PgSQL-Replica-001: Server Reports INACTIVE PHYSICAL Replication slots

Last updated by | Lisa Liu | Nov 6, 2020 at 10:35 AM PST

1. **1. EVEN THOUGH THIS IS A SEV3, INCIDENT PLEASE TRY TO MITIGATE ASAP.**
   WE CAN'T LEAVE THE REPLICA IN THIS STATE FOR DAYS.
   THIS CAN HAVE ADVERSE AFFECT OF THE PRIMARY SERVER BY CAUSING STORAGE FULL ISSUES AND ALSO CAUSE REPLICA TO LAG WAY BEHIND THAT IT CAN NEVER CATCHUP.
   THANK YOU!

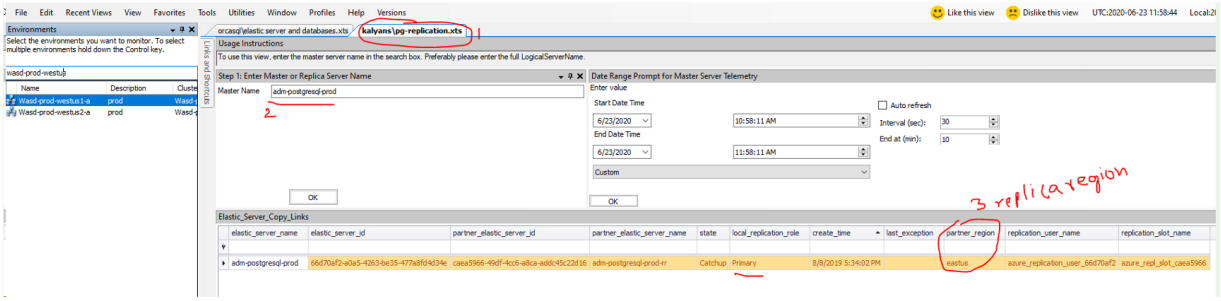
   2. Please check slot type in the incident title, if it is Logical slot, follow this TSG PgSQL-Replica-001: Server Reports Inactive LOGICAL Replication slots

3. FIND PRIMARY AND REPLICA SERVER NAMES AND REGIONS AND CREATION TIMES:

A. First try using the **pg-replication xts view** to determine the name of the primary and replica servers and their regions SOP: PG-Replication XTS View

- Primary server- The name of the server in the tile of the inactive replication slot incident is that of the primary server. **This is important to KNOW**

- Replica server - You have to find the name and the region of the **replica** server using the pg-replication.xts view and investigate the state of the replica in the replica's region.
  In this incidents, it is the replica server that is "usually" problematic, not the master server.
  - Check the replica region (if its cross region). Otherwise both primary and secondary/replica server details are in same section - Elastic_server_copy_links (pic below for cross region)
  - In Views select the partner_region/replica region.
  - Type same master name . Often replica server name prefix will be same as primary server name.
  - For application/instance name of replica use "Elastic Server and Databases.xts" view.



Select the environments you nant to monitor. To select •multiple environments hold down the Control key Tools Utilities Window Profiles Help Versions orcasql%glastc server and databases xts kalya replication.xts Usage Instructions To use this view enter the master server name in the search bon Preferably please enter the full LogicalServerName Step l: Enter Master or Replica Server Name Master Name admvostgresqlvrod 2- Date Range Prom Enter value Start Date Tme 6/23/2020 End Date Tme 6/23/2020 for Master Server Telemet Like this view 30 Dislike this view UT.•2020 06-23 r epli ca bon _slo t_name _66d70af2 Local: 21 nasd-prod4Nestup Name Wasdvrod4vestus2-a Descripton prod Wasdl W asd par tner

_elasbc_serv er _id Auto refresh Interval (sec): End at (min): last_excepbon elasbc server name adm-postgresql-prod elasbc server id par tner _elasbc_serv er _ name adm -po -prod -rr sta te Ca tchup Bcal _r eplica bon Primar y r ole 66dnaf2-aoa5-4263-be35-477a8fd4d34e caea5%6-4gdf-4cc6-a8ca-addc45c22d16 create time 8/8/2019 par tner _r egion r eplicabon _user _name eastus "/>

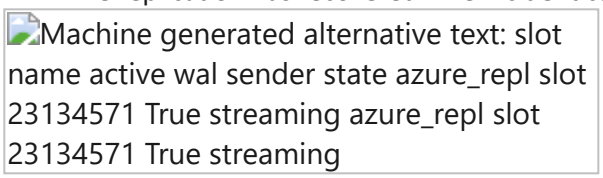
B. Also check the creation times of the primary and replica servers in "elastic server and database" view or any other way.
It will be good to understand if this is a newly recreated replica and also if it is a recently created primary.

