

# Sentry Kafka Queue Reset

## TLDR

Generally this has occurred after a number of pods have failed. This has lead to a build up of unconsumed topics. The build up has then been so great that the consumption rate is to slow to a point that an error state has been created. To remedy this we need to remove the unconsumed topic queue.

Install of K9s will be useful and a useful tool to invest sometime in learning [K9s - Manage Your Kubernetes Clusters In Style](#)

1. Determine which queue needs to be reset, usually the associated deploy / consumer has an error:
  - a. You need to ensure you are in the correct namespace. Sentry is held in the `sentry` namespace and so as when using `vi` you use `:` to initiate a command in K9s. So `:namespace` will show those available.

```
Context: gke_staging-266717_europe-west2-a_staging
Cluster: gke_staging-266717_europe-west2-a_staging
User: gke_staging-266717_europe-west2-a_staging
K9s Rev: v0.32.5
K8s Rev: v1.29.9-gke.1177000
CPU: 7%
MEM: 48%

:namespace

NAME      STATUS      AGE
airflow   Active      3y199d
all       Active      467d
dagster   Active      4y273d
default   Active      4y140d
gitlab-runner   Active      59d
gke-managed-system   Active      452d
kafka     Active      4y223d
kube-node-lease   Active      4y273d
kube-public   Active      4y273d
kube-system   Active      4y273d
```

- b. You can now move down to `sentry` and click `enter`

```
Context: gke_staging-266717_europe-west2-a_staging
Cluster: gke_staging-266717_europe-west2-a_staging
User: gke_staging-266717_europe-west2-a_staging
K9s Rev: v0.32.5
K8s Rev: v1.29.9-gke.1177000
CPU: 5%
MEM: 48%

Pod(sentry)[55]

NAME                                PF READY STATUS  RESTARTS CPU MEM %CPU/R %MEM/L IP NODE
sentry-billing-metrics-consumer-754f6d7c9b-c4xvd  ● 1/1 Running      1 10 138 n/a n/a n/a 10.4.4.46 gke-staging-node-pool-6-49302d32
sentry-clickhouse-0                      ● 1/1 Running      0 39 430 n/a n/a n/a 10.4.16.157 gke-staging-node-pool-6-c49a8c27
sentry-clickhouse-1                      ● 1/1 Running      0 78 680 n/a n/a n/a 10.4.4.91 gke-staging-node-pool-6-49302d32
sentry-clickhouse-2                      ● 1/1 Running      0 33 512 n/a n/a n/a 10.4.0.34 gke-staging-node-pool-6-c49a8c27
sentry-cron-675546dfc-59mfw              ● 1/1 Running      3 3 195 n/a n/a n/a 10.4.16.190 gke-staging-node-pool-6-c49a8c27
sentry-ingest-consumer-attachments-86875bb79-v9x99 ● 1/1 Running      0 4 151 n/a n/a n/a 10.4.0.35 gke-staging-node-pool-6-c49a8c27
sentry-ingest-consumer-events-6cbd9957d9-g4z92    ● 1/1 Running      2 3 155 n/a n/a n/a 10.4.16.159 gke-staging-node-pool-6-c49a8c27
sentry-ingest-consumer-transactions-7dcf56bdd5-szk9q ● 1/1 Running      3 11 153 n/a n/a n/a 10.4.20.108 gke-staging-node-pool-6-49302d32
sentry-ingest-metrics-consumer-perf-f65f7947f-brftw ● 1/1 Running      3 6 296 n/a n/a n/a 10.4.4.69 gke-staging-node-pool-6-49302d32
sentry-ingest-metrics-consumer-rh-674cdf989-2w5bj ● 1/1 Running      3 7 300 n/a n/a n/a 10.4.15.93 gke-staging-node-pool-6-c49a8c27
sentry-ingest-monitors-5bd9b76644-gf7c7           ● 1/1 Running      0 4 160 n/a n/a n/a 10.4.4.42 gke-staging-node-pool-6-49302d32
sentry-ingest-occurrences-79d955d5d-bt2rh         ● 1/1 Running      0 5 320 n/a n/a n/a 10.4.16.116 gke-staging-node-pool-6-c49a8c27
sentry-ingest-replay-recordings-654876f6f8-8l69c  ● 1/1 Running      2 3 139 n/a n/a n/a 10.4.4.47 gke-staging-node-pool-6-49302d32
```

