# KubeCon | CloudNativeCon

## North America 2023

KubeCon | CloudNativeCon

North America 2023

# The Cluster Killer Bug: Learning API Priority and Fairness the Hard Way

*Eddie Zaneski*

# Hi I'm Eddie!

- @eddiezane
- Denver, CO
- Climb big mountains
- Co-chair & tech lead for Kubernetes SIG-CLI
- Staff DevRel + OSS Engineer (hire me)

# Flashback to Feb 1st 2023

https://events.linuxfoundation.org/cloudnativesecuritycon-north-america/

# Karaoke

https://giphy.com/gifs/YbXtbKoi2ZUOc

# Media Interview Tomorrow

https://thenewstack.io/

# Last Minute Prep

https://giphy.com/gifs/ShalitaGrant-sneaking-out-leaving-a-party-not-saying-bye-xlcXy1FF94YaYgxxaO

# Nothing Works

https://giphy.com/gifs/5Zesu5VPNGJlm

# Delete The Cluster & Go To Bed

https://giphy.com/gifs/this-is-fine-9M5jK4GXmD5o1irGrF

# Wake Up…

https://giphy.com/gifs/pnPfFgZi3lnO

# Feb 2nd 2023

- Push interview back
- Triage triage triage
- Here we go…

# Flashback to Feb 1st 2023

https://youtu.be/uPBUKazKjag?si=Rwj3l_erzPe7QzPs

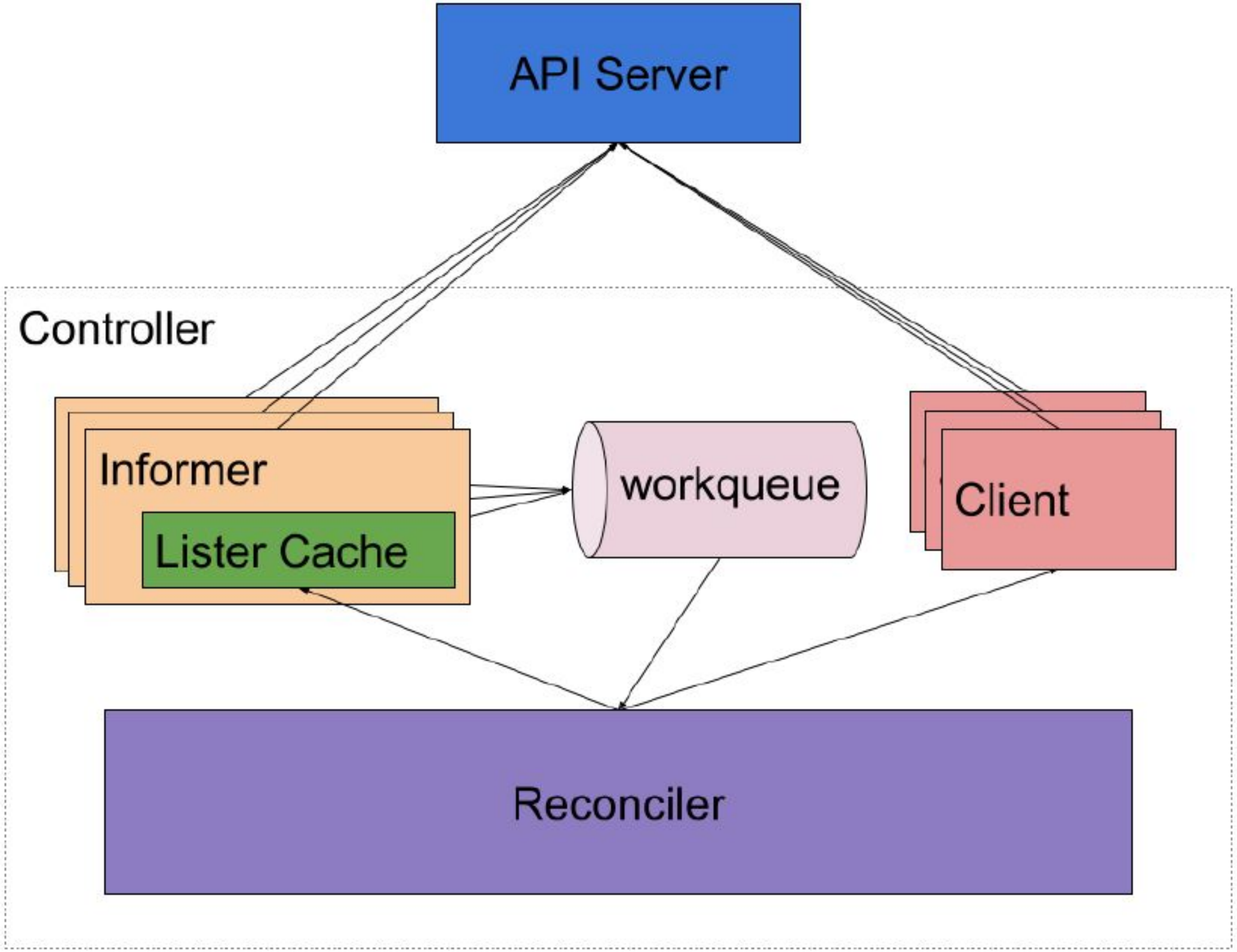
# Early Takeaways

- Don't delete the cluster
- Start ruling out what you can
- Think out loud
- *Glance* at logs + metrics
- Your goal is to reproduce the issue, not understand it yet

# The Controller

- Built around the Sigstore Policy Controller
- Opened long lived watches to the API Server
- Knative
- Ability to run "agentless"

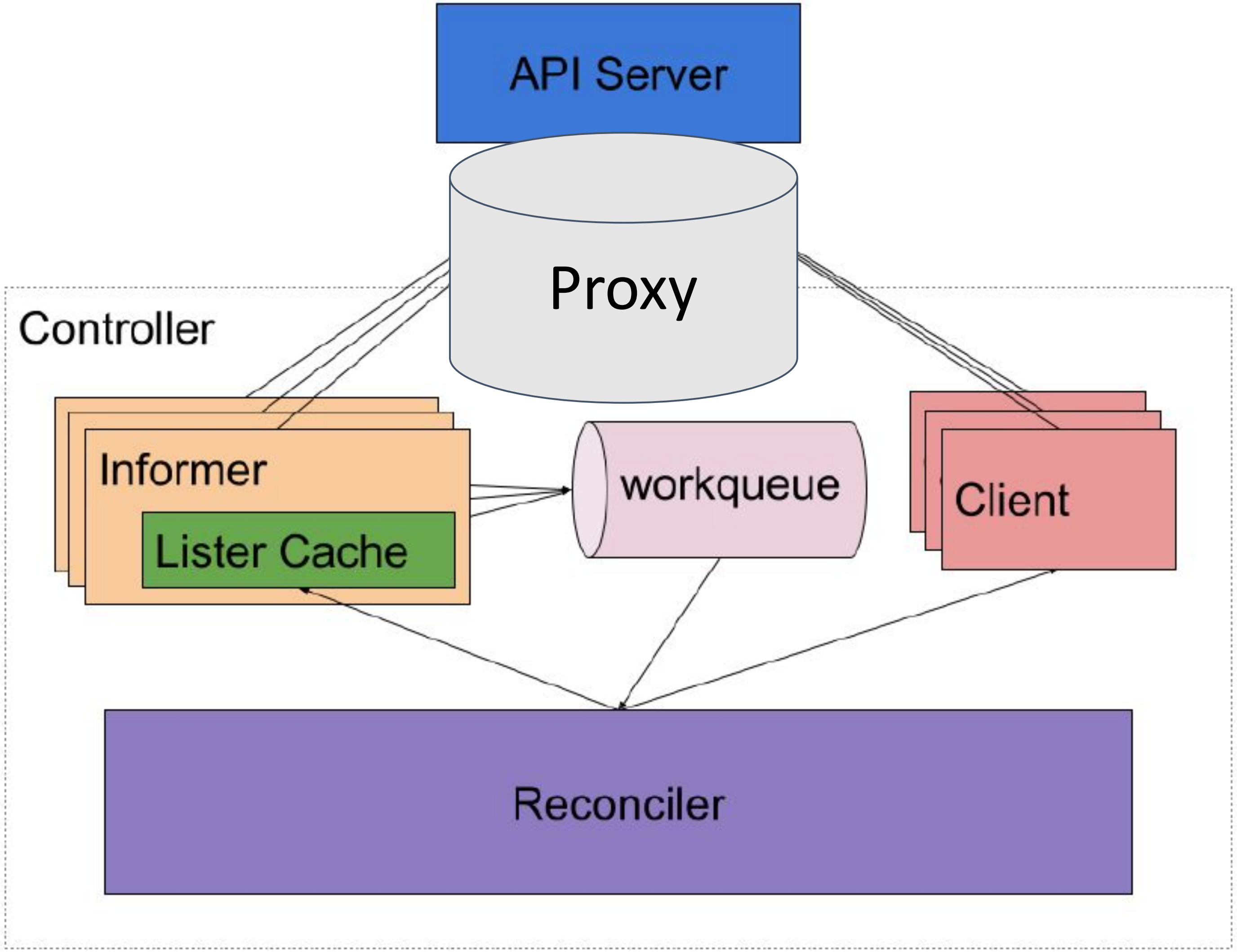
# Controllers



https://edu.chainguard.dev/chainguard/chainguard-enforce/concepts/gulfstream-overview/

# "Agentless"

Some other cluster



https://edu.chainguard.dev/chainguard/chainguard-enforce/concepts/gulfstream-overview/