4. <u>CONFIRM CURRENT STATE OF STREAMING REPLICATION OR FILE SHIPPING REPLICATION</u>

Check if the Server with the appropriate replication slot is currently streaming using the following query using the server name and
slot name in the incident (*slot name* is present in the details column of the table in the incident)

```
MonDmPgSqlReplicationStatsPrimary
| where TIMESTAMP > ago (1h)
| where LogicalServerName == '{primary_server_name}'
| where slot_name ==  '{slot_name}'
| project PreciseTimeStamp, LogicalServerName, AppName, slot_name, active, wal_sender_state, application_name, total_lag_in_bytes
| order by PreciseTimeStamp desc
```

A. If the server is currently streaming (latest 2 records have 'active' column set to 'True' and 'wal_sender_state' column is set to 'streaming') you are done.
The replication has recovered. The incident can be mitigated and resolved



B. Otherwise currently replication is broken for the given replication slot and the master server is accumulating wal files shipping in the data\pg_wal folder and also the backup fileshare;
Continue to step 4 after checking C below

C. Before continuing to step 5 just check if replica is still replicating using FILE SHIPPING REPLICATION.
When streaming breaks and in other conditions where there is a lot to catchup, replica switches to use file shipping replication method.
Check the first query below to see if it is actively doing it. You should see some logs applied in the CURRENT TIME - 20/25 MINs.

```
MonRdmsPgSqlSandbox
| where LogicalServerName == "replica_server_name"
| where text contains "restored transaction log" //Sometimes just search for "restored" for PFS servers

You can use this query to see where the primary is
MonRdmsPgSqlSandbox
| where LogicalServerName == "primary_server_name"
| where text contains "archived transaction log"  //Sometimes just search for "archived" for PFS servers

The different in the log file names of what primary is archiving and replica is restoring will tell how far behind the replica is.
As long is replica is close to primary here it is still getting the latest data, by file shipping and not thru streaming replication.

It is still important to fix the streaming replication, because primary is holding all
```

these files still for the replica.
Only in streaming replication mode, it knows where the replica is, in file shipping mode it does not know that replica already applied all these files.

In this case monitor the storage_percent metric of the primary server to see it if it is not reaching 90%, engage PR DRI if you see a risk of storage full.

5. **CHECK MANAGEMENT SERVICE OPERATIONS ON PRIMARY OR REPLICA**

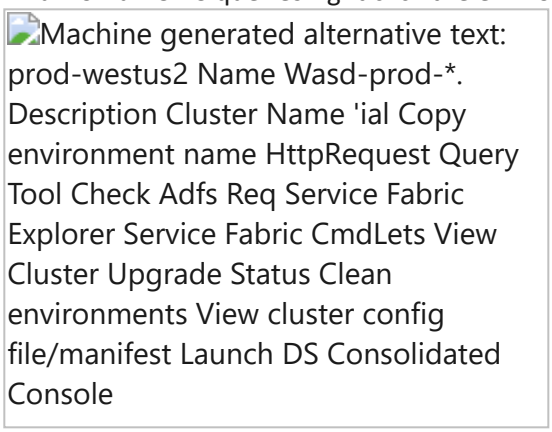
If you cannot find the replica in the above view the replica might already have been deleted or rolled back.
Even otherwise, use the MonManagement query to find the operation's that happened on primary or replica.
If step 1 did not give you the replica name, the operations_parameters in this result will give you the name of the replica server that was created.
You can again use this query to see if that replica server is later promoted or dropped.
It is very important to look at MonMangement to understand all the operations that are performed on the primary or replica servers.
It could be updateslo's, restarts, promote replica or even delete of replica server. Any of these operations can result in inactive replication slot on primary.

```
MonManagement
| where operation_parameters contains "primary server name" //or operation_parameters contains "replica server name"
| project originalEventTimestamp, request_id, ClusterName, operation_type , operation_parameters, operation_result, management_operation_name , action_name, error_code , error_message, error, message, stack_trace, old_state , new_state, ['keys'], exception_message, elasped_time
| order by originalEventTimestamp asc
```

(You can also use "Fast Server Browser.xts" to check if a server starts successfully.
In the "Server important Events" tab, you can see "PG starts" which means PG process starts to run, and "PG ready" means PG server is ready.)