- c. You can now enter `:deploy` and view the sentry deployments and here we can see in this example `sentry-post-process-forward-errors` and `sentry-snuba-consumer` both have errors

```

Tilix: ildemundo@steroid:~/src/Raft/backend
Context: gke_staging
Cluster: gke_staging-266717_europe-west2-a_staging
User: gke_staging-266717_europe-west2-a_staging
K9s Rev: v0.31.5
K8s Rev: v1.26.13-gke.1052000
CPU: 10%
MEM: 20%

<=> all      <ctrl-d> Delete      <shift-f> Port-Forward
<1> sentry   <d> Describe         <r> Restart
<2> default  <e> Edit            <s> Scale
               <?> Help           <f> Show PortForward
               <l> Logs           <y> YAML
               <p> Logs Previous

Deployments(sentry) [29]
NAME                                     READY    UP-TO-DATE    AVAILABLE    AGE
sentry-billing-metrics-consumer          1/1      1              1            1 35d
sentry-cron                              1/1      1              1            1 175d
sentry-ingest-consumer-attachments        1/1      1              1            1 35d
sentry-ingest-consumer-events             1/1      1              1            1 35d
sentry-ingest-consumer-transactions        1/1      1              1            1 35d
sentry-ingest-metrics-consumer-perf        1/1      1              1            1 35d
sentry-ingest-metrics-consumer-rh          1/1      1              1            1 35d
sentry-ingest-monitors                     1/1      1              1            1 35d
sentry-ingest-occurrences                  1/1      1              1            1 35d
sentry-ingest-replay-recordings             1/1      1              1            1 35d
sentry-issue-occurrence-consumer            1/1      1              1            1 35d
sentry-post-process-forward-errors           0/1      1              0            0 35d
sentry-post-process-forward-issue-platform  1/1      1              1            1 35d
sentry-post-process-forward-transactions     1/1      1              1            1 35d
sentry-relay                               1/1      1              1            1 35d
sentry-sessions-consumer                   1/1      1              1            1 35d
sentry-shuba-vol                           1/1      1              1            1 175d
sentry-shuba-consumer                      0/1      1              0            0 35d
sentry-shuba-outcomes-consumer              1/1      1              1            1 35d
sentry-shuba-replacer                      1/1      1              1            1 35d
sentry-shuba-replays-consumer               1/1      1              1            1 35d
sentry-shuba-subscription-consumer-events    1/1      1              1            1 35d
sentry-shuba-subscription-consumer-sessions  1/1      1              1            1 35d
sentry-shuba-subscription-consumer-transactions  1/1      1              1            1 35d
sentry-shuba-transactions-consumer           1/1      1              1            1 35d
sentry-subscription-consumer-events          1/1      1              1            1 35d
sentry-subscription-consumer-transactions    1/1      1              1            1 35d
sentry-web                                  1/1      1              1            1 175d
sentry-worker                              2/2      3              3            3 175d

