Resizing ElasticSearch Storage w/o dataloss

Christian Tawfik, 9. 11. 2021

APA TECH

Die APA-Tech

- IT-Tochter der APA – Austria Presse Agentur
- Dienstleistungen
  - Hosting
  - Housing
  - Streaming
  - E-Paper Lösungen
  - Softwareentwicklung

- https://www.apa-tech.at

https://apa.at/about/always-on-apa-tech/

Über mich

- Christian Tawfik
  - Application Engineer @ APA-Tech < 2009
  - Linux Systemmanager @ APA-Tech < 2012
  - Docker / Container Enthusiast < 2015
  - OpenShift Architect / ClusterAdmin < 2020

  - christian.tawfik@apa.at
  - https://twitter.com/toughiq
  - https://github.com/toughiq
  - https://www.linkedin.com/in/christian-tawfik-a48405/

    (Oder einfach meinen Namen in der LinkedIn Suche eingeben…der ist so speziell, dass wird schon passen )

TL;DR

- https://access.redhat.com/solutions/6075191
- ElasticSearch Status Commands
  - https://gist.github.com/toughIQ/ae299dd8ac4dffed4387c92f89bdf5a8

Danke für eure Aufmerksamkeit!

Worüber reden wir?

- OpenShift 4.7.x
- NetApp/Trident Block Storage Backend
- OpenShift Logging Stack
- Redhat OpenShift Logging Operator
- ElasticSearch Cluster
- Day2 and DayX Operations

OpenShift Logging Stack - EFK

- ElasticSearch - *Store*
- FluentD - *Collect*
- Kibana – *Display*

- Operatoren
  - OpenShift Elasticsearch Operator
    - https://github.com/openshift/elasticsearch-operator
  - Red Hat OpenShift Logging
    - https://github.com/openshift/cluster-logging-operator

- Installationsanleitung
  - https://docs.openshift.com/container-platform/4.7/logging/cluster-logging-deploying.html

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  collection:
    logs:
      fluentd: {}
      type: fluentd
  curation:
    curator:
      nodeSelector:
        node-role.kubernetes.io/infra: ''
      schedule: 30 3 * * *
    type: curator
  logStore:
    elasticsearch:
      nodeCount: 3
      nodeSelector:
        node-role.kubernetes.io/infra: ''
      redundancyPolicy: SingleRedundancy
      resources:
        limits:
          memory: 16Gi
        requests:
          cpu: 8
          memory: 16Gi
      storage:
        size: 500Gi
        storageClassName: MyBlockStorage
    retentionPolicy:
      application:
        maxAge: 7d
      audit:
        maxAge: 7d
      infra:
        maxAge: 7d
    type: elasticsearch
  managementState: Managed
  visualization:
    kibana:
      nodeSelector:
        node-role.kubernetes.io/infra: ''
      replicas: 1
    type: kibana
```

Warum überhaupt?

- Diskspace der ElasticSearch Nodes wird knapp
  - Längere Behaltedauer der Logs
  - Mehr Apps als ursprünglich erwartet
  - Debugging Logs in diversen Namespaces
- ElasticSearch/Logging Performance ist eingeschränkt
  - Watermarks werden erreicht
  - FluentD Queues steigen an
  - Kibana Views sind langsam bzw. verzögert
- **Verlust der Logdaten ist keine Option**

Was man klassisch tut …

- Redhat Knowledge Base-Artikel
  - https://access.redhat.com/solutions/5233001
- Redhat OpenShift Support
- Support verweist auf den KB-Artikel … und täglich grüßt …

- Logische Fehler, technische Hürden, Typos …
- **Logging Operator verhindert manuelle Eingriffe!**
- Wissen über Cluster Systeme und ElasticSearch Resilienz
- Schritt für Schritt Weiterentwicklung
- Endergebnis
  - https://access.redhat.com/solutions/6075191

Die Voraussetzungen … zumeist Default

- 3-Node ElasticSearch Cluster
- Operator managed
- PVC Block Storage
- ElasticSearch redundancyPolicy != ZeroRedundancy
  - *SingleRedundancy*
  - MultipleRedundancy
  - FullRedundancy
- Die Grundlagen
  - https://docs.openshift.com/container-platform/4.7/logging/config/cluster-logging-log-store.html

Anleitung – 101 Version

1. ClusterLogging ElasticSearch Storage Größe neu setzen

   ```
   oc edit clusterloggings.logging.openshift.io instance -n openshift-logging
   ```

2. Elasticseach Instanz stoppen

   ```
   oc scale deployment elasticsearch-cdm-<HASH>-X --replicas=0 -n openshift-logging
   ```

3. Entsprechendes Storage entfernen

   ```
   oc delete pvc elasticsearch-elasticsearch-cdm-<HASH>-X -n openshift-logging
   ```

4. Neuerstellung des Storages mit neuer Größe durch Operator abwarten
5. ElasticSearch Instanz starten

   ```
   oc scale deployment elasticsearch-cdm-<HASH>-X --replicas=1 -n openshift-logging
   ```

6. Überwachung des ElasticSearch Clusters: 100% Verfügbarkeit bzw. Status GREEN
7. Zurück zu Schritt 2 für die nächste ElasticSearch Instanz
8. Fertig = alle ElasticSearch Instanzen wurden sequenziell gestoppt, Storage gelöscht und wieder gestartet.

Die Grundlagen …

