The storage started increasing "without reason"

Last updated by | Gaby Ramirez | Feb 9, 2021 at 11:14 AM PST

Contents

Issue

The storage started increasing "without reason" and the customer wants to know why.

You can start checking this **TSG:** Colocation and Performance Flexible Server

Investigation/Analysis

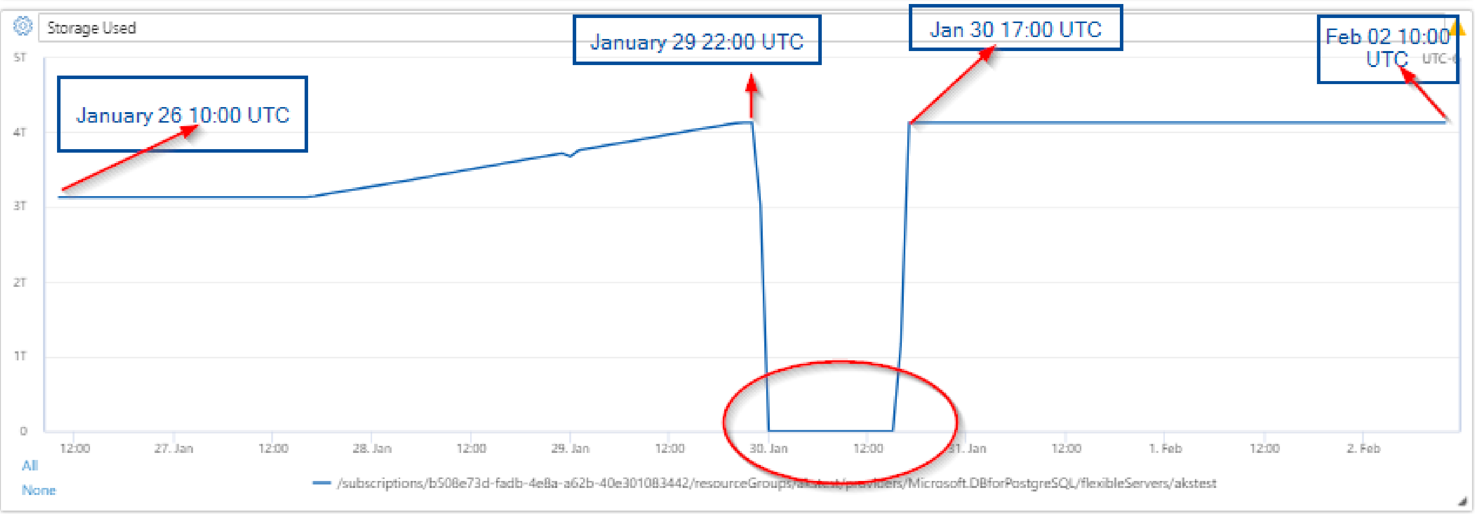
First, try to have SPECIFIC TIMESTAMP and TIMEZONE to get the exact logs when the issue happened or is happening.

Run *ASC* and check everything, IOPS, memory percentage, sandbox, operations, and document it in your notes. You can check here how many IOPS the server has based on the customer configuration.

After that you can check Jarvis Customer Facing Metrics. Definitely, you can check everything but try to focus on these,