```

### kubectl option:

```
kubectl get deployments -n sentry
```

2. We now need to work in the sentry kafka. To navigate here we can view and execute into here by first finding this.

a. `:statefulset` and `enter` will show all and you can then select `sentry-kafka`

```

x k9s (k9s)
Context: gke_staging-266717_europe-west2-a_staging
Cluster: gke_staging-266717_europe-west2-a_staging
User: gke_staging-266717_europe-west2-a_staging
K9s Rev: v0.32.5
K8s Rev: v1.29.9-gke.1177000
CPU: 8%
MEM: 55%

<=> all      <ctrl-d> Delete      <shift-f> Port-Forward
<1> sentry   <d> Describe         <r> Restart
<2> kube-public  <e> Edit            <s> Scale
<3> default  <?> Help           <f> Show PortForward
               <l> Logs           <y> YAML
               <p> Logs Previous

statefulset

Statefulsets(sentry) [8]
NAME                                     READY    SERVICE                                     AGE
sentry-clickhouse                      3/3      sentry-clickhouse-headless                 342d
sentry-kafka                           3/3      sentry-kafka-headless                     418d
sentry-rabbitmq                         3/3      sentry-rabbitmq-headless                   418d
sentry-sentry-postgresql-hl             1/1      sentry-sentry-postgresql-hl                342d
sentry-sentry-redis-master               1/1      sentry-sentry-redis-headless               418d
sentry-sentry-redis-replicas             3/3      sentry-sentry-redis-headless               418d
sentry-zookeeper                       1/1      sentry-zookeeper-headless                  418d
sentry-zookeeper-clickhouse              3/3      sentry-zookeeper-clickhouse-headless        418d

```

b. You need to then get a shell on one of the pods. Press `s` to do this.

```

k9s (k9s)
Context: gke_staging-266717_europe-west2-a_staging
Cluster: gke_staging-266717_europe-west2-a_staging
User: gke_staging-266717_europe-west2-a_staging
K9s Rev: v0.32.5
K8s Rev: v1.29.9-gke.1177000
CPU: 8%
MEM: 55%

<ctrl-a> Attach <ctrl-k> Kill <ctrl-o> Show Node
<ctrl-d> Delete <ctrl-l> Logs <ctrl-f> Show PortForward
<ctrl-e> Describe <ctrl-p> Logs Previous <ctrl-t> Transfer
<ctrl-s> Edit <ctrl-q> Port-Forward <ctrl-y> YAML
<ctrl-h> Help <ctrl-z> Sanitize
<ctrl-j> Jump Owner <ctrl-s> Shell

Pod(sentry/sentry-kafka) [3] <app.kubernetes.io/component=kafka,app.kubernetes.io/>
NAME+ PF READY STATUS RESTARTS CPU MEM %CPU/R %CPU/L %MEM/R %MEM/L IP NODE AGE
sentry-kafka-0 ● 1/1 Running 0 21 1668 1 n/a 27 n/a 10.4.39.9 gke-staging-node-pool-6-c49a8c27-jdr6 8d
sentry-kafka-1 ● 1/1 Running 0 17 1165 0 n/a 18 n/a 10.4.20.37 gke-staging-node-pool-6-49302d32-ffl8 7d23h
sentry-kafka-2 ● 1/1 Running 0 26 1351 1 n/a 22 n/a 10.4.39.29 gke-staging-node-pool-6-c49a8c27-jdr6 8d

<statefulset> <pod>

```

c. You will be greeted with a command line for you work in the kafka cluster

```

k9s (kubectl)
<k9s-Shell> Pod: sentry/sentry-kafka-0 Container: kafka
I have no name!@sentry-kafka-0:/$

```

### Kubectl option:

First we find the statefulsets

```
kubectl get statefulsets -n sentry
```

1	NAME	READY	AGE
2	sentry-clickhouse	3/3	342d
3	sentry-kafka	3/3	418d
4	sentry-rabbitmq	3/3	418d
5	sentry-sentry-postgresql	1/1	342d
6	sentry-sentry-redis-master	1/1	418d
7	sentry-sentry-redis-replicas	3/3	418d
8	sentry-zookeeper	1/1	418d
9	sentry-zookeeper-clickhouse	3/3	418d

List the pods in your StatefulSet:

```
kubectl get pods -l app.kubernetes.io/component=kafka -n sentry
```

1	NAME	READY	STATUS	RESTARTS	AGE
2	sentry-kafka-0	1/1	Running	0	8d
3	sentry-kafka-1	1/1	Running	0	8d
4	sentry-kafka-2	1/1	Running	0	8d

Once you have the pod names, you can shell into a specific pod using:

```
1 kubectl exec -it sentry-kafka-0 -n sentry -- /bin/bash
```

Remember that StatefulSet pods have predictable names, typically in the format

<statefulset-name>-<ordinal>. So if your StatefulSet is named "my-statefulset", the pods would be named "my-statefulset-0", "my-statefulset-1", and so on.

