

Create or drop database take too long and never complete - PGFS

Last updated by | Eduardo Santana | Dec 30, 2022 at 1:39 AM PST

Create or Drop Database takes too long to complete or timeout

The Create or Drop Database statements internally trigger a checkpoint.

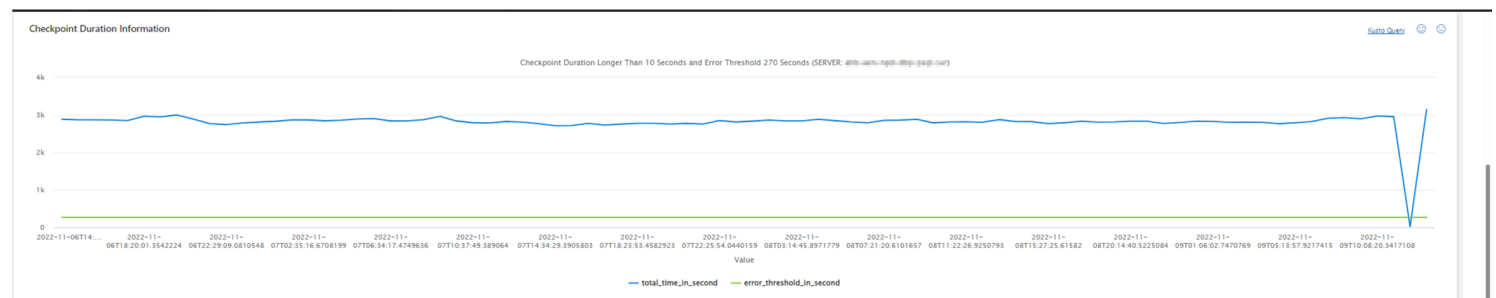
Checkpoints are known to take long time when there are large number of snapshot change files (WALs), which are produced during a high workload on a server. When replication is set (read-replicas or logical replication) there can be some WAL's files increase if the "consumer" cannot keep up with the applying all the WAL's sent by primary server, thus WAL files cannot be deleted / reused on the primary server. Also, it could be an issue with the WAL archiver process.

The most common issue we have faced several times is when logical replication is set, but consumer does not consume the changes, resulting in such files accumulation on primary server side. These files must be "visited" for every checkpoint, causing checkpoint to slow down and consequently the Create/Drop Database to take long time to execute or even leading to a timeout.

Regarding WAL files, they are need for PITR or replication purposes and will be deleted/reused when no longer needed. They are deleted automatically once replica consumer consume the changes by reactivating or dropping the replication slots.

You will notice in this situation an increase of storage usage due to WAL files being accumulated and high checkpoint duration time.

Check our telemetry to see if the checkpoints are taking really long which correspondingly causes DB create/drop to take long as well. You can check that from ASC (Perf tab) :



Please check below TSG about unexpected increase in the storage consumption:

https://dev.azure.com/Supportability/AzureDBPostgreSQL/_wiki/wikis/AzureDBPostgreSQL/767325/Unexpected-increase-in-storage-usage

Please share this link with customer - they are strongly advised to monitor replication slots and drop unused/inactive ones to avoid severe problems, including unavailability in case of a restart/failover/deployment.

Once they drop the inactive slots, they should be able to create the database and the checkpoint time should be under normal rang.

<https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-logical#monitoring> 