- Storage used
- Write IOPS
- Transaction Logs Storage used

E.g. case, 121012324000230



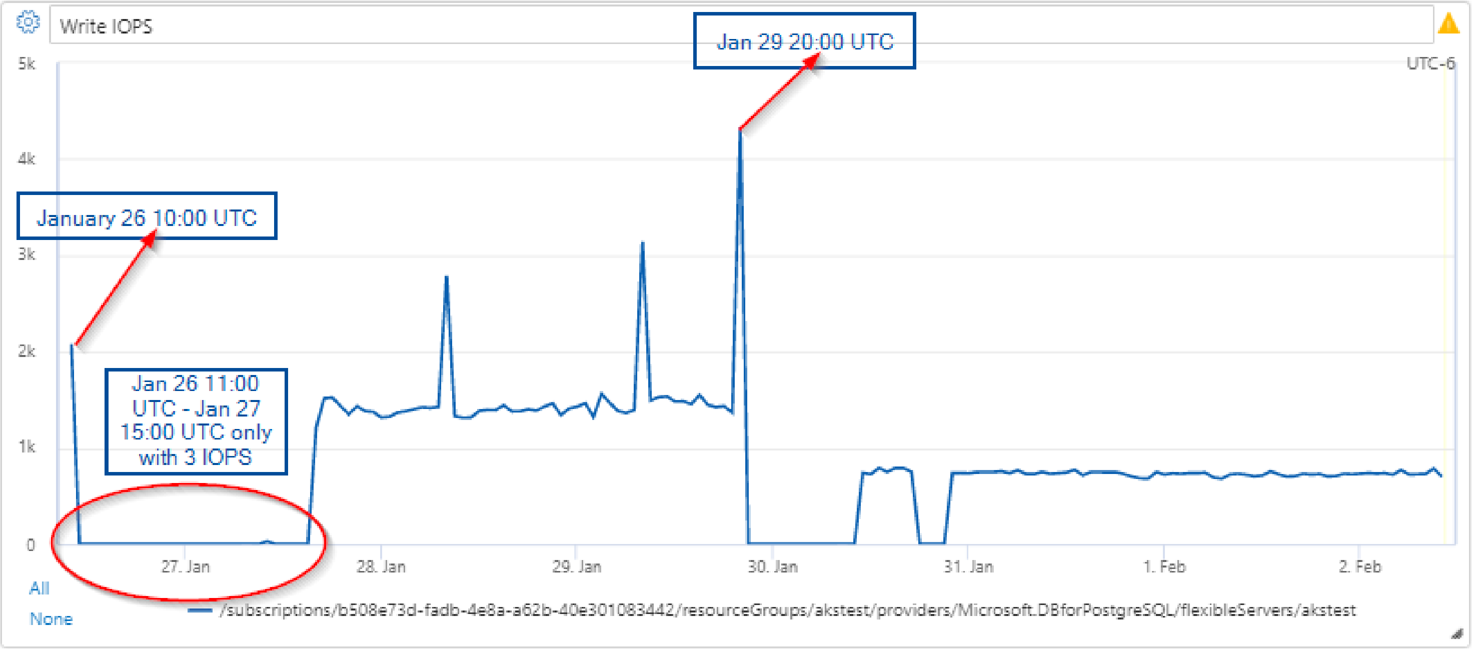
January 26 10:00 UTC

January 29 22:00 UTC

Jan 30 17:00 UTC

Feb 02 10:00 UTC

The storage used started to increase from (Jan 27- Jan 30 10:00 UTC) which seems lined up with the customer IOPS spike. In the below chart we can see write IOPS was in 2.08k, from (Jan 26 11:00 UTC – Jan 27 15:00 UTC) the server only consumed 3 IOPS, but after 15:00 UTC they started increasing and the highest peak was Jan 29 20:00 UTC, if you see in the previous screenshot the storage started increasing at the same TIMESTAMP, which means that someone was writing in the database and this triggered the storage increased.