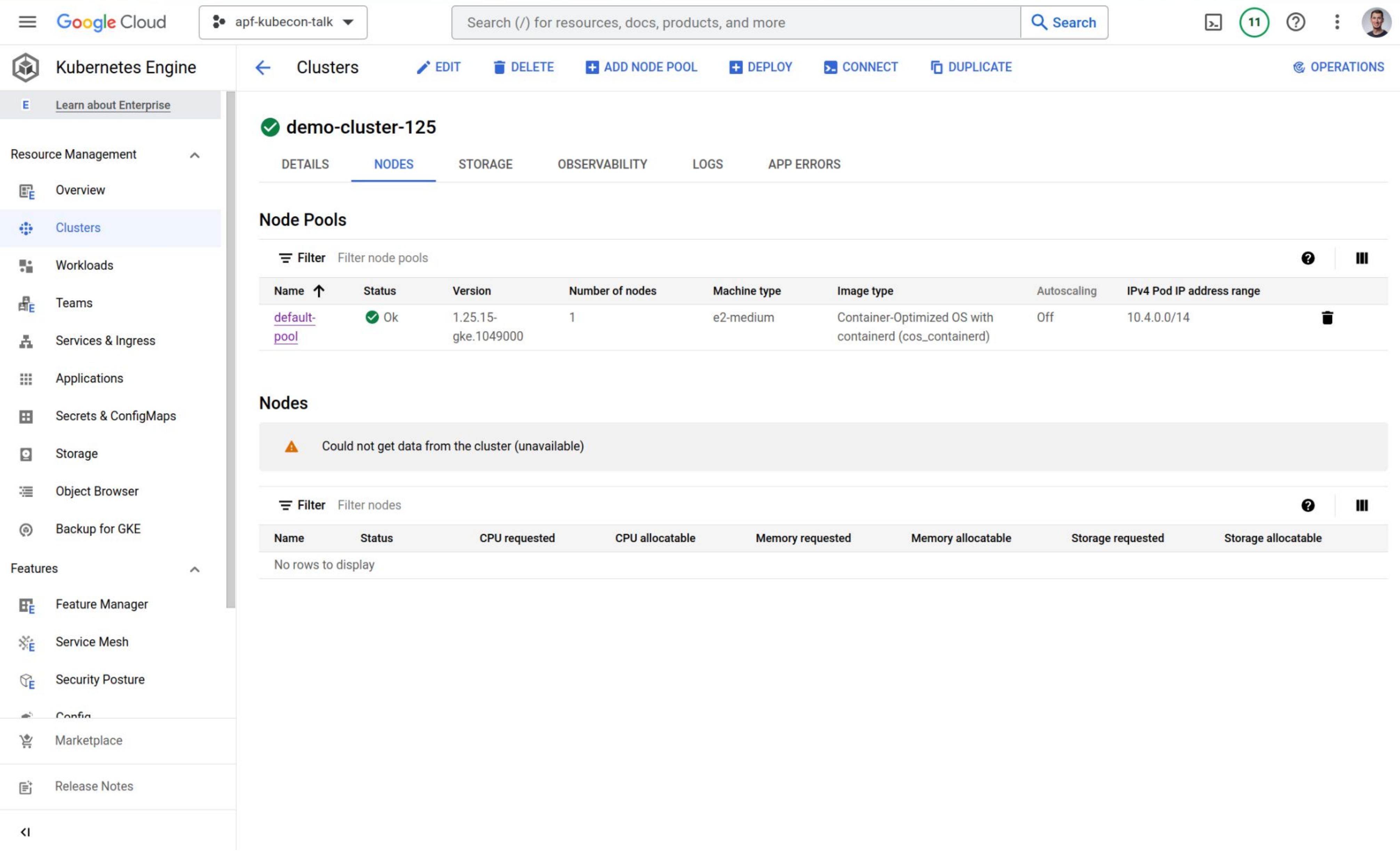
# What We Saw

- The cluster appeared super dead
- Client requests were timing out
- GKE dashboard was dead
- Happened after 4-6 hours

# The Cluster Killer Bug

- Visibly only affected Kubernetes 1.25
- Only GKE
- Only "agentless"
- Only solution seemed to be deleting the cluster

# Uh Oh



✓ demo-cluster-125

## Node Pools

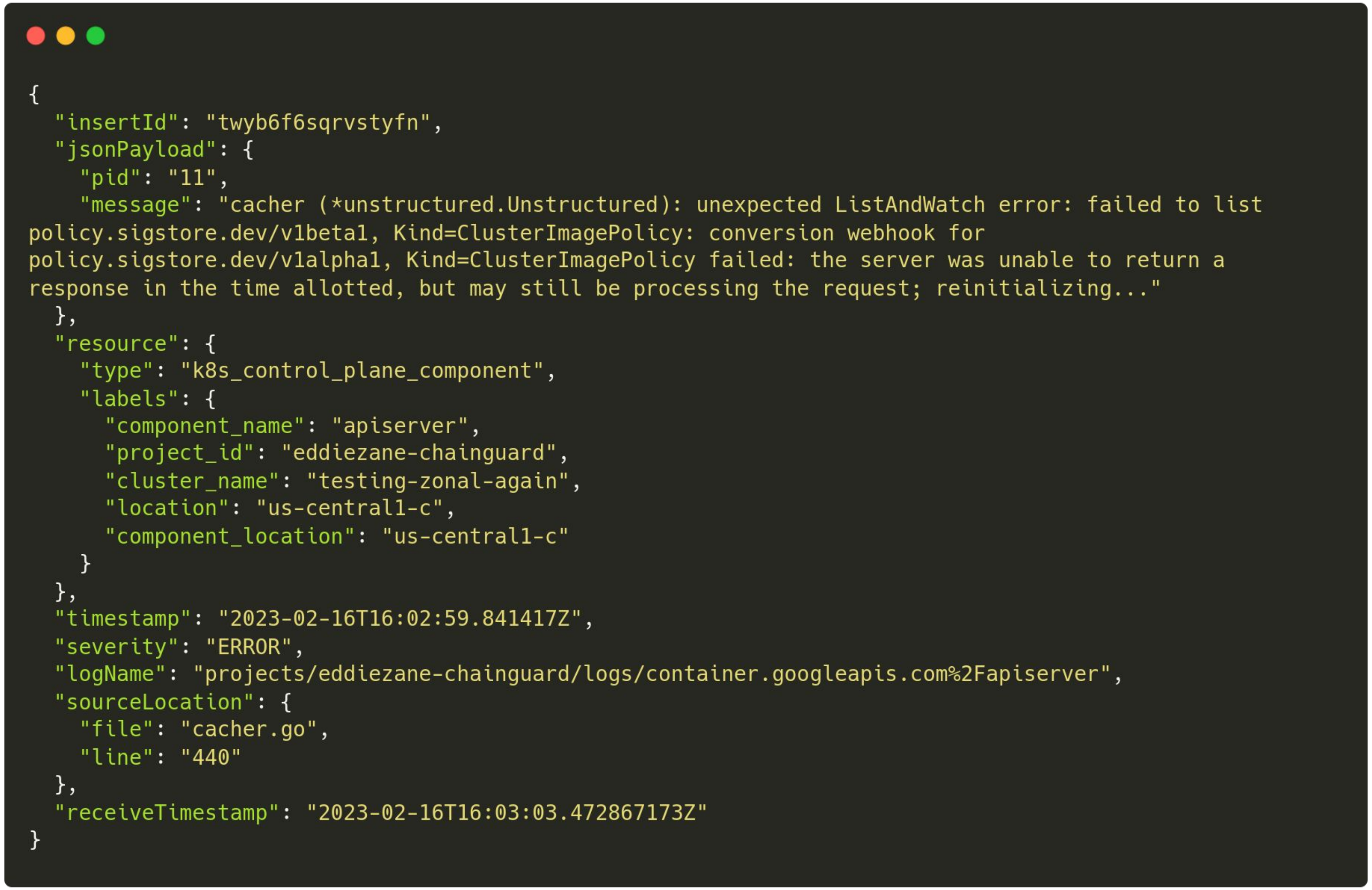
| Name ↑ | Status | Version | Number of nodes | Machine type | Image type | Autoscaling | IPv4 Pod IP address range |
|---|---|---|---|---|---|---|---|
| default-pool | ✓ Ok | 1.25.15-gke.1049000 | 1 | e2-medium | Container-Optimized OS with containerd (cos_containerd) | Off | 10.4.0.0/14 |

## Nodes

⚠ Could not get data from the cluster (unavailable)

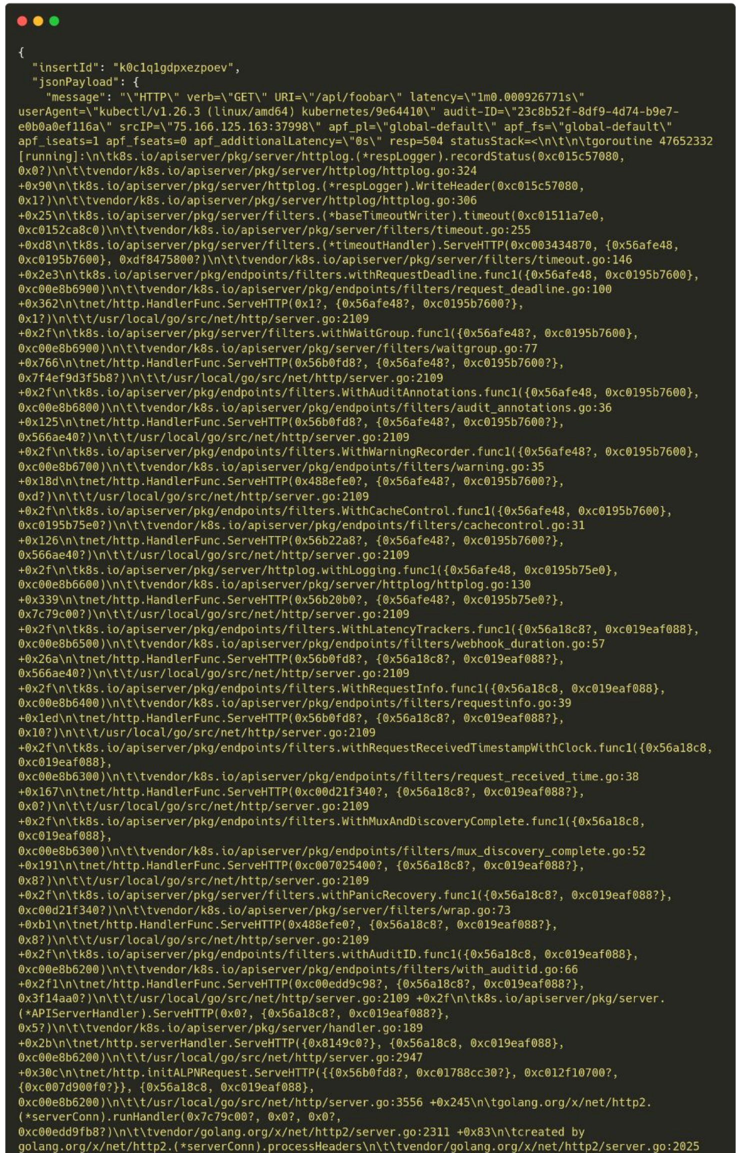
| Name | Status | CPU requested | CPU allocatable | Memory requested | Memory allocatable | Storage requested | Storage allocatable |
|---|---|---|---|---|---|---|---|
| No rows to display | | | | | | | |

