Fixing Corrupted Bucket Runbook

Table of Contents

Document Information

Department	Cloud Ops
Applies To	SRE
Category	Ops Runbook
Author(s)	@ Ravi Nandasana @ Kaushal Hirani @ Kanaiya Tarapara (C) (Deactivated)
Primary SME(s)	@ Ravi Nandasana @ Kaushal Hirani @ Kanaiya Tarapara (C) (Deactivated)
Required Restarts	☐ Search Head ☐ Cluster Master ☑ Indexers (Rolling) ☐ License Master
Customer Impacting?	Yes
Status	IN PROGRESS
Last Updated	03 Jan 2022

Context

This runbook illustrates a non-standard process for resolving corrupt buckets in a Cloudworks environment, which can directly impact the customer's search capabilities.

Repair Corrupt Bucket

Important
This runbook is not used for cloud stacks as of now. Please use official RB - Troubleshooting Corrupt Buckets as per this thread - https://splunk.slack.com/archives/GMGDD53T3/p1645606546058509?thread_ts=1645442014.944649&cid=GMGDD53T3

Pre-checks

Important
Repeat all below mention steps for each requested index individually.

1. Check Requested index is present.

2. Run the below command on the indexer to check the corrupted bucket for a specific index and save the output of this command in **<index-name>.txt** your local machine (Give index name as a Filename) **(Modify Index name in below Command)**

   FSCK Scan

   ```
   splunkd fsck scan --all-buckets-one-index --index-name=<index-name>
   ```

   Note: **Run the below command to scan all buckets for all indexes**

   > **Scan for all indexes**
   > splunk fsck scan --all-buckets-all-indexes

3. Find bucket paths for corrupted buckets by running the below command on your local terminal. **(Modify Filename in below Command)**

   ```
   pr -tms" " <(pr -t -m -s~ <(cat <index-name>.txt | grep -B 2 -i "corruption" --color | grep idx | awk -F "[= ]" '{print $4}') <(cat <index-name>.txt | grep -B 1 -i "corruption" --color | grep -i "bucket=" | awk -F "bucket=" '{print $2}' | awk -F "_" '{print $(NF - 1) "~" $NF}')) <(cat <index-name>.txt | grep -B 1 -i "corruption" | grep -i "bucket=" | awk -F "bucket=" '{print $2}') > bucket_ids_paths_<index-name>.txt
   ```

   > **Explanations of above command**
   > The above command will find the bucket paths and bucket IDs and store them into bucket_ids_paths_<index-name>.txt file

4. Find bucket IDs for corrupted buckets by running the below command on your local terminal. **(Modify Filename in below Command)**

   ```
   pr -t -m -s~ <(cat <index-name>.txt | grep -B 2 -i "corruption" --color | grep idx | awk -F "[= ]" '{print $4}') <(cat <index-name>.txt | grep -B 1 -i "corruption" --color | grep -i "bucket=" | awk -F "bucket=" '{print $2}' | awk -F "_" '{print $(NF - 1) "~" $NF}') > bucket_ids_<index-name>.txt
   ```

   > **Explaination**
   > The above command will find the bucket IDs of all corrupted buckets and store them in file bucket_ids_<index-name>.txt.

5. SSH in c0m1 and perform Splunk login using the below command

   Splunk Login

   ```
   splunk login
   ```

6. Copy **bucket_ids_<index-name>.txt file** (corrupted bucket IDs which is output of step 4) from local machine to **/opt/splunk/tmp/TO-XXXXX/bucket_ids_<index-name>.txt** then run below command on **c0m1** **(Modify Filename in below Command)**

Find Primary Buckets

```
cat bucket_ids_<index-name>.txt | xargs -I % /bin/bash -c 'splunk
search "| rest splunk_server=local /services/cluster/master/buckets
/% | table title primaries_by_site.site0 | rename
primaries_by_site.site0 as primary_guid | join [search
index=_introspection earliest=-15m | stats count by host
splunk_server data.instance_guid | rename data.instance_guid as
primary_guid] | rename host AS primary_indexer, title AS bucket_id
| table primary_indexer,bucket_id" | tail -1' >>
primary_buckets_<index-name>.txt
```

Explanations of above command
This command will run a search query on c0m1 to find the primary indexer of all buckets and store it in file primary_buckets_<index-name>.txt

error while finding primary indexer
If you face any error while finding a primary indexer then mention those buckets in JIRA. Also, give a list of these buckets to the customer and inform the customer that we can not repair them

Example

7. Copy **primary_buckets_<index-name>.txt (generated from step 6)** from **c0m1** to **local machine**.
8. Run the below command to get the list of primary indexers, bucket paths, and bucket IDs.