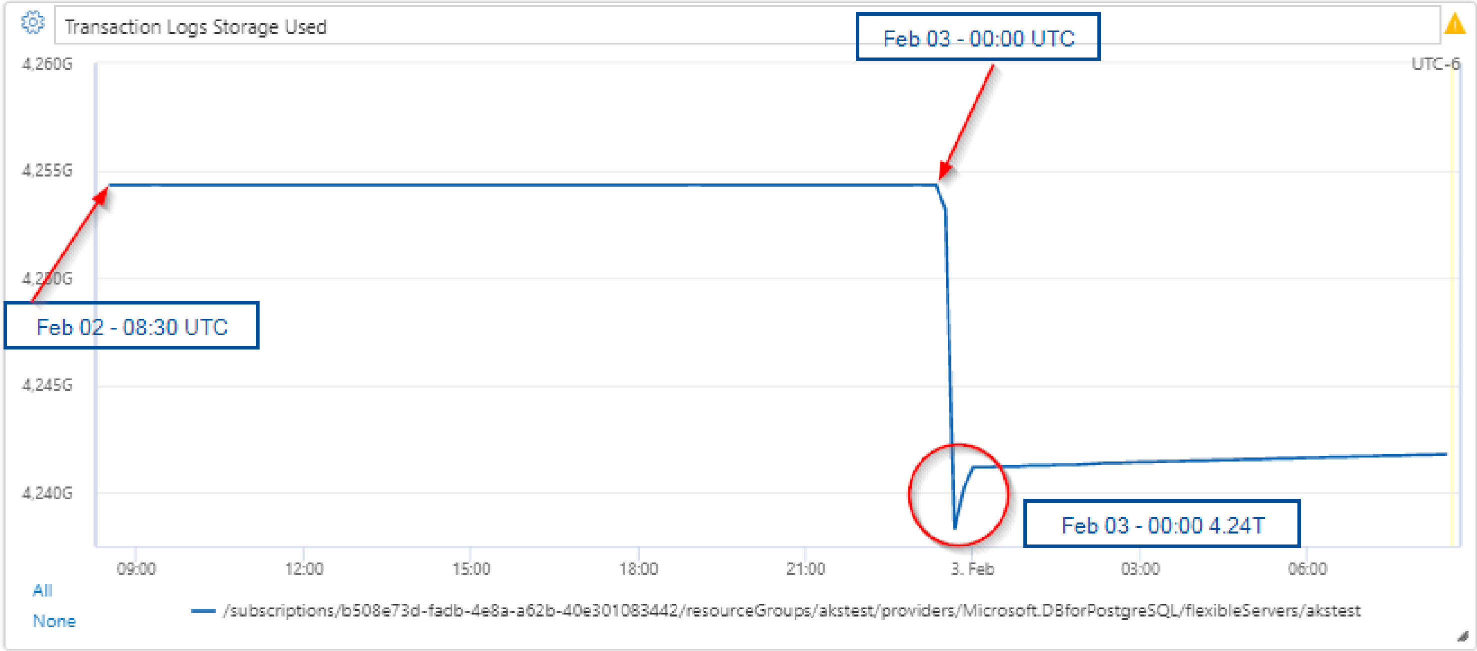


Jan 29 20:00 UTC

January 26 10:00 UTC

Jan 26 11:00 UTC - Jan 27 15:00 UTC only with 3 IOPS

In the last screenshot that I want to focus is in transaction logs storage used, as you can see transaction logs increased at the same TIMESTAMP when the storage started increasing too.



Feb 03 - 00:00 UTC

Feb 02 - 08:30 UTC

Feb 03 - 00:00 4.24T

The storage is used for the database files, temporary files, transaction logs, and the PostgreSQL server logs as you can see in our documentation ↗. Also, activity trigger storage increased and how metrics increase depends on how much data the customer inserts and other transactions.

From our backend, we notice transaction log was used largely portion of your storage.

You may ask to the customer if he ever used replication? If the customer says yes, you can provide the below query, because any unused replication slot must be dropped. If customer subscriber or consumer fails or has not been properly configured, the unconsumed logs will pile up and fill your storage.

SELECT * FROM pg_replication_slots;

Also you can run the below query to see *Sidecar logs* show 0 ready files.

OBvmagentsidecarpgsql

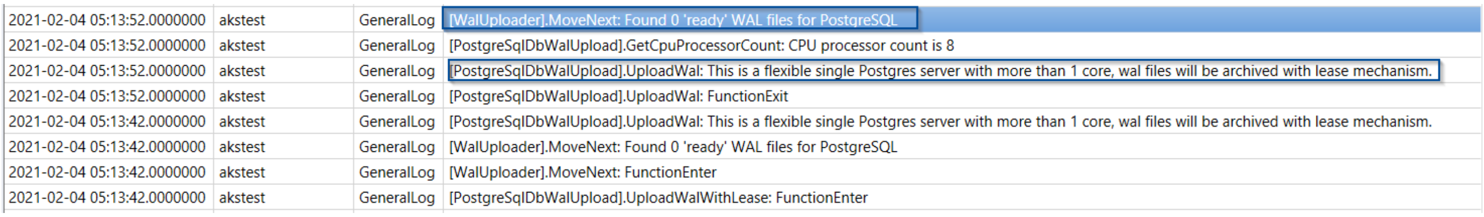
| where LogicalServerName =~ "akstest"

| where MessageString contains"Wal"

| order by TIMESTAMP desc

| project PreciseTimeStamp, LogicalServerName, LogType, MessageString

| take 10000



```
2021-02-04 05:13:52.0000000  akstest  GeneralLog  [WalUploader].MoveNext: Found 0 'ready' WAL files for PostgreSQL
2021-02-04 05:13:52.0000000  akstest  GeneralLog  [PostgreSqlDbWalUpload].GetCpuProcessorCount: CPU processor count is 8
2021-02-04 05:13:52.0000000  akstest  GeneralLog  [PostgreSqlDbWalUpload].UploadWal: This is a flexible single Postgres server with more than 1 core, wal files will be archived with lease mechanism.
2021-02-04 05:13:52.0000000  akstest  GeneralLog  [PostgreSqlDbWalUpload].UploadWal: FunctionExit
2021-02-04 05:13:42.0000000  akstest  GeneralLog  [PostgreSqlDbWalUpload].UploadWal: This is a flexible single Postgres server with more than 1 core, wal files will be archived with lease mechanism.
2021-02-04 05:13:42.0000000  akstest  GeneralLog  [WalUploader].MoveNext: Found 0 'ready' WAL files for PostgreSQL
2021-02-04 05:13:42.0000000  akstest  GeneralLog  [WalUploader].MoveNext: FunctionEnter
2021-02-04 05:13:42.0000000  akstest  GeneralLog  [PostgreSqlDbWalUpload].UploadWalWithLease: FunctionEnter
```

Public Doc Reference

- Monitoring
- Storage
- Backup Process in flexible server

ICM reference: 225597399

RCA

Customer will need to drop the replication slot or consume the slot changes so that the wal files can be removed form the pg_Wal driectory.