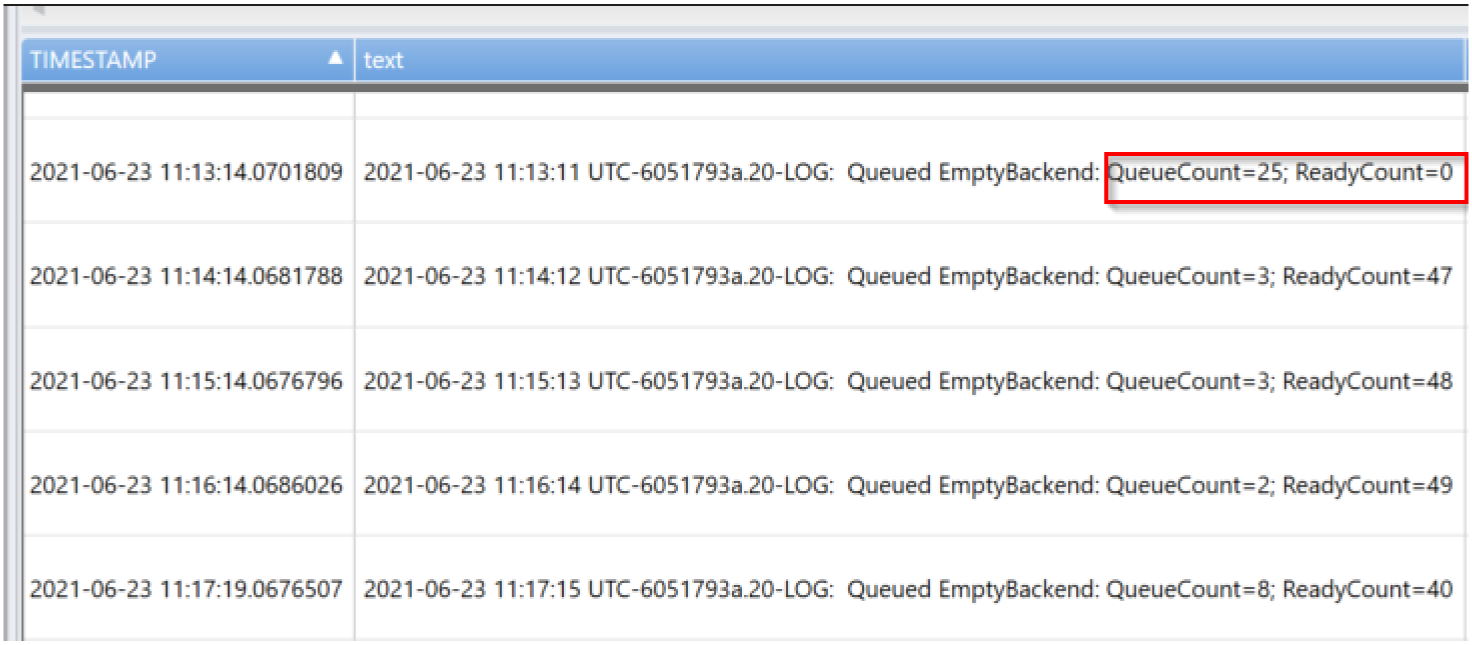
Verify using below Kusto:

```
MonRdmsPgSqlSandbox
| where LogicalServerName in ("servername")
|where text contains "too many"
|project TIMESTAMP,text
```



TIMESTAMP	text
2021-06-23 14:14:34.6174221	2021-06-23 14:14:36 UTC-6051793a.20-LOG: sorry, too many clients already
2021-06-23 14:14:39.6087230	2021-06-23 14:14:36 UTC-6051793a.20-LOG: sorry, too many clients already
2021-06-23 14:14:39.6092405	2021-06-23 14:14:38 UTC-6051793a.20-LOG: sorry, too many clients already
2021-06-23 14:14:44.6084223	2021-06-23 14:14:40 UTC-6051793a.20-LOG: sorry, too many clients already

```
MonRdmsPgSqlSandbox
| where LogicalServerName in ("servername")
|where TIMESTAMP >=  datetime(2021-06-14 01:00) and originalEventTimestamp <= datetime(2021-06-23 17:00)
|where text contains "ReadyCount"
//|where text contains "too many clients"
|project TIMESTAMP,text
```