# WTF Is This?



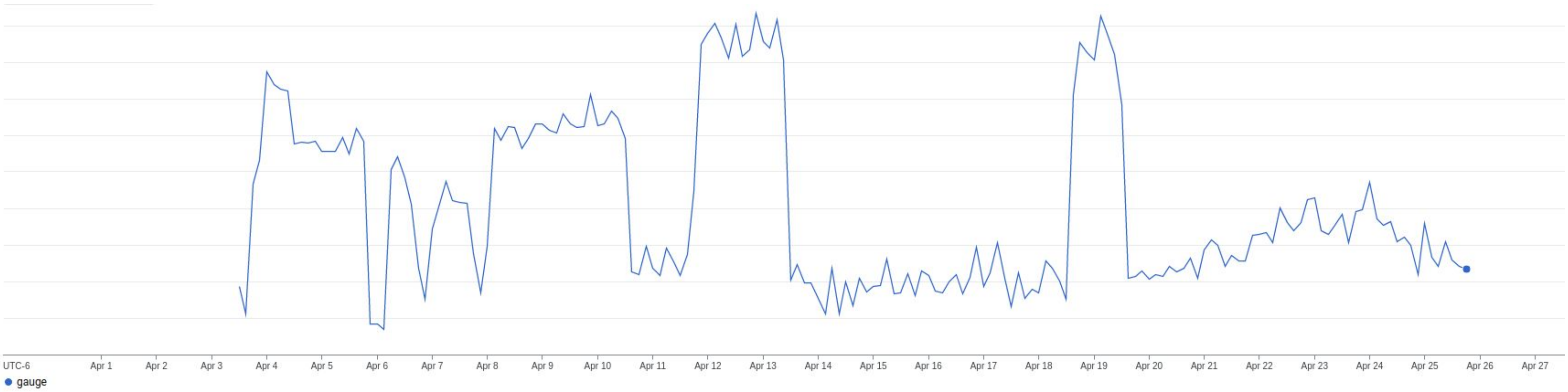
```
{
  "insertId": "twyb6f6sqrvstyfn",
  "jsonPayload": {
    "pid": "11",
    "message": "cacher (*unstructured.Unstructured): unexpected ListAndWatch error: failed to list
policy.sigstore.dev/v1beta1, Kind=ClusterImagePolicy: conversion webhook for
policy.sigstore.dev/v1alpha1, Kind=ClusterImagePolicy failed: the server was unable to return a
response in the time allotted, but may still be processing the request; reinitializing..."
  },
  "resource": {
    "type": "k8s_control_plane_component",
    "labels": {
      "component_name": "apiserver",
      "project_id": "eddiezane-chainguard",
      "cluster_name": "testing-zonal-again",
      "location": "us-central1-c",
      "component_location": "us-central1-c"
    }
  },
  "timestamp": "2023-02-16T16:02:59.841417Z",
  "severity": "ERROR",
  "logName": "projects/eddiezane-chainguard/logs/container.googleapis.com%2Fapiserver",
  "sourceLocation": {
    "file": "cacher.go",
    "line": "440"
  },
  "receiveTimestamp": "2023-02-16T16:03:03.472867173Z"
}
```

# Or This??

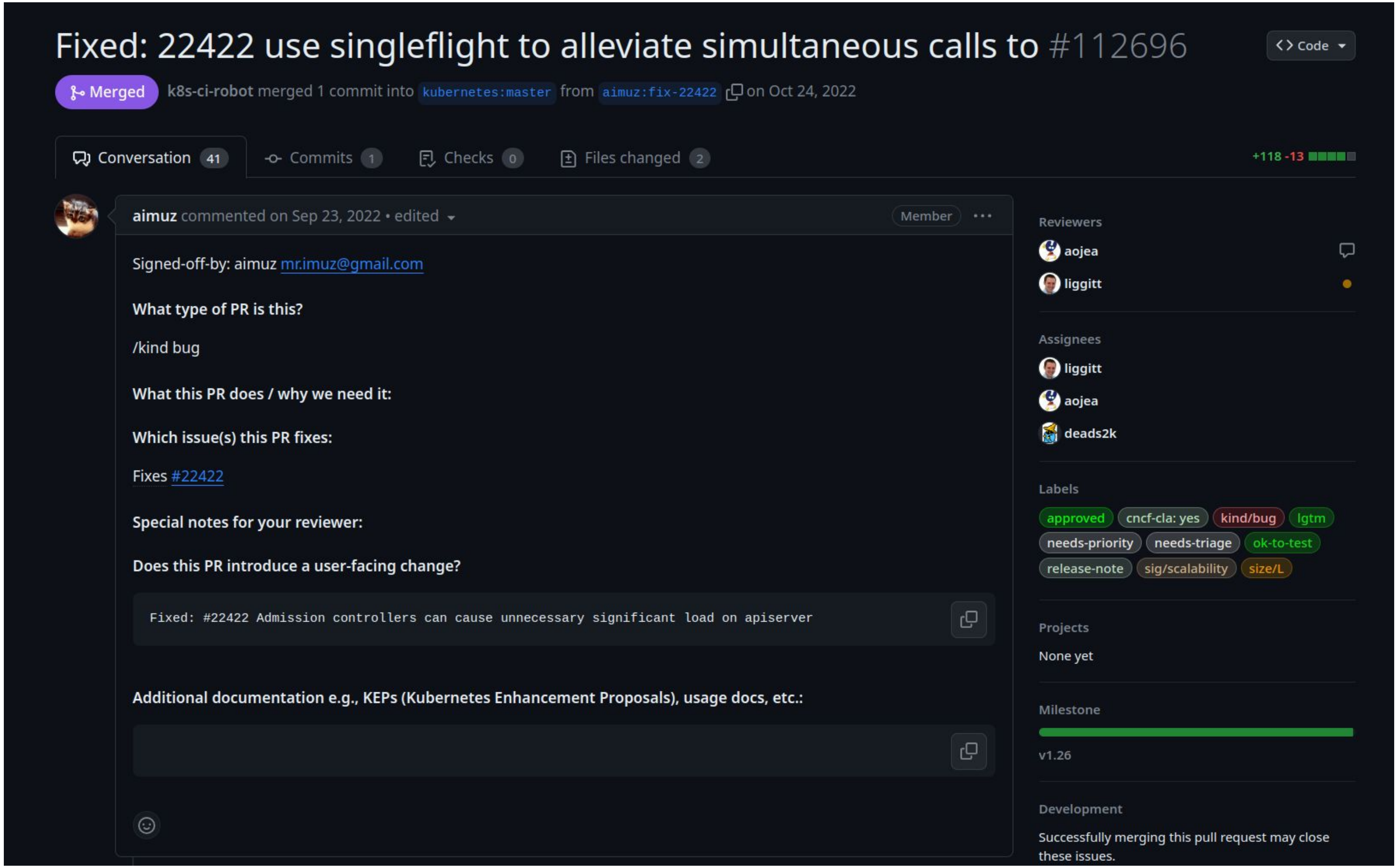


```
{
  "insertId": "k0c1q1gdpxezpoev",
  "jsonPayload": {
    "message": "\"HTTP\" verb=\"GET\" URI=\"/api/foobar\" latency=\"1m0.000926771s\"
userAgent=\"kubectl/v1.26.3 (linux/amd64) kubernetes/9e64410\" audit-ID=\"23c8b52f-8df9-4d74-b9e7-
e0b0a0ef116a\" srcIP=\"75.166.125.163:37998\" apf_pl=\"global-default\" apf_fs=\"global-default\"
apf_iseats=1 apf_fseats=0 apf_additionalLatency=\"0s\" resp=504 statusStack=<\n\t\n\tgoroutine 47652332
[running]:\n\tk8s.io/apiserver/pkg/server/httplog.(*respLogger).recordStatus(0xc015c57080,
0x0?)\n\t\tvendor/k8s.io/apiserver/pkg/server/httplog/httplog.go:324
+0x90\n\tk8s.io/apiserver/pkg/server/httplog.(*respLogger).WriteHeader(0xc015c57080,
0x1?)\n\t\tvendor/k8s.io/apiserver/pkg/server/httplog/httplog.go:306
+0x25\n\tk8s.io/apiserver/pkg/server/filters.(*baseTimeoutWriter).timeout(0xc01511a7e0,
0xc0152ca8c0)\n\t\tvendor/k8s.io/apiserver/pkg/server/filters/timeout.go:255
+0xd8\n\tk8s.io/apiserver/pkg/server/filters.(*timeoutHandler).ServeHTTP(0xc003434870, {0x56afe48,
0xc0195b7600}, 0xdf8475800?)\n\t\tvendor/k8s.io/apiserver/pkg/server/filters/timeout.go:146
+0x2e3\n\tk8s.io/apiserver/pkg/endpoints/filters.withRequestDeadline.func1({0x56afe48, 0xc0195b7600},
0xc00e8b6900)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/request_deadline.go:100
+0x362\n\tnet/http.HandlerFunc.ServeHTTP(0x1?, {0x56afe48?, 0xc0195b7600?},
0x1?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/server/filters.withWaitGroup.func1({0x56afe48?, 0xc0195b7600},
0xc00e8b6900)\n\t\tvendor/k8s.io/apiserver/pkg/server/filters/waitgroup.go:77
+0x766\n\tnet/http.HandlerFunc.ServeHTTP(0x56b0fd8?, {0x56afe48?, 0xc0195b7600?},
0x7f4ef9d3f5b8?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.WithAuditAnnotations.func1({0x56afe48, 0xc0195b7600},
0xc00e8b6800)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/audit_annotations.go:36
+0x125\n\tnet/http.HandlerFunc.ServeHTTP(0x56b0fd8?, {0x56afe48?, 0xc0195b7600?},
0x566ae40?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.WithWarningRecorder.func1({0x56afe48?, 0xc0195b7600},
0xc00e8b6700)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/warning.go:35
+0x18d\n\tnet/http.HandlerFunc.ServeHTTP(0x488efe0?, {0x56afe48?, 0xc0195b7600?},
0xd?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.WithCacheControl.func1({0x56afe48, 0xc0195b7600},
0xc0195b75e0?)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/cachecontrol.go:31
+0x126\n\tnet/http.HandlerFunc.ServeHTTP(0x56b22a8?, {0x56afe48?, 0xc0195b7600?},
0x566ae40?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/server/httplog.withLogging.func1({0x56afe48, 0xc0195b75e0},
0xc00e8b6600)\n\t\tvendor/k8s.io/apiserver/pkg/server/httplog/httplog.go:130
+0x339\n\tnet/http.HandlerFunc.ServeHTTP(0x56b20b0?, {0x56afe48?, 0xc0195b75e0?},
0x7c79c00?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.WithLatencyTrackers.func1({0x56a18c8?, 0xc019eaf088},
0xc00e8b6500)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/webhook_duration.go:57
+0x26a\n\tnet/http.HandlerFunc.ServeHTTP(0x56b0fd8?, {0x56a18c8?, 0xc019eaf088?},
0x566ae40?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.WithRequestInfo.func1({0x56a18c8, 0xc019eaf088},
0xc00e8b6400)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/requestinfo.go:39
+0x1ed\n\tnet/http.HandlerFunc.ServeHTTP(0x56b0fd8?, {0x56a18c8?, 0xc019eaf088?},
0x10?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.withRequestReceivedTimestampWithClock.func1({0x56a18c8,
0xc019eaf088},
0xc00e8b6300)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/request_received_time.go:38
+0x167\n\tnet/http.HandlerFunc.ServeHTTP(0xc00d21f340?, {0x56a18c8?, 0xc019eaf088?},
0x0?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.WithMuxAndDiscoveryComplete.func1({0x56a18c8,
0xc019eaf088},
0xc00e8b6300)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/mux_discovery_complete.go:52
+0x191\n\tnet/http.HandlerFunc.ServeHTTP(0xc007025400?, {0x56a18c8?, 0xc019eaf088?},
0x8?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/server/filters.withPanicRecovery.func1({0x56a18c8?, 0xc019eaf088?},
0xc00d21f340?)\n\t\tvendor/k8s.io/apiserver/pkg/server/filters/wrap.go:73
+0xb1\n\tnet/http.HandlerFunc.ServeHTTP(0x488efe0?, {0x56a18c8?, 0xc019eaf088?},
0x8?)\n\t\t/usr/local/go/src/net/http/server.go:2109
+0x2f\n\tk8s.io/apiserver/pkg/endpoints/filters.withAuditID.func1({0x56a18c8, 0xc019eaf088},
0xc00e8b6200)\n\t\tvendor/k8s.io/apiserver/pkg/endpoints/filters/with_auditid.go:66
+0x2f1\n\tnet/http.HandlerFunc.ServeHTTP(0xc00edd9c98?, {0x56a18c8?, 0xc019eaf088?},
0x3f14aa0?)\n\t\t/usr/local/go/src/net/http/server.go:2109 +0x2f\n\tk8s.io/apiserver/pkg/server.
(*APIServerHandler).ServeHTTP(0x0?, {0x56a18c8?, 0xc019eaf088?},
0x5?)\n\t\tvendor/k8s.io/apiserver/pkg/server/handler.go:189
+0x2b\n\tnet/http.serverHandler.ServeHTTP({0x8149c0?}, {0x56a18c8, 0xc019eaf088},
0xc00e8b6200)\n\t\t/usr/local/go/src/net/http/server.go:2947
+0x30c\n\tnet/http.initALPNRequest.ServeHTTP({{0x56b0fd8?, 0xc01788cc30?}, 0xc012f10700?,
{0xc007d900f0?}}, {0x56a18c8, 0xc019eaf088},
0xc00e8b6200)\n\t\t/usr/local/go/src/net/http/server.go:3556 +0x245\n\tgolang.org/x/net/http2.
(*serverConn).runHandler(0x7c79c00?, 0x0?, 0x0?,
0xc00edd9fb8?)\n\t\tvendor/golang.org/x/net/http2/server.go:2311 +0x83\n\tcreated by
golang.org/x/net/http2.(*serverConn).processHeaders\n\t\tvendor/golang.org/x/net/http2/server.go:2025
```

