rubrik

11 Pitfalls of Native Cassandra Backup Tools

Cassandra (Apache OSS version, DataStax Enterprise Edition/DSE) is a NoSQL non-relational database that is becoming increasingly popular with the emergence of enterprise use cases such as customer 360, Internet of Things (IoT), and personalization.

One of the things to keep in mind is that you need data protection that was built from the ground up to incorporate zero trust data security principles to ensure your data is readily available when you need it most.

Although native backup tools exist, here are 11 pitfalls to avoid.

1. NODE-LEVEL SNAPSHOTS ARE NOT EQUAL TO DATABASE BACKUP

The method of backup in native solutions is node-by-node snapshots, which are essentially local snapshot backups **(i.e., hand-coded scripts)**. This is the most rudimentary solution, which means that it is not scalable and is error prone. Above all, this solution does not provide a point-in-time backup, so you cannot recover in the event of catastrophic data loss.

2. INCREASED BACKUP STORAGE COST

In a native backup solution, **all replica copies are kept in backup storage**. These backups are stored in the Cassandra nodes themselves as well as in optional secondary storage. In addition, there is no special handling of compacted sstables, which means that newly generated compacted sstables, are backed up in addition to the sstable from which the new sstables originated. All of this translates into increased backup storage costs.

3. LONGER RECOVERY TIMES WITH REPAIRS UPON RECOVERY

Scripted solutions require an enormous amount of manual effort to restore the data, and node-by-node restore from backup storage increases recovery time and network traffic. In addition, you need to restore all replicas during recovery, which increases the time it takes for restore. But, it isn't just about the length of time; it's also the hidden cost of restore. After restoring the data, **DBAs** need to run cluster-wide repairs to bring the cluster to a consistent state.

4. LACK OF ANY POINT-IN-TIME RECOVERY (APIT)

Enterprises frequently need to refresh their test and development clusters with the latest production data to enable Continuous Integration (CI) and Continuous Development (CD). However, these clusters have different topologies **(number of nodes)** than production database clusters. It takes hours, if not days to refresh each cluster using native solutions—this leads to loss of developer productivity.

5. NO DATA MASKING OPTION DURING RECOVERY

Native tools do not give you the option to mask out certain columns during recovery of confidential data such as personally identifiable information **(PII)**. This has large implications for enterprises that handle sensitive data, including name, address, phone number, and social security number.

6. LACK OF FAILURE-HANDLING SUPPORT DURING BACKUP/RECOVERY OPERATIONS

In a native solution, if a source node fails during backup operations, the backups for that node stop. This can result in data loss or a large amount of inconsistency in a backedup dataset. This is a significant limitation in any large-scale production environment in which nodes can fail often. In fact, you need your backup solution to perform most when failures occur.

7. LACK OF SUPPORT OF TIME-TO-LIVE HANDLING

There is no ability to adjust **Time-to-Live (TTL)** during restores. This means that if TTL is already expired during recovery, restored data is automatically expired by Cassandra.

8. LIMITED SUPPORT FOR BACKUP STORAGE TARGETS

Native tools are limited to choosing the local filesystem or Amazon Simple Storage Service (Amazon S3) as backup storage targets. There is no option to store backups to other **"S3 compatible"** object storage providers. Furthermore, there is no option to store backups to Google Cloud Storage or object storage targets for on-premises deployments.

9. LIMITED PROTECTION GRANULARITY LEVEL

In native solutions, only keyspace-level backup is available. There is no flexibility to back up using column-family level. This means that all column families in a keyspace will be backed up using the same policy **(backup frequency and retention)**. Additionally, column families that are not needed, but are in the same keyspace will be backed up, as well.

10. LOWER PERFORMANCE

Native solutions use sequential data streams for both backup and recovery, as opposed to using parallel and distributed data movement.

11. LIMITED DATA MANAGEMENT USE CASES

Native solutions do not give you the ability to restore to a different cluster with a different topology. What this means is that you cannot use the same cluster for restore to **QA/Dev/Test** clusters of different topology/capacity.

rubrik

Global HQ
3495 Deer Creek Road
Palo Alto, CA 94304
United States

1-844-4RUBRIK
inquiries@rubrik.com
www.rubrik.com

brf-eleven-pitfalls-of-native-cassandra-backup-tools / 20230913