3. Now we are within the kafka cluster, we can run kafka control commands

a. Run: `kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list` to list the consumer groups

```

1 I have no name!@sentry-kafka-0:/$ kafka-consumer-groups.sh --bootstrap-server localhost:9092 --
  list
2

```

```

3 sentry-commit-log-de3db334d61211ee847682060857d1f1
4 transactions_group
5 ingest-occurrences
6 ingest-monitors
7 snuba-replacers
8 query-subscription-consumer
9 snuba-consumers
10 ingest-replay-recordings
11 ingest-consumer
12 replays_group
13 snuba-transactions-subscriptions-consumers
14 sentry-commit-log-de622d04d61211eeb04126f6bcde8156
15 snuba-events-subscriptions-consumers
16 ingest-metrics-consumer
17 billing-metrics-consumer
18 post-process-forwarder
19 snuba-sessions-subscriptions-consumers
20
21 I have no name!@sentry-kafka-0:/$

```

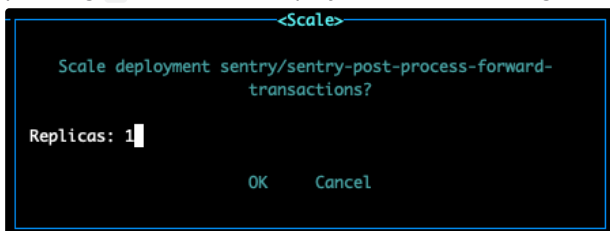
- b. From the output we found earlier in k9s, we can infer that the issue is related to the `snuba-consumers` and `post-process-forwarder` consumer groups, let's dig in and run `kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group post-process-forwarder` to describe the the group

```

1 I have no name!@sentry-kafka-0:/$ kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe -
  -group post-process-forwarder
2
3 GROUP              TOPIC              PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
4 CONSUMER-ID                                     HOST            CLIENT-ID
5 post-process-forwarder transactions  0              37423158        37423220        62
6 rdkafka-13ca77d5-754c-4a2c-b213-4ec31e83358f /10.4.17.44    rdkafka
7 post-process-forwarder generic-events 0              -               81288           -
8 rdkafka-93a0b2c1-94ec-407a-8fb8-70b2f2e9860b /10.4.17.58    rdkafka
9 post-process-forwarder events          0              6805170         7104482         299312
10 -                                     -               -               -
11
12 I have no name!@sentry-kafka-0:/$

```

- c. Looks like the `events` topic has an unresolvable lag, here we see 299312 messages held. To resolve this we need to stop (*by scaling down the deployments to 0*), the corresponding consumer pods, also note that there are no consumers connected to the erroring group listed under `CONSUMER-ID`. These will be removed by the scale down but if there is a consumer, this may cause further errors that will need to be resolved.
- d. To scale down we can return to `k9s` and we can highlight the erroring pods and scale these down by simply pressing `s` to scale the deployment. Note the original replica set and then replace with `0`, then select `OK`



