Ezmeral Unified Analytics 1.5: Addressing ETCD excessive size usage and kube-apiserver memory issues

Table of contents

Issue

This article describes two different problem scenarios:

- **Excessive memory usage by kube-apiserver -** A customer reported the memory usage of the `kube-apiserver` steadily increasing until it occupied over 90% of the system's memory, causing node outage and UI accessibility issues.
- **ETCD Database excessive space consumption -** The ETCD database grew to approximately **4.5 GB**, much larger than the optimal size for the platform, leading to potential performance bottlenecks.

Observed using the `top` command on all master nodes:

```
top -p $(pgrep kube-apiserver)
```

Example output:

```
PID   USER PR NI VIRT  RES   SHR S %CPU %MEM TIME+    COMMAND
15310 root 20 0  93.6g 29.5g 0   S 24.2 94.0 47703:55 kube-apiserver
```

Environment

EZUA 1.5

Cause

- **Memory Leak in kube-apiserver:** The `kube-apiserver` process had an unaddressed memory leak, leading to steady memory consumption after node reboots.
- **Excessive ETCD space utilization:** A significant number of unnecessary ETCD entries were found, especially for `kyverno.io`, contributing to database bloat. This is a known kyverno issue https://github.com/kyverno/kyverno/issues/6462
  - **Kyverno Admission Reports:** The accumulation of `admissionReports` in Kyverno was a major factor causing excessive ETCD entries.
  - Found an excessive number of `kyverno.io` entries using:

```
ETCDCTL_API=3 etcdctl --cacert="/etc/kubernetes/pki/etcd/ca.crt" \
--cert="/etc/kubernetes/pki/apiserver-etcd-client.crt" \
--key="/etc/kubernetes/pki/apiserver-etcd-client.key" \
--endpoints="127.0.0.1:2379" \
get /registry --prefix --keys-only | grep -v ^$ | awk -F '/' '{ h[$3]++ } END {for (k in h) print h[k],

1157882 kyverno.io 11053 events 3644 services
```

- **Kyverno Admission Reports:**
  - Large accumulation of `admissionReports` causing excessive ETCD storage.

Resolution

- **ETCD Cleanup:** Removed unnecessary ETCD entries related to `kyverno.io` and performed defragmentation and compaction on all ETCD nodes to reclaim disk space. The database size was reduced from **4.5 GB** to approximately **250 MB**.
- **Kyverno Configuration Update:** Disabled `admissionReports` in the `kyverno-admission-controller` deployment to prevent further accumulation of unwanted data in ETCD.
- **Kube-API Server Restart:** Restarted the `kube-apiserver` on all nodes, resolving the UI accessibility issue and stabilizing the system.

1. ETCD Cleanup

- **Backup the ETCD database:**

```
ETCDCTL_API=3 etcdctl etcdctl --cacert="/etc/kubernetes/pki/etcd/ca.crt" \
  --cert="/etc/kubernetes/pki/apiserver-etcd-client.crt" \
  --key="/etc/kubernetes/pki/apiserver-etcd-client.key" \
  --endpoints="127.0.0.1:2379" \ snapshot save /tmp/backup.db on all master nodes
```

- **Set env variables of all master etcd nodes to access the cluster from a single master node**

```
export ETCDCTL_API=3
export ETCDCTL_CACERT="/etc/kubernetes/pki/etcd/ca.crt"
export ETCDCTL_CERT="/etc/kubernetes/pki/etcd/server.crt"
export ETCDCTL_KEY="/etc/kubernetes/pki/etcd/server.key"
export ETCDCTL_ENDPOINTS="https://<etcd-node1>:2379,https://<etcd-node2>:2379,https://<etcd-node3>:2379"
```

- **Then run Delete excessive `kyverno.io` entries:**

```
etcdctl del /registry/kyverno.io/admissionreports --prefix
```

2. ETCD Compaction & Defragmentation

- **Get current revision number:**

```
ETCDCTL_API=3 etcdctl --cacert="/etc/kubernetes/pki/etcd/ca.crt" \
  --cert="/etc/kubernetes/pki/apiserver-etcd-client.crt" \
  --key="/etc/kubernetes/pki/apiserver-etcd-client.key" \
  --endpoints="127.0.0.1:2379" \
  endpoint status --write-out="json" | egrep -o '"revision":[0-9]*' | egrep -o '[0-9].*'
```

  - Example output:
    `12567890`

- **Run compaction on the latest revision on all etcd nodes starting from leader following by slave nodes:**

```
ETCDCTL_API=3 etcdctl --cacert="/etc/kubernetes/pki/etcd/ca.crt" \
  --cert="/etc/kubernetes/pki/apiserver-etcd-client.crt" \
  --key="/etc/kubernetes/pki/apiserver-etcd-client.key" \
  --endpoints="127.0.0.1:2379" \ compact 12567890
```

- **Defragment ETCD on all nodes:**

```
ETCDCTL_API=3 etcdctl --cacert="/etc/kubernetes/pki/etcd/ca.crt" \
  --cert="/etc/kubernetes/pki/apiserver-etcd-client.crt" \
  --key="/etc/kubernetes/pki/apiserver-etcd-client.key" \
  --endpoints="127.0.0.1:2379" \
  defrag  --dial-timeout 10s --command-timeout 500s
```

- **Verified database size reduced from ~4.5GB to ~250MB.**

```
ETCDCTL_API=3 etcdctl --endpoints=etcd-node1:2379, etcd-node2:2379, etcd-node3:2379 \   --cacert=/etc/kubernete
```

3. Kyverno Configuration Update

- **Disable admissionReports in `kyverno-admission-controller` deployment:**
  `kubectl edit deployment kyverno-admission-controller -n kyverno`

  - Set the flag to `false` under `args`:
    `--admissionReports=false`

4. Restart `kube-apiserver`

- **Restarted kube-apiserver on all nodes:**
  `systemctl restart kube-apiserver`

5. Validate System Stability

- **Check `kube-apiserver` memory usage again:**
  `top -p $(pgrep kube-apiserver)`

- **Confirm all ETCD endpoints are healthy:**
  `ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379`
  _____
  _____
  _____
  _____ \
  `--cert /etc/kubernetes/pki/etcd/server.crt \`
  `--key /etc/kubernetes/pki/etcd/server.key \`
  `--cacert /etc/kubernetes/pki/etcd/ca.crt \`
  `endpoint health`