Apache Cassandra Rapid Node Replacement Using Snapshots

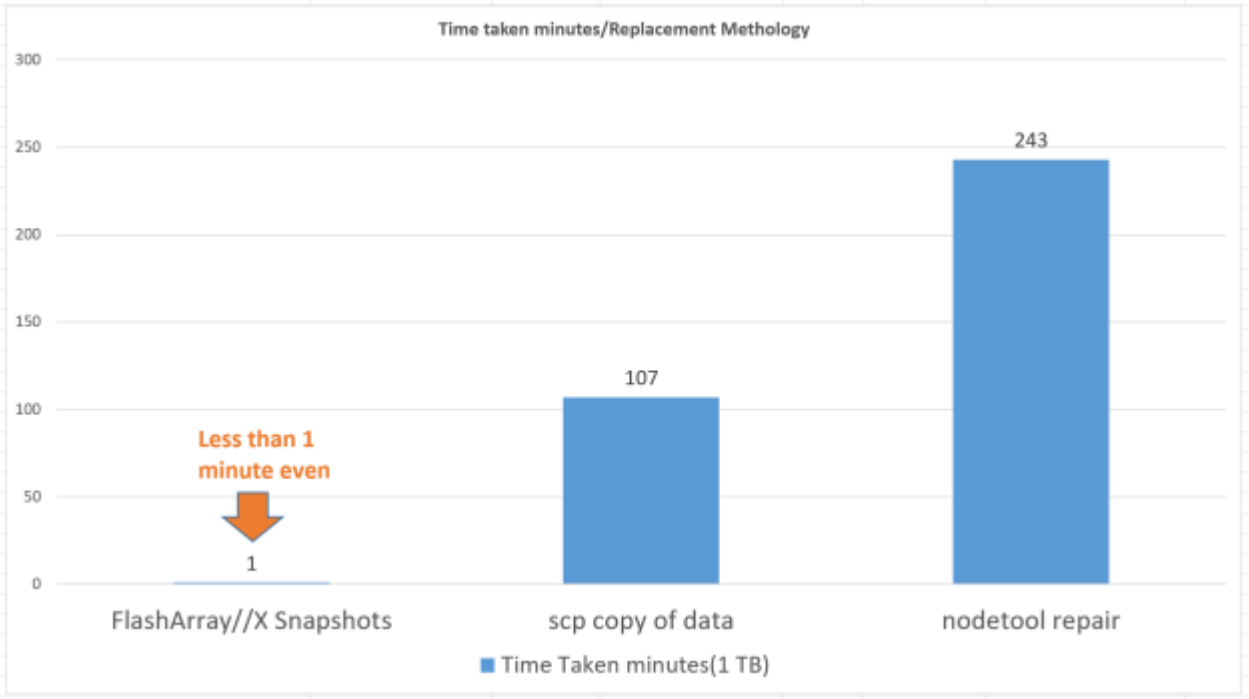
In this blog, I am going to showcase replacement of a node up for maintenance or a failed node in Apache Cassandra cluster by a new healthy node using Pure Storage FlashArray//X snapshots. Apache Cassandra administrators use either node repair process or have to copy the data from the failed node to a healthy new node. This is a very slow and painful process and can also slow down your Apache Cluster considerably which in turn affects your business transactions. This process can be simplified using the FlashArray snapshots. This is not the only advantage of snapshots for Apache Cassandra, in the previous blog I have shown how snapshots can be used to build dev/test clusters or cluster copy. In my next blog, I am going to showcase how to do backup and recovery of Apache Cassandra cluster using snapshots.

Why use FlashArray//X snapshots for Apache Cassandra node replacement?