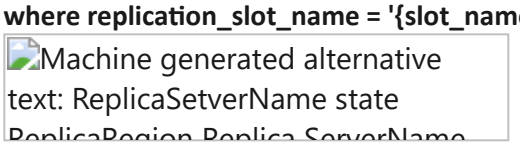
Also Check the dbo.elastic_servers_copy_links table from CMS by running the following query to get the replica server details and region

a. To Run CMS queries right click the environment (example westeu2) and select the **'HttpRequestQueryTool'**



Run the following query to get the replica server details

select partner_elastic_server_name as ReplicaServerName, partner_region as ReplicaRegion, State from elastic_server_copy_links
where replication_slot_name = '{slot_name}' and local_replication_role = 'Primary'



Note: If **state != 'Catchup'** the replica server was likely never "successfully" created, contiue to next step to see if it hit any error in sanbox during creation/startup.

6. **CHECK ANY SANDBOX RESTARTS FOR PRIMARY OR REPLICA**

Use the below query and run it with master and later replica server name in the appropriate regions to check the states of the master and replica servers.
This query will give information on if the servers restarted (because of restart, updateslo requests or because of some deployments, or crashes caused by corruptions etc)
Also if the time that streaming replication is broken aligns with the restart on a master or a replica, that gives some more context.
For example replica may not be coming up successfully and constantly restarting.
In this case we have to fix this issue first. Follow the process of fixing a continuously restarting server by looking at why it is failing. Elastic Server Restarts TSG

```
MonRdmsPgSqlSandbox
| where LogicalServerName == "<servername>"
| summarize min(originalEventTimestamp), max(originalEventTimestamp) by LogicalServerName ,
NodeName, AppName, ClusterName , process_id, code_package_version
| project LogicalServerName, min_originalEventTimestamp, max_originalEventTimestamp, NodeName,
AppName, ClusterName, process_id, code_package_version
| order by max_originalEventTimestamp asc
```

7. **CHECK SANDBOX FOR REPLICA TO PRIMARY CONNECTION ERRORS:**

   If there are no repeated/continuous restarts of primary or replica, then check the sandbox of the replica for the error why it is not able to connect to master.

Go to the *Kusto region of the ReplicaServer* and check for errors in the PGSandbox logs to determine cause of replication failures

```
MonRdmsPgSqlSandbox
| where TIMESTAMP > ago(1h)
| where LogicalServerName == "{replica server_name}"
| where text contains "error" or text contains "fail" or text contains "fatal" or text contains
"invalid lp" or text contains "streaming"
| project  originalEventTimestamp, event, text
| order by originalEventTimestamp asc
```

Based on the exception message perform the following mitigation

a. If you see any of the following errors

"FATAL: could not connect to the primary server: could not load private key file "replica.key": key values mismatch"

"FATAL: could not connect to the primary server: SSL error: tlsv1 alert **unknown ca"**

"FATAL: could not connect to the primary server: SSL error: sslv3 alert certificate expired"

"FATAL: could not connect to the primary server: Schannel: could not accept SSL connection: error 80090318"

"System.Exception: Certificate c:\FileShareDir\replica.key is empty."

