Reviewing data buckets retrieved during restore job

You ran a Dynamic Data Active Archive (DDAA) restore job to thaw data for a one-month period. The job status shows that it finished successfully and a big number of GBs of data has been restored. But when you search for the data for that month, search results provide a small number of events, only 3,000. You expect a large number of events for this size of data retrieved. You suspect that the data has not been retrieved properly and want to investigate.

Check for error logs related to the DDAA restoration job.

Run the following search for the period when the DDAA restore job has been performed:

```
index=_internal source IN (*splunk_archiver_restoration.log*, *restoration.log*, *python.log*)
| search "CRITICAL" OR "ERROR" OR "FAILURE" OR "WARN*" OR "FATAL"
```

If there are any error logs found, review them accordingly, in the context of the DDAA restore job ID.

Check the index size of events restored in the DDAA job.

Run the following `dbinspect` search for DDAA period restored:

```
| dbinspect index=<INDEX>
| eval rawSizeGB=rawSize/1024/1024/1024
| stats sum(rawSizeGB) by index
```

The expected result is a larger volume of data compared to the one listed in the DDAA job (buckets replication in indexer cluster).

Check the number of characters counted for the events retrieved in the restoration period.

Run the following search for DDAA period restored:

```
index=<INDEX>
| eval CharCount=len(_raw)
| stats sum(CharCount)
```

Multiply the result by 4 (1 character = 4 bytes max). Compare the product with the raw size of events listed in the `dbinspect` raw summary in the previous search. The expected result is that the `dbinspect` size of raw data will be larger. If the number of bytes is somewhere near the number of the raw data (that will always be larger due to index replication), you have a potential explanation of the situation.
Review buckets structure retrieved in the DDAA job.

Run the following search for DDAA period restored:

```
| dbinspect index=<INDEX>
| eval _time=endEpoch
| rename bucketId AS bid
| search bid=<INDEX>~*~*
| join bid
| search earliest=-30d latest=now() index=_internal source=*splunkd* <INDEX> <INDEX>~*~* "Skipping the restored bucket since it is still required by current restoration requests"
| stats count by bid
| eval source="not removing" }
| eval duration=(endEpoch-startEpoch)*1000
| eval bucket_number=mvindex(split(bid,"~"),1) |dedup bucket_number
| table startEpoch bid duration
```

If the index contains very wide buckets, quarantine buckets, you have an explanation why so much data has been retrieved in the DDAA job. When you restore the data from Splunk, you restore all buckets that contain the data for the period you listed in the restore job. By default, buckets are being rolled to warm state, after a trigger is being met (size of the bucket). When buckets are being rolled to a warm state, they have epoch time assigned (startEpoch and end epochs) - based on the earliest timestamp and last timestamp of events saved in a bucket. When an event you want to index (save to Splunk) cannot be assigned to a currently opened hot bucket (eg. it is not in the current time range, for a reason like an event timestamp is from the future or the past), it is saved in a quarantine bucket that will collect this kind of event.

Check whether the data has been made unsearchable with the 'delete' command.

Retrieved buckets contain '/deletes' folders ($SPLUNK_DB/<index>/db/<bucket_id>/deletes/).

Next steps

The content in this article comes from the Splunk Support Knowledge Base, one of the thousands of Splunk resources available to help users succeed. In addition, these Splunk resources might help you understand and implement this use case:

- Splunk Docs: [How Dynamic Data Active Archive works](link)
- Splunk Docs: [Buckets and indexer clusters](link)
- Splunk Docs: [Search Reference: dbinspect](link)