```
$ oc project openshift-logging
Now using project "openshift-logging" on server "https://api.ocp.yourdomain.com:6443".
```

```
$ oc get deployment -l component=elasticsearch
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
elasticsearch-cdm-<HASH>-1   1/1     1            1           26h
elasticsearch-cdm-<HASH>-2   1/1     1            1           26h
elasticsearch-cdm-<HASH>-3   1/1     1            1           26h
```

```
$ oc get pod -l component=elasticsearch
NAME                                        READY   STATUS    RESTARTS   AGE
elasticsearch-cdm-<HASH>-1-76bbd48d8c-tlgqg   2/2     Running   0          52m
elasticsearch-cdm-<HASH>-2-675947f7d4-vfk24   2/2     Running   0          39m
elasticsearch-cdm-<HASH>-3-795d4dc9c-dnwc4    2/2     Running   0          25m
```

```
$ oc get pvc
NAME                                     STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
elasticsearch-elasticsearch-cdm-<HASH>-1   Bound    pvc-5e6de991-16f3-442d-b803-2222332294ed   95368Mi    RWO            thin           54s
elasticsearch-elasticsearch-cdm-<HASH>-2   Bound    pvc-32000594-0b35-45ea-888e-ba726642e45e   95368Mi    RWO            thin           25h
elasticsearch-elasticsearch-cdm-<HASH>-3   Bound    pvc-06189790-6a1b-4177-b0e7-5b302acdf4b8   95368Mi    RWO            thin           25h
```

Schritt 1 – Das Storage anpassen …

```
$ oc edit clusterloggings.logging.openshift.io instance
     spec:
     [...]
     logStore:
        elasticsearch:
            nodeCount: 3
           nodeSelector:
               node-role.kubernetes.io/infra: ""
           redundancyPolicy: SingleRedundancy
           storage:
             size: 200Gi
           storageClassName: MyBlockStorage
```

Schritt 2 - Das Kochrezept …

```
$ oc scale deployment elasticsearch-cdm-<HASH>-X --replicas=0
deployment.apps/ elasticsearch-cdm-<HASH>-1 scaled
```

```
$ oc delete pvc elasticsearch-elasticsearch-cdm-<HASH>-X
persistentvolumeclaim "elasticsearch-elasticsearch-cdm-<HASH>-1 " deleted
```

```
$ oc get pvc
NAME                                         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
elasticsearch-elasticsearch-cdm-<HASH>-1     Bound    pvc-5e6de991-16f3-442d-b803-2222332294ed   143052Mi   RWO            thin           54s
elasticsearch-elasticsearch-cdm-<HASH>-2     Bound    pvc-32000594-0b35-45ea-888e-ba726642e45e   95368Mi    RWO            thin           25h
elasticsearch-elasticsearch-cdm-<HASH>-3     Bound    pvc-06189790-6a1b-4177-b0e7-5b302acdf4b8   95368Mi    RWO            thin           25h
```

```
$ oc scale deployment elasticsearch-cdm-<HASH>-1 --replicas=1
deployment.apps/ elasticsearch-cdm-<HASH>-1 scaled
```

```
$ watch oc exec -c elasticsearch elasticsearch-cdm-<HASH>-1-<PODid> -- es_util --query=_cat/health?v
Tue Jun 29 14:55:29 UTC 2021
epoch      timestamp cluster       status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1624978529 14:55:29  elasticsearch yellow          3         3     24  17    0    0       10             0                  -                 70.6%

#####
epoch      timestamp cluster       status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1624978529 14:55:29  elasticsearch green           3         3    168  84    0    0        0             0                  -                100.%
```

Serviervorschlag …

```
#!/bin/bash

echo "Switching to Logging project"
oc project openshift-logging

# get PODid
es_pod=$(oc get pod --selector=component=elasticsearch --no-headers -o jsonpath='{range
.items[?(@.status.phase=="Running")]}{.metadata.name}{"\n"}{end}' | head -n1)

echo "### Health ###"
oc exec -c elasticsearch $es_pod -- es_util --query=_cat/health?v

echo "### Nodes ###"
oc exec -c elasticsearch $es_pod -- es_util --query=_cat/nodes?v

echo "### Utilization ###"
oc exec -c elasticsearch $es_pod -- curl -s --key /etc/elasticsearch/secret/admin-key --cert
/etc/elasticsearch/secret/admin-cert --cacert /etc/elasticsearch/secret/admin-ca
https://localhost:9200/_cat/allocation?v
```

- Diese und mehr ElasticSearch Status Abfragen
  - https://gist.github.com/toughIQ/ae299dd8ac4dffed4387c92f89bdf5a8

Fragen und Antworten …

- **Eure Fragen …**
  - Bitte gerne alles und jederzeit

- Themen, die ich anregen kann:
  - Warum *kann* die originale Redhat Lösung nicht funktionieren?
  - Mögliche Lösung für Multi-Node mit ZeroRedundancy?
  - JSON Logging
  - OpenShift 4.8 und 4.9 (Stichwort APIs)

  - Welche Dance-Moves sind gerade auf TikTok aktuell?

Vielen Dank für eure Aufmerksamkeit!

+43 1 36060-6415
christian.tawfik@apa.at
www.apa.at