You will see the deployment drop to `0/0` and this has now scaled down.

```

Tilix: ildemundo@steroid:~/src/Raft/backend
Context: gke_staging
Cluster: gke_staging-266717_europe-west2-a_staging
User: gke_staging-266717_europe-west2-a_staging
K9s Rev: v0.31.9
K8s Rev: v1.26.13-gke.1052000
CPU: 12%
MEM: 31%

<0> all      <ctrl-d> Delete  <shift-f> Port-Forward
<1> sentry  <d> Describe    <r> Restart
<2> default <e> Edit       <s> Scale
              <?> Help      <f> Show PortForward
              <l> Logs      <y> YAML
              <p> Logs Previous

Deployments(sentry) [29]
NAME↑ READY UP-TO-DATE AVAILABLE AGE
sentry-billing-metrics-consumer 1/1 1 1 35d
sentry-cron 1/1 1 1 175d
sentry-ingest-consumer-attachments 1/1 1 1 35d
sentry-ingest-consumer-events 1/1 1 1 35d
sentry-ingest-consumer-transactions 1/1 1 1 35d
sentry-ingest-metrics-consumer-perf 1/1 1 1 35d
sentry-ingest-metrics-consumer-rh 1/1 1 1 35d
sentry-ingest-monitors 1/1 1 1 35d
sentry-ingest-occurrences 1/1 1 1 35d
sentry-ingest-replay-recordings 1/1 1 1 35d
sentry-issue-occurrence-consumer 1/1 1 1 35d
sentry-post-process-forward-errors 0/0 0 0 35d
sentry-post-process-forward-issue-platform 0/0 0 0 35d
sentry-post-process-forward-transactions 0/0 0 0 35d
sentry-relay 1/1 1 1 35d
sentry-sessions-consumer 1/1 1 1 35d
sentry-snuba-api 1/1 1 1 175d
sentry-snuba-consumer 1/1 1 1 35d
sentry-snuba-outcomes-consumer 1/1 1 1 35d
sentry-snuba-replacer 1/1 1 1 35d
sentry-snuba-replays-consumer 1/1 1 1 35d
sentry-snuba-subscription-consumer-events 1/1 1 1 35d
sentry-snuba-subscription-consumer-sessions 1/1 1 1 35d
sentry-snuba-subscription-consumer-transactions 1/1 1 1 35d
sentry-snuba-transactions-consumer 1/1 1 1 35d
sentry-subscription-consumer-events 1/1 1 1 35d
sentry-subscription-consumer-transactions 1/1 1 1 35d
sentry-web 1/1 1 1 175d
sentry-worker 3/3 3 3 175d

```

e. We can now reset the lag. By running `kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group post-process-forwarder` you will see we now have lost a group topic without lag.

```

1 I have no name!@sentry-kafka-0:/$ kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe -
  -group post-process-forwarder
2
3 Consumer group 'post-process-forwarder' has no active members.
4
5 GROUP          TOPIC          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
6 CONSUMER-ID     HOST        CLIENT-ID
7 post-process-forwarder events          0           6805170        7105749        300579      -
8 -               -              -
9 post-process-forwarder transactions 0           37432599       37432955        356         -
10 -               -              -
11 I have no name!@sentry-kafka-0:/$

```

f. Now to reset the offsets in the offending topic: `kafka-consumer-groups.sh --bootstrap-server localhost:9092 --group post-process-forwarder --topic events --reset-offsets --to-earliest --execute`

```

1 I have no name!@sentry-kafka-0:/$ kafka-consumer-groups.sh --bootstrap-server localhost:9092 --group
  post-process-forwarder --topic events --reset-offsets --to-earliest --execute
2
3 GROUP          TOPIC          PARTITION  NEW-OFFSET
4 post-process-forwarder events          0           7063080
5 I have no name!@sentry-kafka-0:/$

```

g. Lag should be more acceptable now and consumers can execute: `kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group post-process-forwarder`

```

1 I have no name!@sentry-kafka-0:/$ kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe -
  -group post-process-forwarder
2

```

```

3 Consumer group 'post-process-forwarder' has no active members.
4
5 GROUP          TOPIC          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
6 post-process-forwarder events          0           7063080        7106107        43027        -
7 post-process-forwarder transactions      0           37432599       37435133       2534         -
8 I have no name!@sentry-kafka-0:/$