# API Server Memory



# First Theory



## Fixed: 22422 use singleflight to alleviate simultaneous calls to #112696

<> Code

Merged k8s-ci-robot merged 1 commit into `kubernetes:master` from `aimuz:fix-22422` on Oct 24, 2022

Conversation 41 | Commits 1 | Checks 0 | Files changed 2 | +118 -13

**aimuz** commented on Sep 23, 2022 • edited | Member

Signed-off-by: aimuz mr.imuz@gmail.com

**What type of PR is this?**

/kind bug

**What this PR does / why we need it:**

**Which issue(s) this PR fixes:**

Fixes #22422

**Special notes for your reviewer:**

**Does this PR introduce a user-facing change?**

```
Fixed: #22422 Admission controllers can cause unnecessary significant load on apiserver
```

**Additional documentation e.g., KEPs (Kubernetes Enhancement Proposals), usage docs, etc.:**

```
```

Reviewers

aojea

liggitt

Assignees

liggitt

aojea

deads2k

Labels

approved | cncf-cla: yes | kind/bug | lgtm | needs-priority | needs-triage | ok-to-test | release-note | sig/scalability | size/L

Projects

None yet

Milestone

v1.26

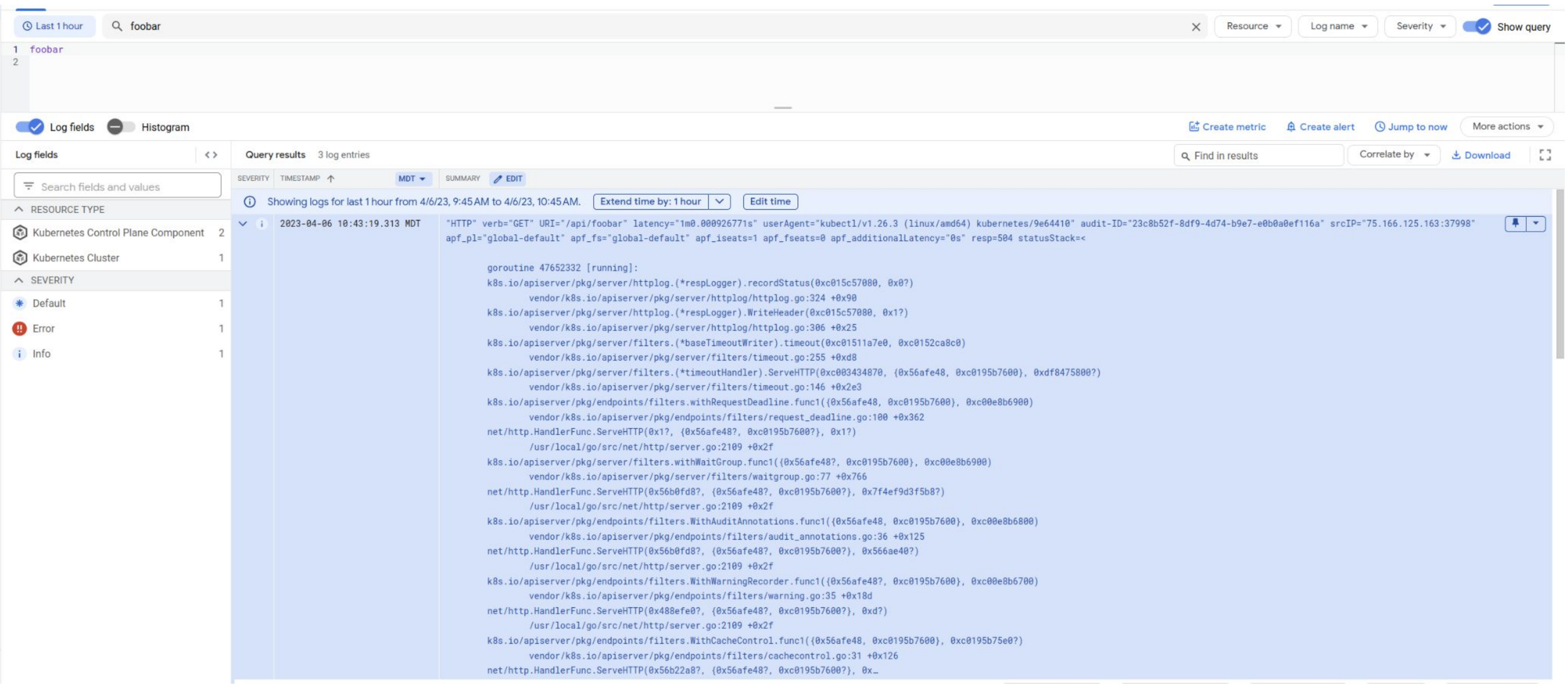
Development

Successfully merging this pull request may close these issues.