There are three ways we can do the Apache Cassandra node replacement:

1. **Using nodetool repair:** When a Cassandra node is down, it needs to start taking care of the writes it missed. The hints are stored in an attempt to inform a node of missed writes and aren't guaranteed to inform a node of 100% of the writes it missed. These inconsistencies can cause data loss as nodes are replaced or tombstones expire. So we use nodetool repair so that these inconsistencies are fixed with the repair process. The repair synchronizes the data between nodes by comparing their respective datasets for their common token ranges and streaming the differences for any out of sync sections between the nodes. It compares the data by creating merkle trees, which are a hierarchy of hashes. This process is extremely slow and takes lots of time depending on the data it needs to transfer to the new node from the other nodes in the cluster.
2. **Scp or rsync:** Using scp or rsync to copy all the data from the node to be replaced to new node. Using scp has been faster to do the initial transfer of the data from old Cassandra node to the new Cassandra node. The scp copy has been 5-7 times faster to transfer than rsync.
3. **Pure Storage FlashArray//X's instant snapshots:** Now the most efficient method or fastest method is using FlashArray//X snapshots. Pure storage FlashArray//X snapshots are instantaneous and initially do not consume any additional disk space. Hence the failed node replacement process is accelerated using Pure Storage FlashArray//X snapshots.

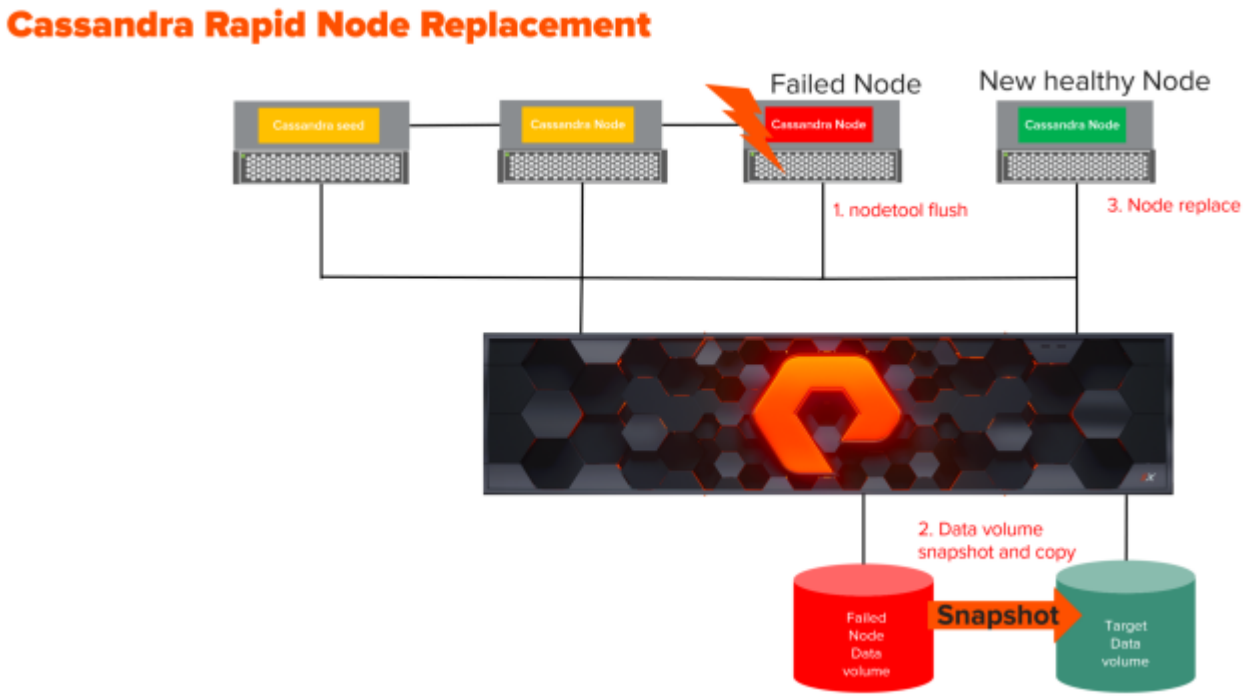
We can see the time comparison as below between these three different methodologies. I have tested with 1 TB of data to compare these different methodologies.



Clearly, it shows that FlashArray//X snapshot process is instantaneous even when data grows or multiplies. Whereas other methodologies(scp or repair) will get even slower if there is more data per node.

Apache Cassandra node replacement process:

To test the Rapid replacement from the failed node, I have created a three-node Apache Cassandra cluster.