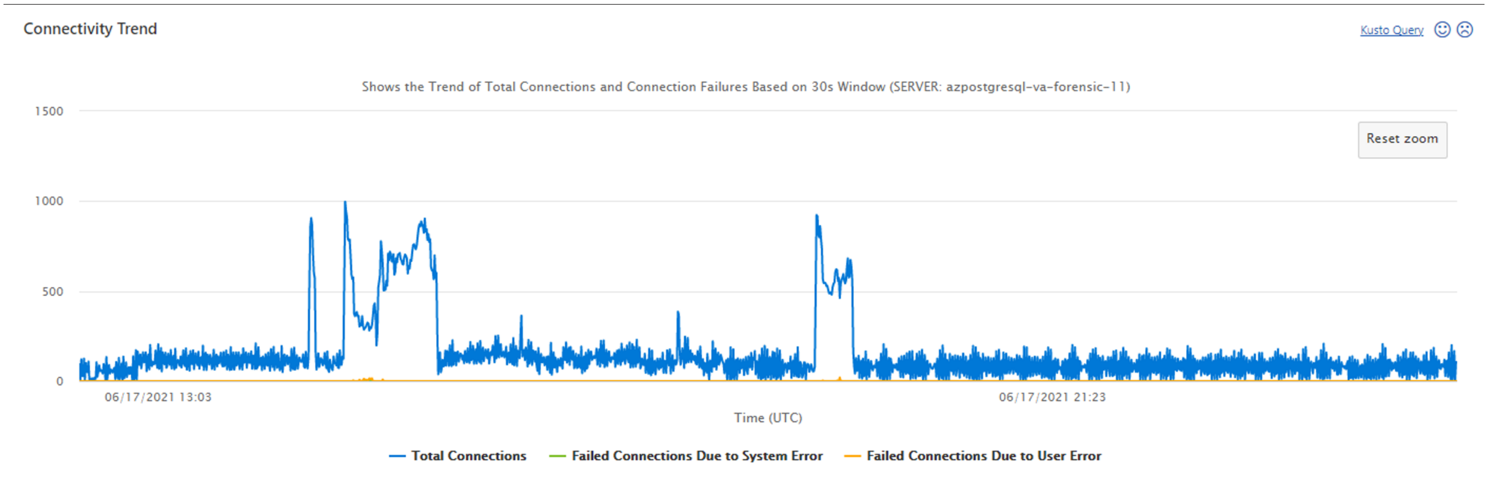


TIMESTAMP	text
2021-06-23 11:13:14.0701809	2021-06-23 11:13:11 UTC-6051793a.20-LOG: Queued EmptyBackend: QueueCount=25; ReadyCount=0
2021-06-23 11:14:14.0681788	2021-06-23 11:14:12 UTC-6051793a.20-LOG: Queued EmptyBackend: QueueCount=3; ReadyCount=47
2021-06-23 11:15:14.0676796	2021-06-23 11:15:13 UTC-6051793a.20-LOG: Queued EmptyBackend: QueueCount=3; ReadyCount=48
2021-06-23 11:16:14.0686026	2021-06-23 11:16:14 UTC-6051793a.20-LOG: Queued EmptyBackend: QueueCount=2; ReadyCount=49
2021-06-23 11:17:19.0676507	2021-06-23 11:17:15 UTC-6051793a.20-LOG: Queued EmptyBackend: QueueCount=8; ReadyCount=40