# More Lessons

- Have a baseline of what your logs + metrics look like
- Slow down
- Get creative and experiment

# 0xDEADBEEF



```
foobar
```

Search fields and values

RESOURCE TYPE

| | |
|---|---|
| Kubernetes Control Plane Component | 2 |
| Kubernetes Cluster | 1 |

SEVERITY

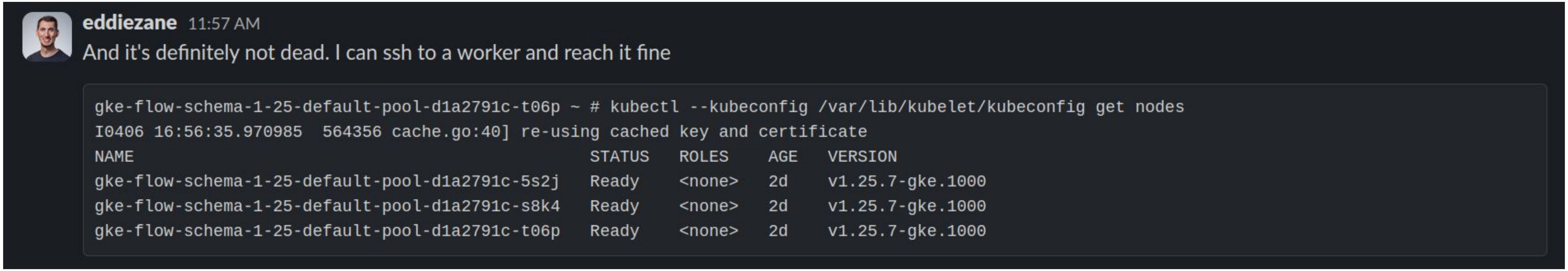
| | |
|---|---|
| Default | 1 |
| Error | 1 |
| Info | 1 |

Showing logs for last 1 hour from 4/6/23, 9:45 AM to 4/6/23, 10:45 AM.

```
2023-04-06 10:43:19.313 MDT   "HTTP" verb="GET" URI="/api/foobar" latency="1m0.000926771s" userAgent="kubectl/v1.26.3 (linux/amd64) kubernetes/9e64410" audit-ID="23c8b52f-8df9-4d74-b9e7-e0b0a0ef116a" srcIP="75.166.125.163:37998"
apf_pl="global-default" apf_fs="global-default" apf_iseats=1 apf_fseats=0 apf_additionalLatency="0s" resp=504 statusStack=<

        goroutine 47652332 [running]:
        k8s.io/apiserver/pkg/server/httplog.(*respLogger).recordStatus(0xc015c57080, 0x0?)
                vendor/k8s.io/apiserver/pkg/server/httplog/httplog.go:324 +0x90
        k8s.io/apiserver/pkg/server/httplog.(*respLogger).WriteHeader(0xc015c57080, 0x1?)
                vendor/k8s.io/apiserver/pkg/server/httplog/httplog.go:306 +0x25
        k8s.io/apiserver/pkg/server/filters.(*baseTimeoutWriter).timeout(0xc01511a7e0, 0xc0152ca8c0)
                vendor/k8s.io/apiserver/pkg/server/filters/timeout.go:255 +0xd8
        k8s.io/apiserver/pkg/server/filters.(*timeoutHandler).ServeHTTP(0xc003434870, {0x56afe48, 0xc0195b7600}, 0xdf8475800?)
                vendor/k8s.io/apiserver/pkg/server/filters/timeout.go:146 +0x2e3
        k8s.io/apiserver/pkg/endpoints/filters.withRequestDeadline.func1({0x56afe48, 0xc0195b7600}, 0xc00e8b6900)
                vendor/k8s.io/apiserver/pkg/endpoints/filters/request_deadline.go:100 +0x362
        net/http.HandlerFunc.ServeHTTP(0x1?, {0x56afe48?, 0xc0195b7600?}, 0x1?)
                /usr/local/go/src/net/http/server.go:2109 +0x2f
        k8s.io/apiserver/pkg/server/filters.withWaitGroup.func1({0x56afe48?, 0xc0195b7600}, 0xc00e8b6900)
                vendor/k8s.io/apiserver/pkg/server/filters/waitgroup.go:77 +0x766
        net/http.HandlerFunc.ServeHTTP(0x56b0fd8?, {0x56afe48?, 0xc0195b7600?}, 0x7f4ef9d3f5b8?)
                /usr/local/go/src/net/http/server.go:2109 +0x2f
        k8s.io/apiserver/pkg/endpoints/filters.WithAuditAnnotations.func1({0x56afe48, 0xc0195b7600}, 0xc00e8b6800)
                vendor/k8s.io/apiserver/pkg/endpoints/filters/audit_annotations.go:36 +0x125
        net/http.HandlerFunc.ServeHTTP(0x56b0fd8?, {0x56afe48?, 0xc0195b7600?}, 0x566ae40?)
                /usr/local/go/src/net/http/server.go:2109 +0x2f
        k8s.io/apiserver/pkg/endpoints/filters.WithWarningRecorder.func1({0x56afe48?, 0xc0195b7600}, 0xc00e8b6700)
                vendor/k8s.io/apiserver/pkg/endpoints/filters/warning.go:35 +0x18d
        net/http.HandlerFunc.ServeHTTP(0x488efe0?, {0x56afe48?, 0xc0195b7600?}, 0xd?)
                /usr/local/go/src/net/http/server.go:2109 +0x2f
        k8s.io/apiserver/pkg/endpoints/filters.WithCacheControl.func1({0x56afe48, 0xc0195b7600}, 0xc0195b75e0?)
                vendor/k8s.io/apiserver/pkg/endpoints/filters/cachecontrol.go:31 +0x126
        net/http.HandlerFunc.ServeHTTP(0x56b22a8?, {0x56afe48?, 0xc0195b7600?}, 0x…
```

# But it's not dead?

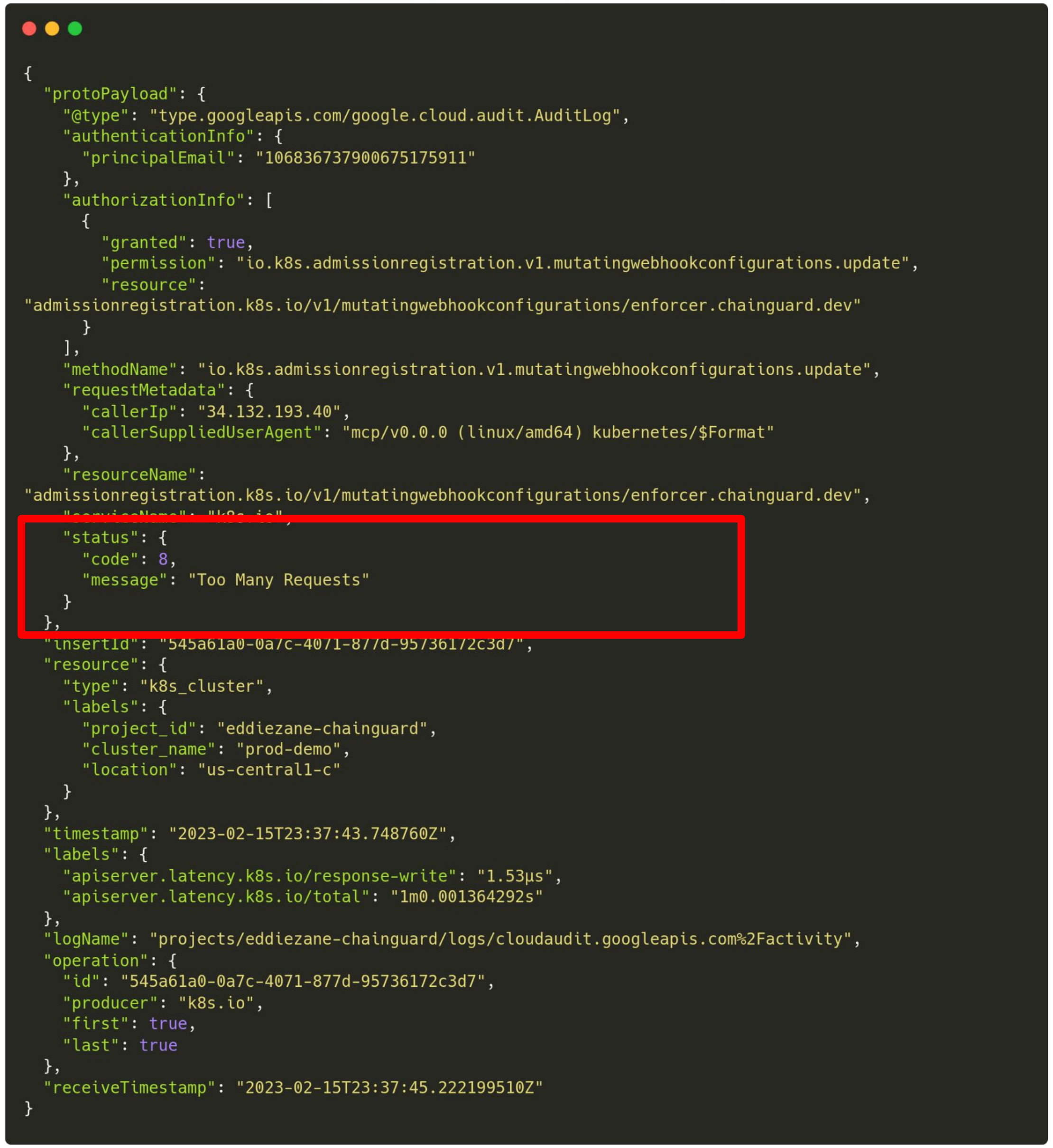


**eddiezane** 11:57 AM

And it's definitely not dead. I can ssh to a worker and reach it fine

```
gke-flow-schema-1-25-default-pool-d1a2791c-t06p ~ # kubectl --kubeconfig /var/lib/kubelet/kubeconfig get nodes
I0406 16:56:35.970985  564356 cache.go:40] re-using cached key and certificate
NAME                                              STATUS   ROLES    AGE   VERSION
gke-flow-schema-1-25-default-pool-d1a2791c-5s2j   Ready    <none>   2d    v1.25.7-gke.1000
gke-flow-schema-1-25-default-pool-d1a2791c-s8k4   Ready    <none>   2d    v1.25.7-gke.1000
gke-flow-schema-1-25-default-pool-d1a2791c-t06p   Ready    <none>   2d    v1.25.7-gke.1000
```