As seen in the nodetool status , the cluster has three nodes 67, 69, 70. Now I am going to fail one of the

nodes, in this case, it is node .67 and bring up a healthy node and attach node .68 to the cluster. There is a keyspace with replication factor 3 on the cluster and populated it with data.

```
[root@node2 mapper]# nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address       Load       Tokens       Owns (effective)  Host ID                               Rack
UN  10.21.238.69  101.3 GiB  256          33.7%             6a1cecd2-941f-4822-adeb-aa05aaa44e7f  rack1
UN  10.21.238.70  100.72 GiB  256          33.2%             8ad1593e-404a-493b-9995-2a2b33ddb0b1  rack1
UN  10.21.238.67  100.46 GiB  256          33.1%             7fbc24b1-5f90-4cc9-b56a-48ec042a92b7  rack1
```

Here are the steps for node replacement

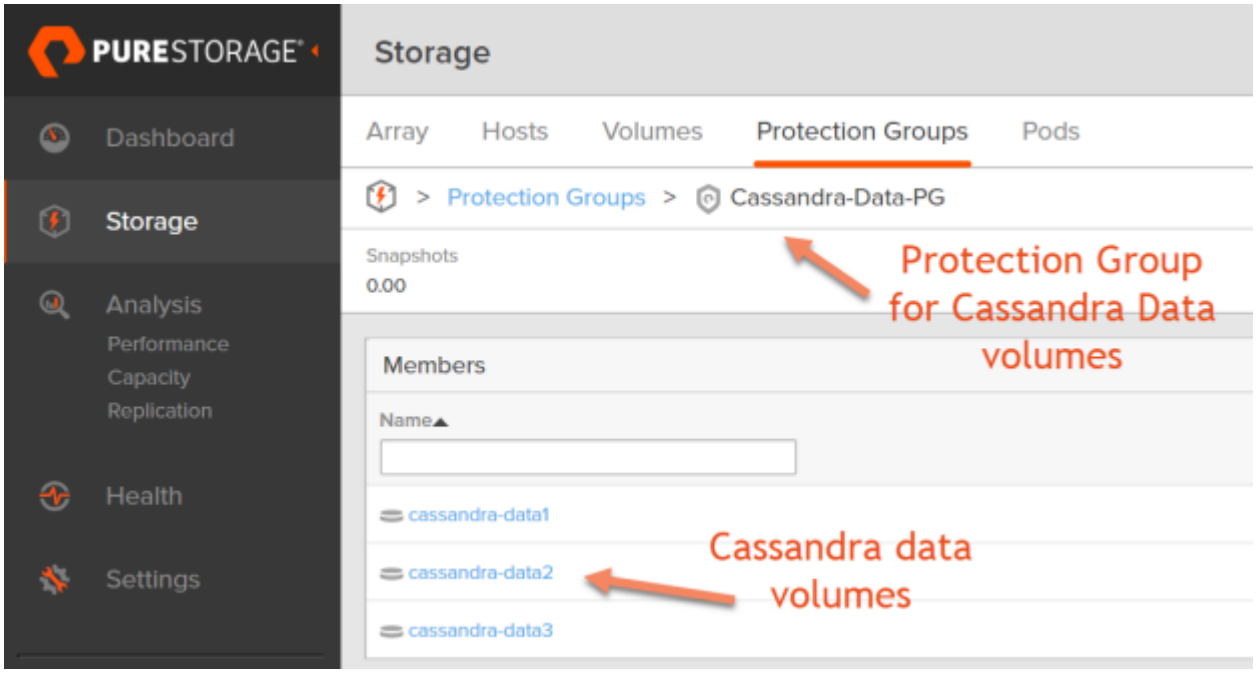
1. On the Node which is going to be replaced, Run nodetool flush or nodetool drain. The difference between them is nodetool drain flushes memtables to SSTables on disk and then it stops listening for connections from the client and other nodes. The command nodetool drain is used when you want to upgrade the node. Where as nodetool flush just flushes the memtables to SSTables on disk.

   Run nodetool flush or nodetool drain.

