Title: Migrating from Self-Managed Elastic Stack to Elastic Cloud using Snapshot and Restore with Google Cloud Storage

Introduction: Migrating from a self-managed Elastic Stack deployment to Elastic Cloud offers various advantages, including reduced operational burden, enhanced scalability, and access to managed services. A crucial aspect of this migration involves seamlessly transferring data while minimizing downtime. This document outlines the process of migrating Elasticsearch indices using snapshot and restore functionality, leveraging Google Cloud Storage (GCS) as the intermediary storage solution.

Prerequisites:

- Access to the self-managed Elastic Stack deployment.
- Access to an Elastic Cloud deployment.
- A Google Cloud Platform (GCP) account with Google Cloud Storage (GCS) configured and accessible.
- Kubernetes command-line tool (kubectl) configured to interact with the Kubernetes cluster.

Procedure:

1. Obtain Elasticsearch Secret:

- Retrieve the Elasticsearch password secret from the Kubernetes secret:

  because we donot have kibana pod running on kubernetes cluster. So we need to do all activites through curl. For this need elastic user and password.

  elastic user: elastic

  for password use below command..

  kubectl get secret ec-deploy-es-elastic-user -o=jsonpath='{.data.elastic}' | base64 --decode; echo

#Command to list indices

- curl -k -u elastic:4S392wm9po05JIE7abVl0zi2 -X GET "https://localhost:7979/_cat/indices"

  so you can verify that indices once it storaged in elastic cloud.

2. Copy Secret to Pod:

- Copy the gcp service account key for bucket to Elasticsearch pod. For sending indices to gcp bucket.

  kubectl cp secret.json ec-deploy-es-default-0:/usr/share/elasticsearch/data/secret.json

  here:

  pod name is “ ec-deploy-es-default”
  secret file is “secret.json”

  path location to in pod is “/usr/share/elasticsearch/data/secret.json”

3. Add Secret to Elasticsearch Keystore:

- Add the secret to the Elasticsearch keystore within the pod:

elasticsearch-keystore add-file gcs.client.default.credentials_file /usr/share/elasticsearch/data/secret.json

so GKE elastic search can use this key for communication..

you need to run this in /bin directory

4. Reload Secure Settings:

- Reload the secure settings in Elasticsearch to apply the changes:

  curl -k -X POST -u elastic:<ELASTIC_PASSWORD>
  "https://localhost:9200/_nodes/reload_secure_settings" -H "Content-Type: application/json" -d
  '{"secure_settings_password": ""}'

  because it is need to restart the pod. but once i restart it secret file will be lost. that is why we use the above commad to this..

5. Connect Snapshot Repository to GCS:

- Configure the snapshot repository to point to Google Cloud Storage:

curl -k -X PUT -u elastic:<ELASTIC_PASSWORD> -H "Content-Type: application/json"
https://localhost:9200/_snapshot/disearch-backup -d '{ ---> here we have repo name disearch-backup

"type": "gcs",

"settings": {

"bucket": "disearch_k8_es_db_backup", ----> this is bucket name

"base_path": "disearch-stag-backup", ---> this is bucket folder name where data bill be store

"compress": true,

"client": "default"

}

}'

use “port-forward” for reverse proxy..

6. Verify Repository Status:

- Confirm that the snapshot repository is create or not:

curl -k -X GET -u elastic:<ELASTIC_PASSWORD> "https://localhost:9200/_snapshot/disearch-backup/_status"

7. Create Snapshot Lifecycle Management (SLM) Policy:

- Define an SLM policy to automate snapshot creation:

curl -k -X PUT -u elastic:<ELASTIC_PASSWORD> -H "Content-Type: application/json"
https://localhost:9200/_slm/policy/disearch-backup -d '{ here is the policy name

"name": "<diseachsnap_{now/d}>", ----------this is indices name

"schedule": "0 0 0 1 1 ?",

"repository": "disearch-backup", ----------this repository name

"config": {

"include_global_state": true,

"feature_states": []

}

}'

9. Execute Policy:

- Manually trigger the SLM policy execution if needed:

curl -k -X PUT -u elastic:<ELASTIC_PASSWORD> "https://localhost:9200/_slm/policy/disearch-backup/_execute"

#get policy

curl -k -X GET -u elastic:4S392wm9po05JIE7abVl0zi2 "https://localhost:9200/_snapshot/_slm/policy/"