# Bread Crumbs



```
{
  "protoPayload": {
    "@type": "type.googleapis.com/google.cloud.audit.AuditLog",
    "authenticationInfo": {
      "principalEmail": "106836737900675175911"
    },
    "authorizationInfo": [
      {
        "granted": true,
        "permission": "io.k8s.admissionregistration.v1.mutatingwebhookconfigurations.update",
        "resource":
"admissionregistration.k8s.io/v1/mutatingwebhookconfigurations/enforcer.chainguard.dev"
      }
    ],
    "methodName": "io.k8s.admissionregistration.v1.mutatingwebhookconfigurations.update",
    "requestMetadata": {
      "callerIp": "34.132.193.40",
      "callerSuppliedUserAgent": "mcp/v0.0.0 (linux/amd64) kubernetes/$Format"
    },
    "resourceName":
"admissionregistration.k8s.io/v1/mutatingwebhookconfigurations/enforcer.chainguard.dev",
    "serviceName": "k8s.io",
    "status": {
      "code": 8,
      "message": "Too Many Requests"
    }
  },
  "insertId": "545a61a0-0a7c-4071-877d-95736172c3d7",
  "resource": {
    "type": "k8s_cluster",
    "labels": {
      "project_id": "eddiezane-chainguard",
      "cluster_name": "prod-demo",
      "location": "us-central1-c"
    }
  },
  "timestamp": "2023-02-15T23:37:43.748760Z",
  "labels": {
    "apiserver.latency.k8s.io/response-write": "1.53µs",
    "apiserver.latency.k8s.io/total": "1m0.001364292s"
  },
  "logName": "projects/eddiezane-chainguard/logs/cloudaudit.googleapis.com%2Factivity",
  "operation": {
    "id": "545a61a0-0a7c-4071-877d-95736172c3d7",
    "producer": "k8s.io",
    "first": true,
    "last": true
  },
  "receiveTimestamp": "2023-02-15T23:37:45.222199510Z"
}
```

# API Server Tuning

- `--max-requests-inflight`
- `--max-mutating-requests-inflight`

# API Priority and Fairness

The API Priority and Fairness feature (APF) is an alternative that improves upon aforementioned max-inflight limitations. APF classifies and isolates requests in a more fine-grained way. It also introduces a limited amount of queuing, so that no requests are rejected in cases of very brief bursts.

https://kubernetes.io/docs/concepts/cluster-administration/flow-control/

# API Priority and Fairness

Without APF enabled, overall concurrency in the API server is limited by the *kube-apiserver* flags *--max-requests-inflight* and *--max-mutating-requests-inflight*. With APF enabled, the concurrency limits defined by these flags are summed and then the sum is divided up among a configurable set of priority levels. Each incoming request is assigned to a single priority level, and each priority level will only dispatch as many concurrent requests as its particular limit allows.

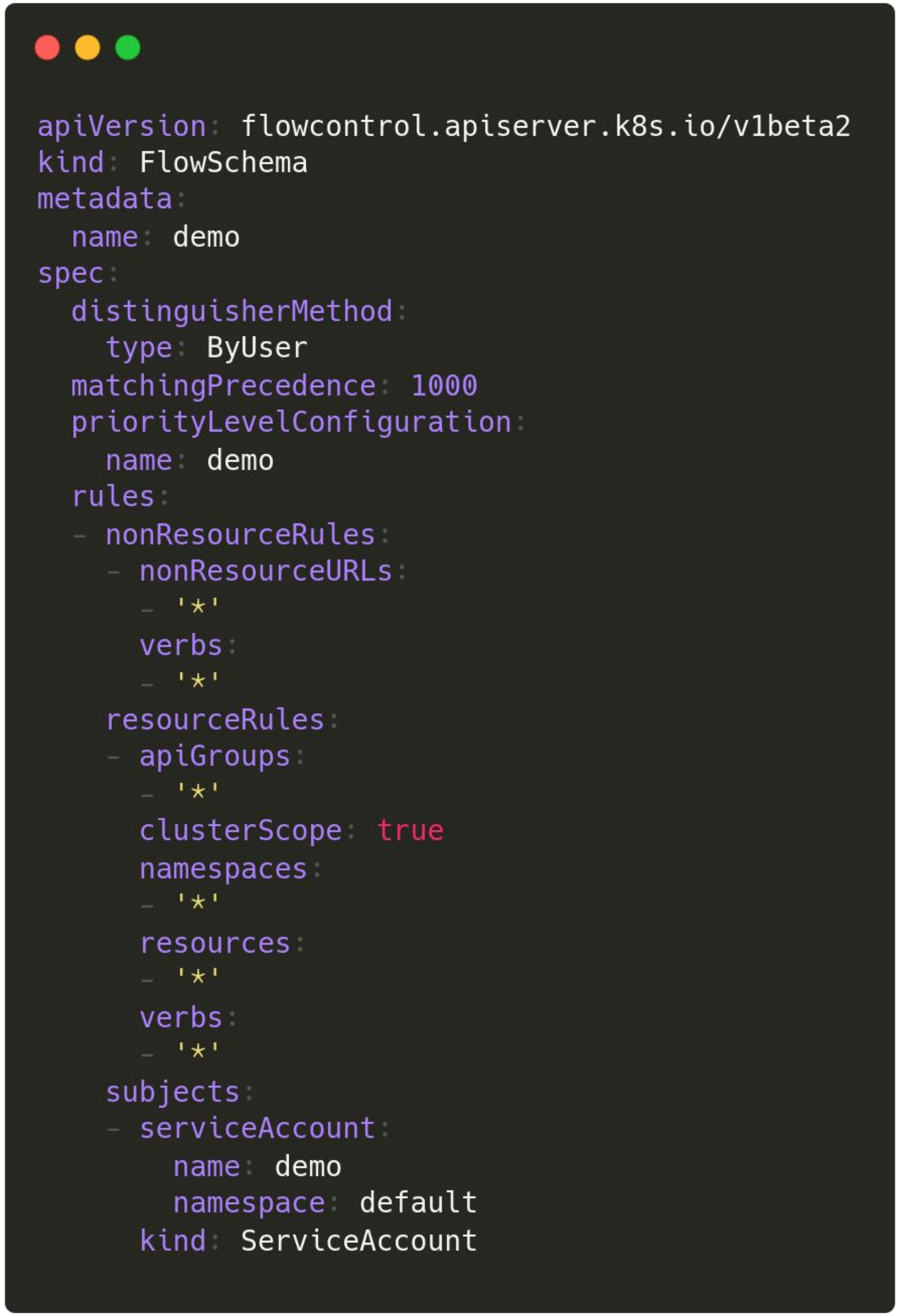
https://kubernetes.io/docs/concepts/cluster-administration/flow-control/

# API Priority and Fairness

- Beta in 1.20
- Enabled by default
- How you control traffic in an overload situation
- "It doesn't matter until it does"

# FlowSchema

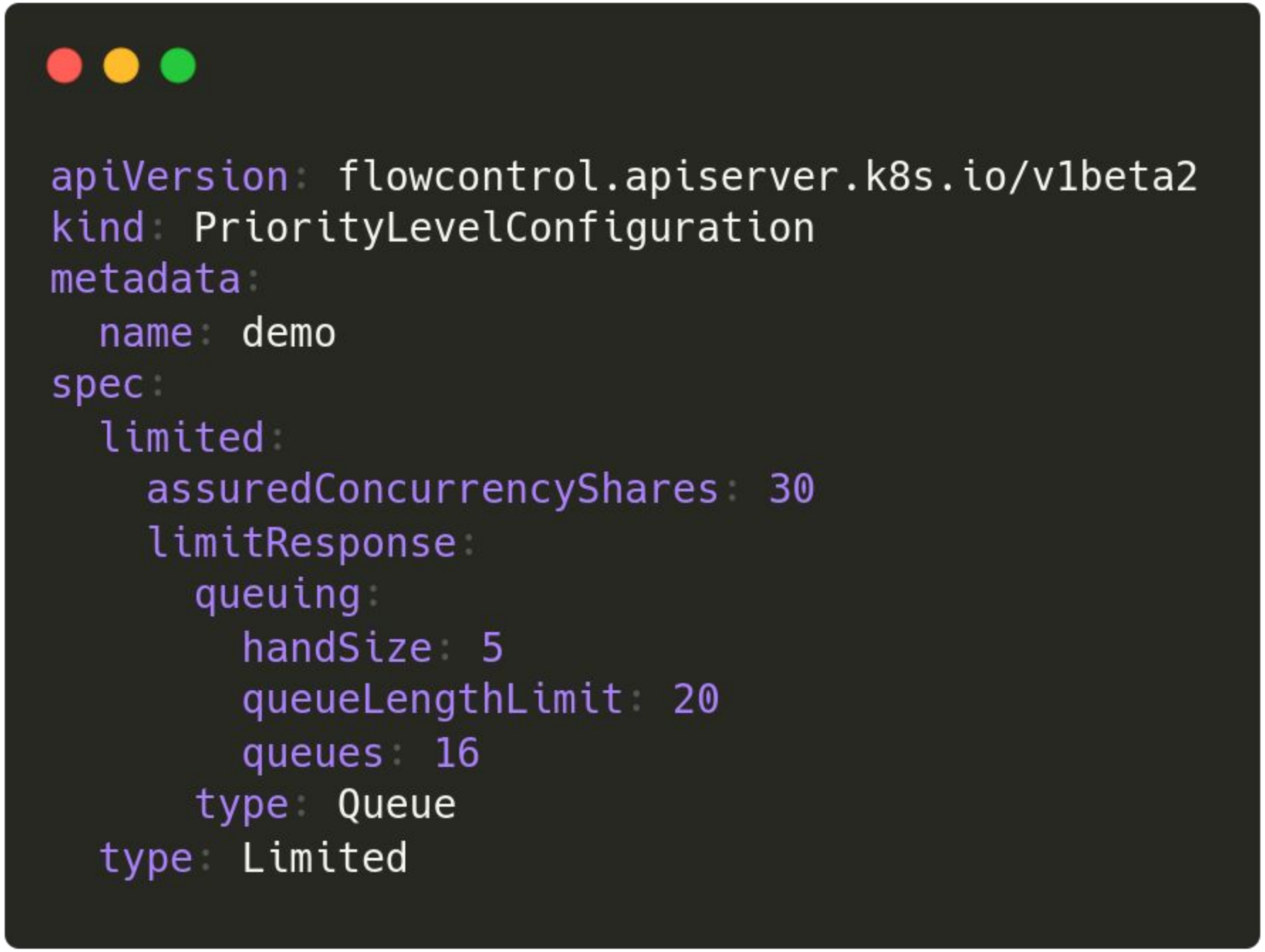
- flowcontrol.apiserver.k8s.io/v1beta3
- Similar to RBAC. The "who" and "what"
    - Subjects - Users, Service Accounts, Groups
    - (non)Resources - GVRs (Pods, Deployments), Paths (/metrics)
- Matching precedence
- Maps to a PriorityLevelConfiguration



```yaml
apiVersion: flowcontrol.apiserver.k8s.io/v1beta2
kind: FlowSchema
metadata:
  name: demo
spec:
  distinguisherMethod:
    type: ByUser
  matchingPrecedence: 1000
  priorityLevelConfiguration:
    name: demo
  rules:
  - nonResourceRules:
    - nonResourceURLs:
      - '*'
      verbs:
      - '*'
    resourceRules:
    - apiGroups:
      - '*'
      clusterScope: true
      namespaces:
      - '*'
      resources:
      - '*'
      verbs:
      - '*'
    subjects:
    - serviceAccount:
        name: demo
        namespace: default
      kind: ServiceAccount
```