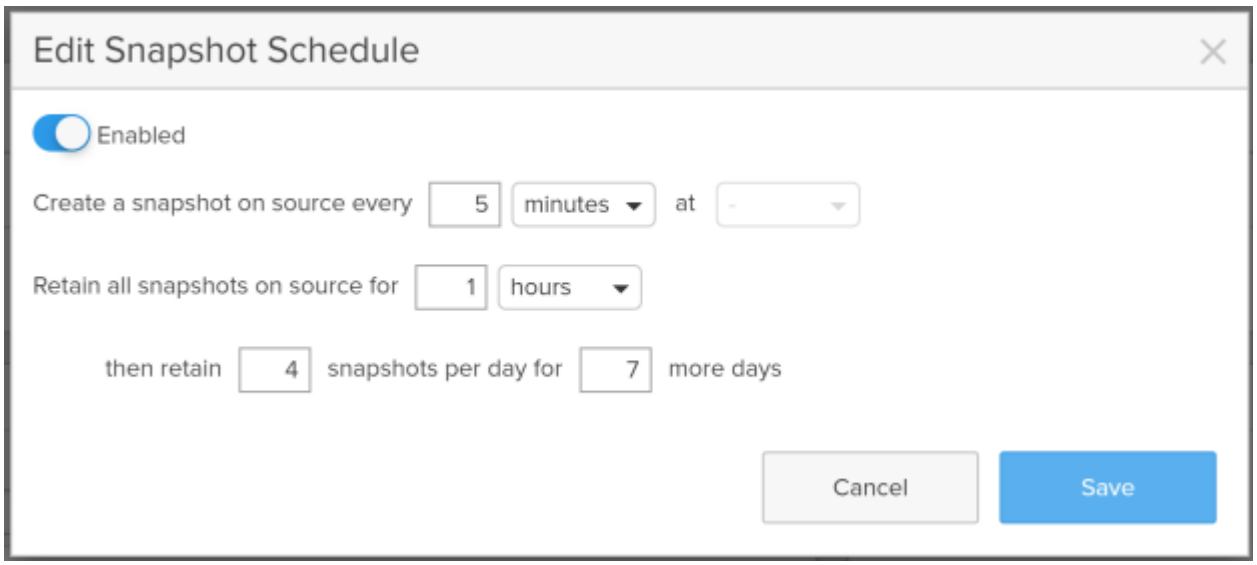
2. Freeze the filesystem for data volume in Cassandra node which is going to be replaced. The command is

   xfs_freeze -f /var/lib/cassandra/data

3. Next step is to take FlashArray//X snapshot. The recommendation is to create a protection group for the Cassandra data volumes as shown below.



Take the FlashArray snapshot of all the commit log volumes in the protection group for every 5 minutes schedule. The snapshots which are more than 1 hour old will be removed automatically by setting up retention.