Command

```
join -1 2 -2 1 primary_buckets_<index-name>.txt
bucket_ids_paths_<index-name>.txt | awk '{print "|" $2 "|" $3 "|"
$1 "|"}' | sort > all_details_<index-name>.txt
```

Explaination
Document the content of file all_details_<index_name>.txt in Jira.

9. Check if rawdata is present in the bucket or not. by running the below python script in your local machine

Script to check rawdata

```
import os

bucket_to_recover = ""
bucket_to_not_recover = ""
bucket_manual_check = ""
FAIL_COLOR = '\033[91m'
```

```
ENDC_COLOR = '\033[0m'
SUCCESS_COLOR = '\033[92m'
WARNING_COLOR = '\033[93m'
OKBLUE_COLOR = '\033[94m'
HEAD_COLOR = '\u001b[34m'

print("Example Path = /Users/rnandasana/Downloads/all_details_main.
txt")
location = input("Enter absolute path of all_details_<index_name>.
txt file = ")

with open(location,'r') as f:
    for lines in f:
        bucket = lines.strip().split("|")
        idx = bucket[1].split(".")[0]
        path = bucket[2]
        print("===========")
        print(idx)
        print(path)
        print(" ")
        cmd = "sft ssh " + idx + " --command 'sudo -u splunk sh -c
\"cd /opt/splunk/; ls -la " + path + " ; hostname -f ; date\"'"
        op = os.popen(cmd).read()
        #print("op output" + op + "close" )
        if op:
            if "rawdata" in op.lower():
                bucket_to_recover = bucket_to_recover + lines
                print(SUCCESS_COLOR + "\n rawdata is present for -
" + path + ENDC_COLOR)
            else:
                bucket_to_not_recover = bucket_to_not_recover +
lines
                print(WARNING_COLOR + "\n rawdata is not present
for - " + path + ENDC_COLOR)
        else:
            bucket_manual_check = bucket_manual_check + lines
            print(FAIL_COLOR + "\n Connection is Unsuccessful for
Indexer - " + idx + ENDC_COLOR)
        print("===========")

print(" ")
print(HEAD_COLOR + "Paste this output in JIRA" + ENDC_COLOR)
print("============================================")
print(OKBLUE_COLOR + "Below buckets are present with rawdata." +
ENDC_COLOR)
print(SUCCESS_COLOR + bucket_to_recover + ENDC_COLOR)
print("======================")
print(OKBLUE_COLOR + "Below buckets are not present with rawdata."
+ ENDC_COLOR)
print(WARNING_COLOR + bucket_to_not_recover + ENDC_COLOR)
print("======================")
print(OKBLUE_COLOR + "Check below buckets manually." + ENDC_COLOR)
print(FAIL_COLOR + bucket_manual_check + ENDC_COLOR)
print("============================================")
```

Example

Execution

1. Add Downtime
2. Enable maintenance Mode on c0m1
3. Stop Splunk on the instance indexer

Stop Splunk

```
sudo puppet agent --disable "Your Ticket"
sudo systemctl stop splunk
sudo su - splunk
```

4. Verify that rawdata directory is present or not on Primary IDX by using the above script (Precheck Section - Step 9)
5. Repair bucket using below command

Repair Bucket

```
cd /opt/splunk/bin
./splunk rebuild <path_to_bucket>
```

6. If the rebuild fails then use the below command

Workaround

```
gunzip <path_to_bucket>/rawdata/journal.gz

gzip <path_to_bucket>/rawdata/journal

./splunk rebuild <path_to_bucket>
```

7. Start Splunk on the instance

Commands

```
sudo puppet agent --enable
sudo systemctl start splunk
```

8. Repeat the above steps for all bucket

Post-checks

1. Verify the **corrupted buckets** again using the below command

Commands

```
splunkd fsck scan --all-buckets-one-index --index-name=<index-name>
```

2. Verify that SF/RF is met and All data is searchable.
3. Verify that Maintenance mode is disabled.
4. Perform general post checks

Escalation

If you have any issues with this process, please escalate to SRE using the process defined in the Cloud Escalation Policy.

References

TO-125303 - Fix corrupted buckets on - cape **CLOSED**

TO-119726 - apollo - Fix corrupted buckets (restore the networkops index buckets) **CLOSED**