# PriorityLevelConfiguration

- "Limited" or "Exempt"
- "Queue" or "Reject"
- Knobs to turn for queueing



```yaml
apiVersion: flowcontrol.apiserver.k8s.io/v1beta2
kind: PriorityLevelConfiguration
metadata:
  name: demo
spec:
  limited:
    assuredConcurrencyShares: 30
    limitResponse:
      queuing:
        handSize: 5
        queueLengthLimit: 20
        queues: 16
      type: Queue
  type: Limited
```

# QueuingConfiguration

- `assuredConcurrencyShares`
- `handSize`
- `queueLengthLimit`
- `queues`

# QueuingConfiguration

- Increasing *queues* reduces the rate of collisions between different flows, at the cost of increased memory usage. A value of 1 here effectively disables the fair-queuing logic, but still allows requests to be queued.
- Increasing *queueLengthLimit* allows larger bursts of traffic to be sustained without dropping any requests, at the cost of increased latency and memory usage
- Changing *handSize* allows you to adjust the probability of collisions between different flows and the overall concurrency available to a single flow in an overload situation.

# Built-in Levels

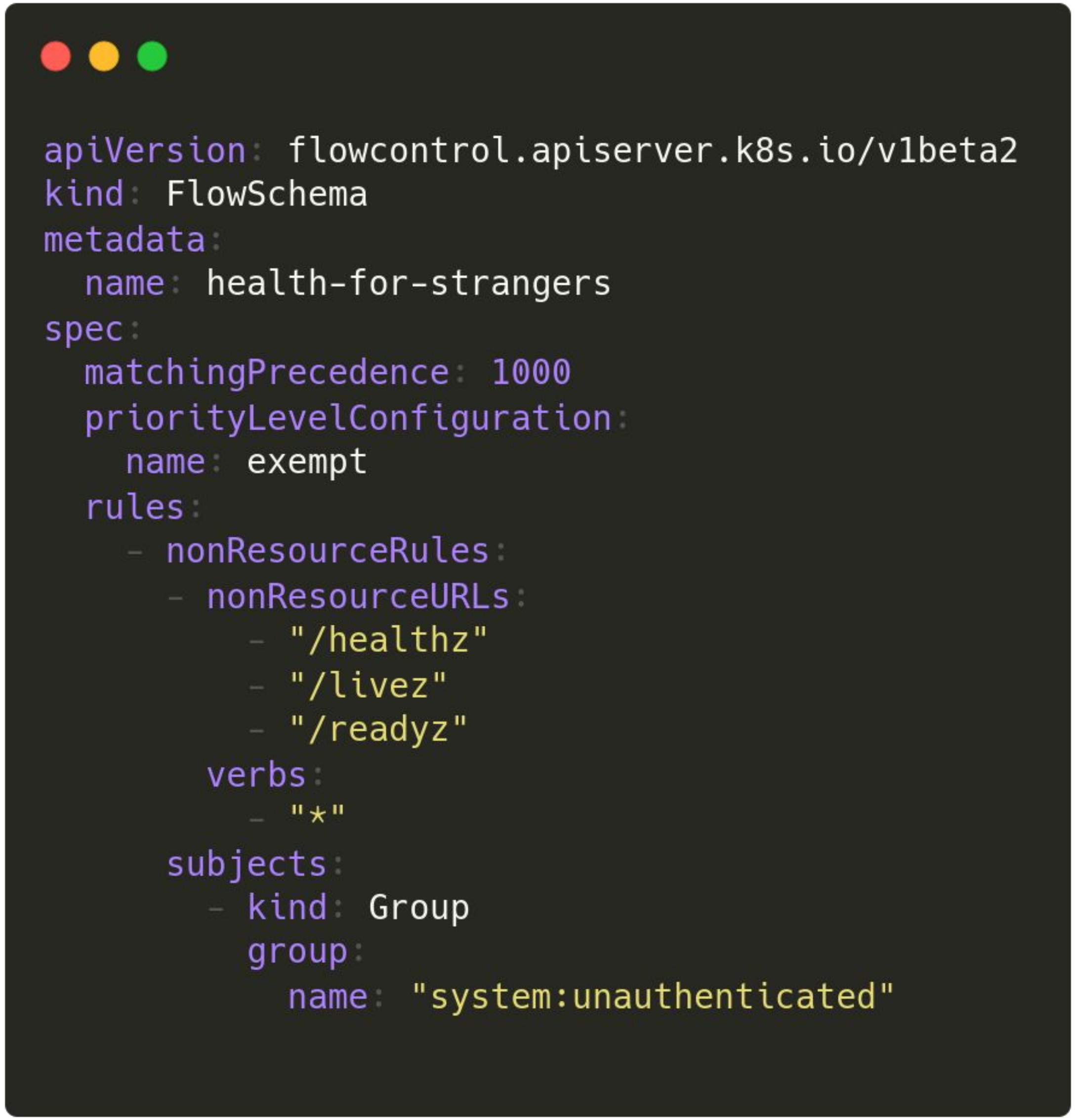
- The workload-low priority level is for requests from any other service account, which will typically include all requests from controllers running in Pods.
- The global-default priority level handles all other traffic, e.g. interactive kubectl commands run by nonprivileged users.

https://kubernetes.io/docs/concepts/cluster-administration/flow-control/#suggested-configuration-objects

# Special Cases

- Lists
- Watches
- system:masters

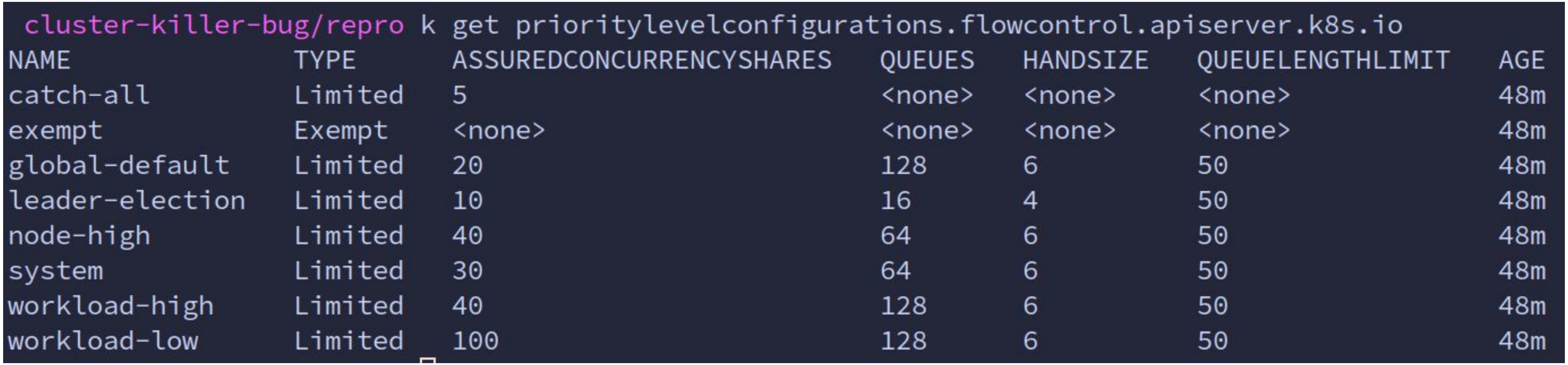
# Health Checks



```
apiVersion: flowcontrol.apiserver.k8s.io/v1beta2
kind: FlowSchema
metadata:
  name: health-for-strangers
spec:
  matchingPrecedence: 1000
  priorityLevelConfiguration:
    name: exempt
  rules:
    - nonResourceRules:
      - nonResourceURLs:
          - "/healthz"
          - "/livez"
          - "/readyz"
        verbs:
          - "*"
      subjects:
        - kind: Group
          group:
            name: "system:unauthenticated"
```

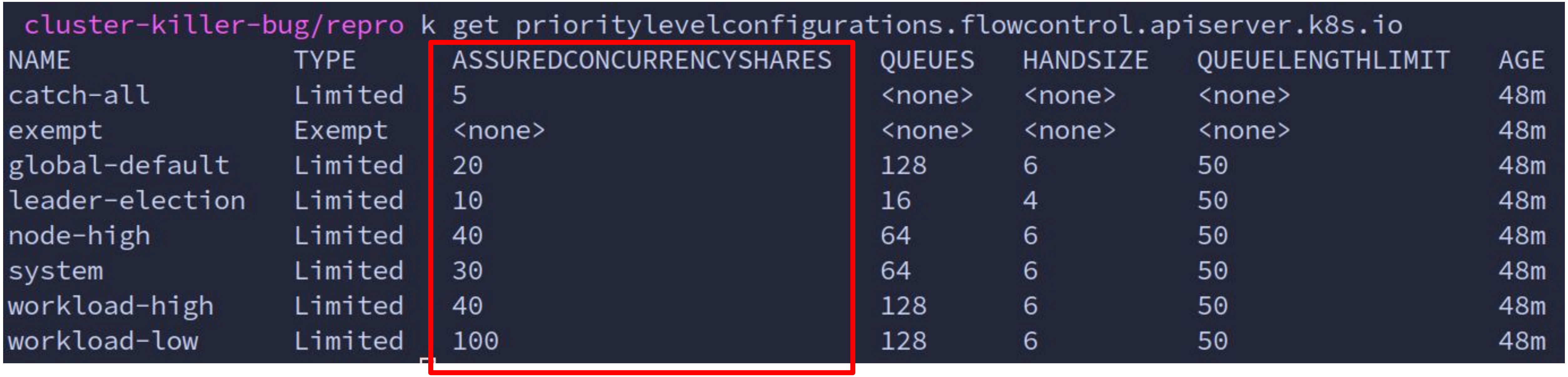
https://kubernetes.io/docs/concepts/cluster-administration/flow-control/