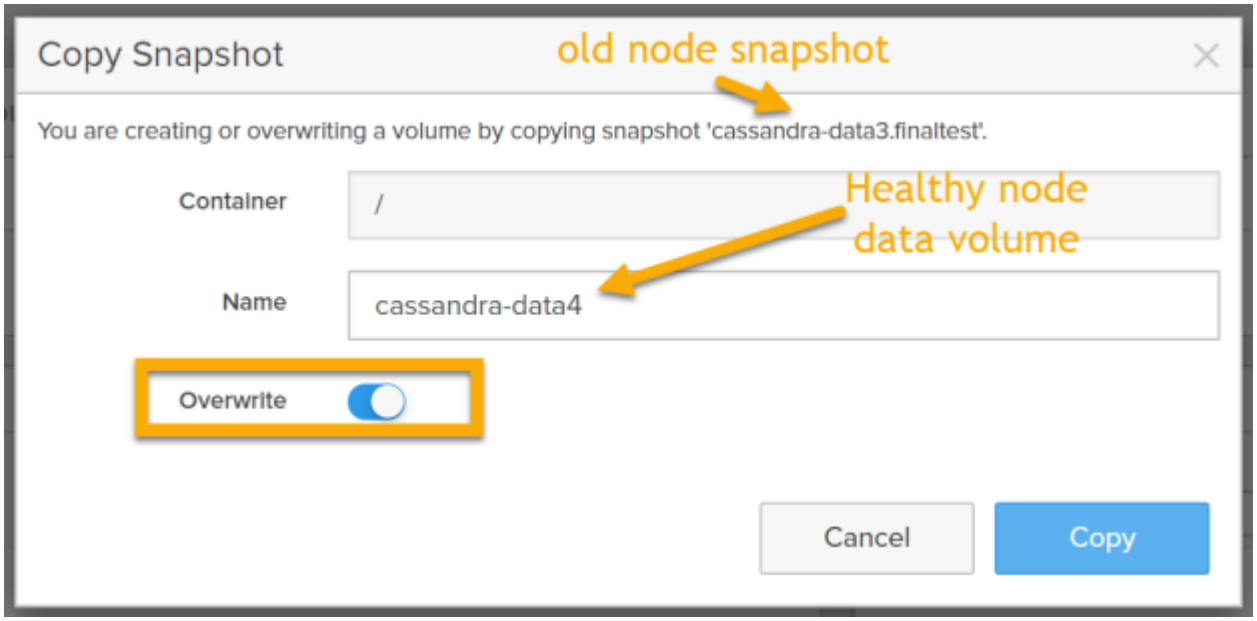


4. Unfreeze the filesystem for data volumes in Cassandra node which is going to be replaced. The command is

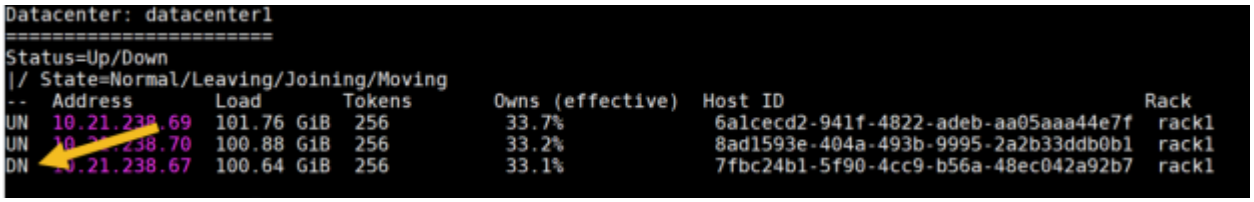
xfs_freeze -u /var/lib/cassandra/data

5. Install Cassandra on the new healthy node which is going to be added to the cluster. Change the configuration in cassandra.yaml to enable it to join the cluster. Copy the latest

snapshot from the failed node or the node which is going to be replaced.



Make sure the node which is replaced in down as shown below.

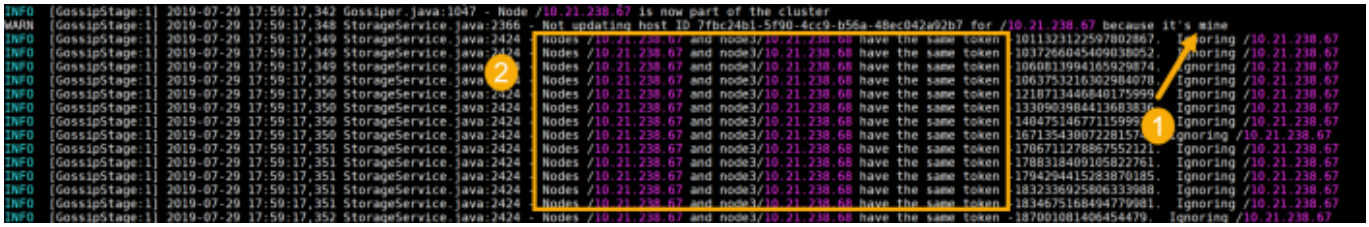


Run the following command on this new healthy node:

service cassandra start -Dcassandra.replace_address_first_boot=10.21.238.67

You will see the new node joins immediately to the cluster and we can see the following on the system.log

1. This new node will say it is not updating the host ID for 10.21.238.67 because it is **mine**
2. It will immediately start validating the tokens and confirming it has the same tokens. This drastically reduces the nodetool repair time even if some changes were done when this node was added to the cluster. This is an instantaneous process when there were no changes during this time.



This shows how easy it is to use Pure Storage FlashArray//X snapshots to replace Cassandra nodes.