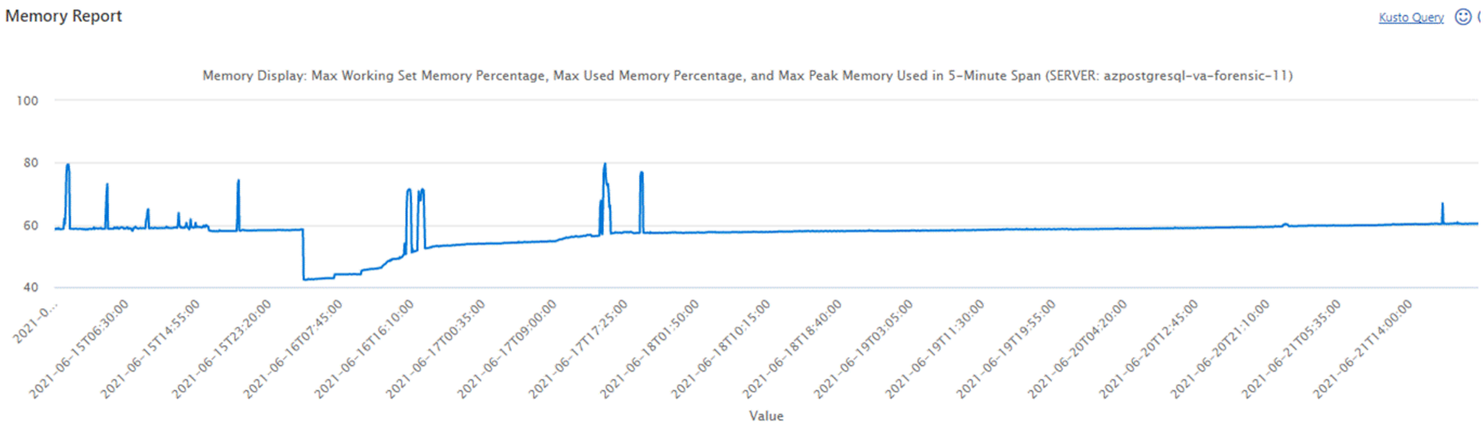
This issue could triggered ,whenever QueueCount is greater than ReadyCount **[MSFT internal only, please do not share with cx]**

Mitigation : ASC verification [Customer ready content: Please update numbers as per your case analysis then share with customer]

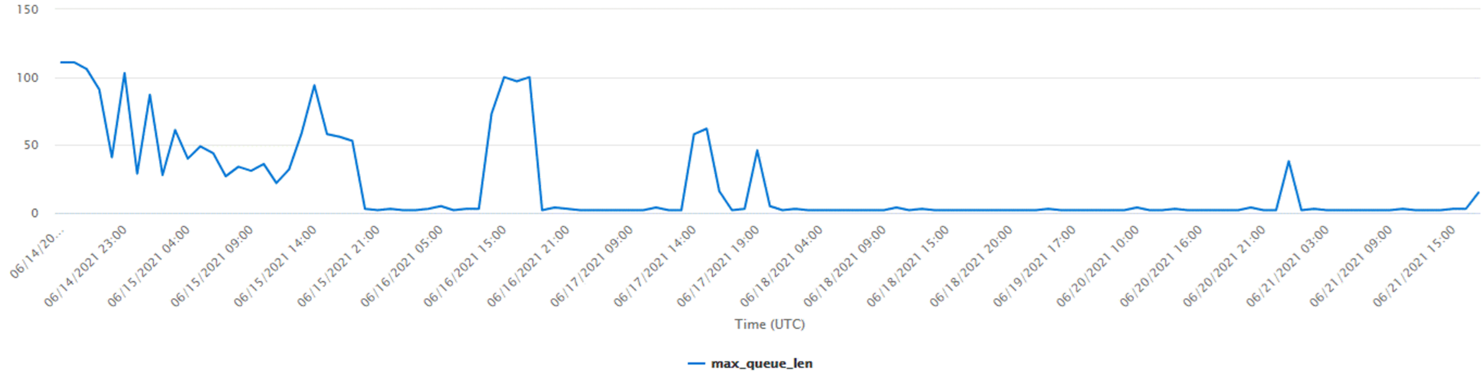
Connectivity trend : There are ~1000 connections made every 30 secs, not a good practice on Postgres running anywhere (cloud or On-prem or IAAS). Connections in Postgres are served serially and take up resources (both CPU and Memory). Postgres also has to do an initial TLS negotiation which takes 100-200 ms, so handling so many connections serially will also take up time.



Memory Spike matches with queue length spike.



Queue length:



Looking at the trend and rate at which connections are churned the database will not be able to keep it up. Too many connections error -> This error doesn't always mean that the active connections on the database server have reached the max limit. Note the bombarding connections on the server will increase the queue depth and as you can see the queue depth here during those humongous spikes reached. With this if the incoming rate continues it will throw too many connections error. Please Note opening and closing connections at faster rate also slows down tearing down the backend. That might also cause too many clients even though there are few active connections the server.

Recommendations: Consider using a connection pooler like pgBouncer between your application and Postgres server to optimize connection management.

Reference document:

Connection handling best practice : https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/connection-handling-best-practice-with-postgresql/ba-p/790883

Steps to install pgbouncer: https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/steps-to-install-and-setup-pgbouncer-connection-pooling-proxy/ba-p/730555