```

h. We can scale the deployments back to normal (1 pod per deploy)

```

Tilix: ildemundo@steroid:~/src/Raft/backend
Context: gke_staging
Cluster: gke_staging-266717_europe-west2-a_staging
User: gke_staging-266717_europe-west2-a_staging
K9s Rev: v0.31.9
K9s Rev: v1.26.13-gke.1052000
CPU: 22%
MEM: 38%

<0> all      <ctrl-d> Delete      <shift-f> Port-Forward
<1> sentry  <d> Describe        <r> Restart
<2> default <e> Edit           <s> Scale
               <f> Help          <f> Show PortForward
               <l> Logs          <y> YAML
               <p> Logs Previous

Deployments(sentry) [29]
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
sentry-billing-metrics-consumer     1/1      1              1            1 35d
sentry-cron                          1/1      1              1            1 175d
sentry-ingest-consumer-attachments  1/1      1              1            1 35d
sentry-ingest-consumer-events       1/1      1              1            1 35d
sentry-ingest-consumer-transactions 1/1      1              1            1 35d
sentry-ingest-metrics-consumer-perf 1/1      1              1            1 35d
sentry-ingest-metrics-consumer-rh    1/1      1              1            1 35d
sentry-ingest-monitors               1/1      1              1            1 35d
sentry-ingest-occurrences            1/1      1              1            1 35d
sentry-ingest-replay-recordings       1/1      1              1            1 35d
sentry-issue-occurrence-consumer      1/1      1              1            1 35d
sentry-post-process-forward-errors     1/1      1              1            1 35d
sentry-post-process-forward-issue-platform 1/1      1              1            1 35d
sentry-post-process-forward-transactions 1/1      1              1            1 35d
sentry-relay                         1/1      1              1            1 35d
sentry-sessions-consumer              1/1      1              1            1 35d
sentry-snuba-api                     1/1      1              1            1 175d
sentry-snuba-consumer                 1/1      1              1            1 35d
sentry-snuba-outcomes-consumer        1/1      1              1            1 35d
sentry-snuba-replacer                 1/1      1              1            1 35d
sentry-snuba-replays-consumer         1/1      1              1            1 35d
sentry-snuba-subscription-consumer-events 1/1      1              1            1 35d
sentry-snuba-subscription-consumer-sessions 1/1      1              1            1 35d
sentry-snuba-subscription-consumer-transactions 1/1      1              1            1 35d
sentry-snuba-transactions-consumer    1/1      1              1            1 35d
sentry-subscription-consumer-events   1/1      1              1            1 35d
sentry-subscription-consumer-transactions 1/1      1              1            1 35d
sentry-web                           1/1      1              1            1 175d
sentry-worker                         3/3      3              3            3 175d

```

Repeat above steps for the “snuba-consumers” - “events” group-topic and sentry should be back on.

#### Kubectl option:

```
kubectl scale deployment post-process-forwarder-transactions --replicas=0 -n sentry
```

## TLDR

## Preparation

1. Install K9s for easier Kubernetes cluster management.
2. Ensure you're in the correct namespace (sentry).

## Identifying the Issue

1. Use K9s or kubectl to check deployments with errors:

```
1 kubectl get deployments -n sentry
```

2. Locate the Kafka statefulset:

```
1 kubectl get statefulsets -n sentry
```

## Accessing Kafka

1. Shell into a Kafka pod:

```
1 kubectl exec -it sentry-kafka-0 -n sentry -- /bin/bash
```

## Diagnosing the Problem

1. List consumer groups:

```
1 kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list
```

2. Describe the problematic consumer group:

```
1 kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group post-process-forwarder
```

## Resolving the Issue

1. Scale down the affected deployments:

```
1 kubectl scale deployment sentry-post-process-forward-transactions --replicas=0 -n sentry
```

2. Reset the offsets for the problematic topic:

```
1 kafka-consumer-groups.sh --bootstrap-server localhost:9092 --group post-process-forwarder --topic events --reset-offsets --to-earliest --execute
```

3. Verify the lag has been reduced:

```
1 kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group post-process-forwarder
```

4. Scale the deployments back up:

```
1 kubectl scale deployment sentry-post-process-forward-transaction --replicas=1 -n sentry
```

\*\*Taken from an original Ildemundo Roque and Andrei Ivan Google Doc