FATAL: could not connect to the primary server: could not connect to server: Network is down

Query for above errors

```
MonRdmsPgSqlSandbox
| where LogicalServerName == "aware-production-04-westus"
| where text contains "could not connect to the primary server" or text contains
"replica.key is empty"
| project originalEventTimestamp , LogicalServerName, process_id, text
| order by originalEventTimestamp  desc
```

If you see any of these errors, mitigation is below

Mitigation:

To mitigate the issue, updateslo the REPLICA SERVER TO THE SAME SLO SIZE

Set-ElasticServer2 -ServerName REPLICA_NAME -SubscriptionId SUB -ResourceGroup RGNAME -ServerType PostgreSQL.Server.PAL -ServiceLevelObjective SLONAME

SLONAME will be of the format PGSQL_GP_GEN5_4 and you can get it from elastic server and database view in the SqlInstance table, if needed

Once the new service fabric app comes up, give it 20 mins or so and check the slot telemetry, it should come to active state.
Also if it has been in inactive state for a long time, replica might be catching up using file-shipping now. See step 3.D above and see if it is close to primary. Once it is done, it will enter streaming replication and slot will be active.

b. If you see the error about connection slots

"FATAL: could not connect to the primary server: FATAL: remaining connection slots are reserved for non-replication superuser connections"

- In this case check the vcore size of the server and its available max_connections at https://docs.microsoft.com/en-us/azure/postgresql/concepts-limits#:~:text=Maximum%20connections%20%20%20%20Pricing%20Tier%20,%20%201982%20%209%20more%20rows%20
- Check the number of active connections on the server using ASC portal
- If server is using up all the available connections, then replica does not have a connection to connect.

Mitigation: In this case usually once a connection is closed on the primary, replica will grab that connection since it is constantly retrying.
But incase connection limit is reached by client connections it wont get a chance.

If this is happening for a few hours, please contact customer and inform them to increase vcore size of primary to give it more connection limit

Or reduce their connections. Each REPLICA needs one connection to the primary.

c. "FATAL: could not connect to the primary server: FATAL: no pg_hba.conf entry for replication connection"
In this case there should have been some update slo on primary or replica server and the firewall rule in pg_hba.conf of the primary was not updated to reflect the current TR/APPNAME of the replica.

Mitigation:

- Get SAS token to primary file share of primary and open pg_hba.conf file
- Check the firewall rule at the top. There should be one rule for each replica.
- The rule has the TR name and APPNAME of the replica. Verify it matches where ever the replica is currently running.
- If not fix the rule with the right appname and tr name and upload the pg_hba.conf file to the file share. (you will need sastoken with read write access, "rwl" instead of "rl")

d. If you see the error like "PANIC: invalid lp on page LSN:D7/E21C0DA0"
or any other errors that are constantly crashing and restarting the replica

Then it means that the replica server has hit a corruption or some other serious error. Proceed to step 10.

8. If any of the cases in step 6, Verify Server is now streaming replicating successfully after you applied the corresponding mitigation

9. If none of the errors in step 8, check if replica is catching up using file shipping and has not yet entered streaming state where the replication slot will be active, like mentioned in 3.C above

```
This query is more handy for same region replicas to check file shipping replication status, so
that you can see both primary and replicas action at the same time.
Use queried in 3.D to see replica and primary different if they are in different regions
MonRdmsPgSqlSandbox
| where LogicalServerName == "replica_server_name" or where LogicalServerName ==
"primary_server_name"
| where text contains "restored transaction log" or text contains "archived transaction log"
| project originalEventTimestamp , LogicalServerName, process_id, code_package_version, text
| order by originalEventTimestamp  desc
```

10. If none of the above,
    a. **check if master or the replica are paused, by some on-call person for mitigation of earlier incidents.**
       For this go to SFE and navigate to the Worker.PAL.PG app for the master and the replica and check the SFE properties for the 'StartPause' property.
       If it is set to some value, it means the server is paused.

    If either the master or the replica was paused it explains why replication is broken.

    To understand why the server was paused, a good way to check for this is to search ICM by the master and replica server names and see if there are any incidents especially corruption related ones where the on-call might have paused the server to avoid repeated crashing of a corrupted server.

    If the master is corrupted and paused, there is no resolution until the master server is fixed. Assign the incident to PG-DRI.

    But if replica is corrupted and paused, proceed to step 11 to recreate the replica.

11. **Steps to recreate a replica server**

    If replica hit some corruption or unrecoverable error, or if it is lagging so far behind primary (more than 10000 wal files or so) that it is very hard to catchup, then

    Mitigation might be to recreate the replica with the same name after dropping the current one.
    The instructions for the same are at
    SOP0172 - Steps to delete and recreate PG Read Replica

    (FIRST, Please do step 2 in this above TSG first to remove any impact on the primary server
    SOP: How to delete a replication slot on a PG server)

How good have you found this content?