# Levels



```
 cluster-killer-bug/repro k get prioritylevelconfigurations.flowcontrol.apiserver.k8s.io
NAME              TYPE      ASSUREDCONCURRENCYSHARES   QUEUES   HANDSIZE   QUEUELENGTHLIMIT   AGE
catch-all         Limited   5                          <none>   <none>     <none>             48m
exempt            Exempt    <none>                     <none>   <none>     <none>             48m
global-default    Limited   20                         128      6          50                 48m
leader-election   Limited   10                         16       4          50                 48m
node-high         Limited   40                         64       6          50                 48m
system            Limited   30                         64       6          50                 48m
workload-high     Limited   40                         128      6          50                 48m
workload-low      Limited   100                        128      6          50                 48m
```

# ACS



```
 cluster-killer-bug/repro k get prioritylevelconfigurations.flowcontrol.apiserver.k8s.io
NAME              TYPE       ASSUREDCONCURRENCYSHARES   QUEUES   HANDSIZE   QUEUELENGTHLIMIT   AGE
catch-all         Limited    5                          <none>   <none>     <none>             48m
exempt            Exempt     <none>                     <none>   <none>     <none>             48m
global-default    Limited    20                         128      6          50                 48m
leader-election   Limited    10                         16       4          50                 48m
node-high         Limited    40                         64       6          50                 48m
system            Limited    30                         64       6          50                 48m
workload-high     Limited    40                         128      6          50                 48m
workload-low      Limited    100                        128      6          50                 48m
```

245

# ACS

ACS = 245

--max-requests-inflight = 400 (default)

--max-mutating-requests-inflight = 200 (default)

workload-low = 100

(400 + 200) / 245 * 100 = 244 requests per second

https://kubernetes.io/docs/concepts/cluster-administration/flow-control/#suggested-configuration-objects

# ACS

Add another PriorityLevel with ACS 55

ACS = 300

(400 + 200) / 300 * 100 = 200 requests per second workload-low

(400 + 200) / 300 * 55 = 110 requests per second new level

https://kubernetes.io/docs/concepts/cluster-administration/flow-control/#suggested-configuration-objects

# Demo

# So What Was The Bug?

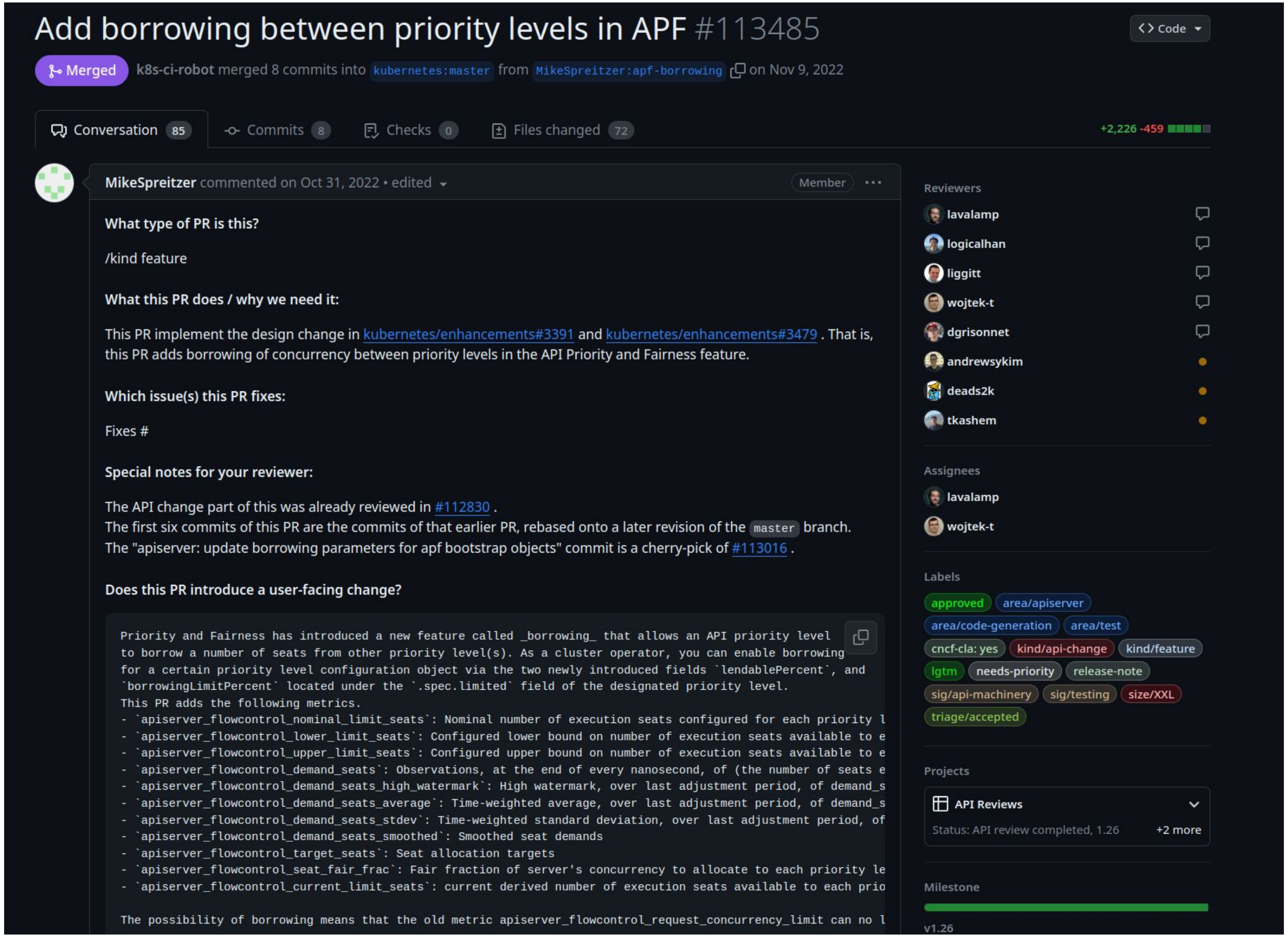
- Off cluster traffic was getting bucketed in `global-default`
- Fix was to create APF resources
- When would you want to do this?

# Hmm

- Why not 1.26?

https://kubernetes.io/docs/concepts/cluster-administration/flow-control/#suggested-configuration-objects

# Borrowing



Add borrowing between priority levels in APF #113485

Merged k8s-ci-robot merged 8 commits into `kubernetes:master` from `MikeSpreitzer:apf-borrowing` on Nov 9, 2022

MikeSpreitzer commented on Oct 31, 2022 • edited

**What type of PR is this?**

/kind feature

**What this PR does / why we need it:**

This PR implement the design change in kubernetes/enhancements#3391 and kubernetes/enhancements#3479 . That is, this PR adds borrowing of concurrency between priority levels in the API Priority and Fairness feature.

**Which issue(s) this PR fixes:**

Fixes #

**Special notes for your reviewer:**

The API change part of this was already reviewed in #112830 .
The first six commits of this PR are the commits of that earlier PR, rebased onto a later revision of the `master` branch.
The "apiserver: update borrowing parameters for apf bootstrap objects" commit is a cherry-pick of #113016 .

**Does this PR introduce a user-facing change?**

```
Priority and Fairness has introduced a new feature called _borrowing_ that allows an API priority level
to borrow a number of seats from other priority level(s). As a cluster operator, you can enable borrowing
for a certain priority level configuration object via the two newly introduced fields `lendablePercent`, and
`borrowingLimitPercent` located under the `.spec.limited` field of the designated priority level.
This PR adds the following metrics.
- `apiserver_flowcontrol_nominal_limit_seats`: Nominal number of execution seats configured for each priority l
- `apiserver_flowcontrol_lower_limit_seats`: Configured lower bound on number of execution seats available to e
- `apiserver_flowcontrol_upper_limit_seats`: Configured upper bound on number of execution seats available to e
- `apiserver_flowcontrol_demand_seats`: Observations, at the end of every nanosecond, of (the number of seats e
- `apiserver_flowcontrol_demand_seats_high_watermark`: High watermark, over last adjustment period, of demand_s
- `apiserver_flowcontrol_demand_seats_average`: Time-weighted average, over last adjustment period, of demand_s
- `apiserver_flowcontrol_demand_seats_stdev`: Time-weighted standard deviation, over last adjustment period, of
- `apiserver_flowcontrol_demand_seats_smoothed`: Smoothed seat demands
- `apiserver_flowcontrol_target_seats`: Seat allocation targets
- `apiserver_flowcontrol_seat_fair_frac`: Fair fraction of server's concurrency to allocate to each priority le
- `apiserver_flowcontrol_current_limit_seats`: current derived number of execution seats available to each prio

The possibility of borrowing means that the old metric apiserver_flowcontrol_request_concurrency_limit can no l
```

Reviewers

lavalamp
logicalhan
liggitt
wojtek-t
dgrisonnet
andrewsykim
deads2k
tkashem

Assignees

lavalamp
wojtek-t

Labels

approved area/apiserver area/code-generation area/test cncf-cla: yes kind/api-change kind/feature lgtm needs-priority release-note sig/api-machinery sig/testing size/XXL triage/accepted

Projects

API Reviews
Status: API review completed, 1.26 +2 more

Milestone

v1.26

# Changes in 1.26

- `borrowingLimitPercent`
- `lendablePercent`
- `nominalConcurrencyShares`
- NominalCL(i) = ceil( ServerCL * NCS(i) / sum_ncs ) sum_ncs = sum[limited priority level k] NCS(k)
- Bigger numbers mean a larger nominal concurrency limit, at the expense of every other Limited priority level.

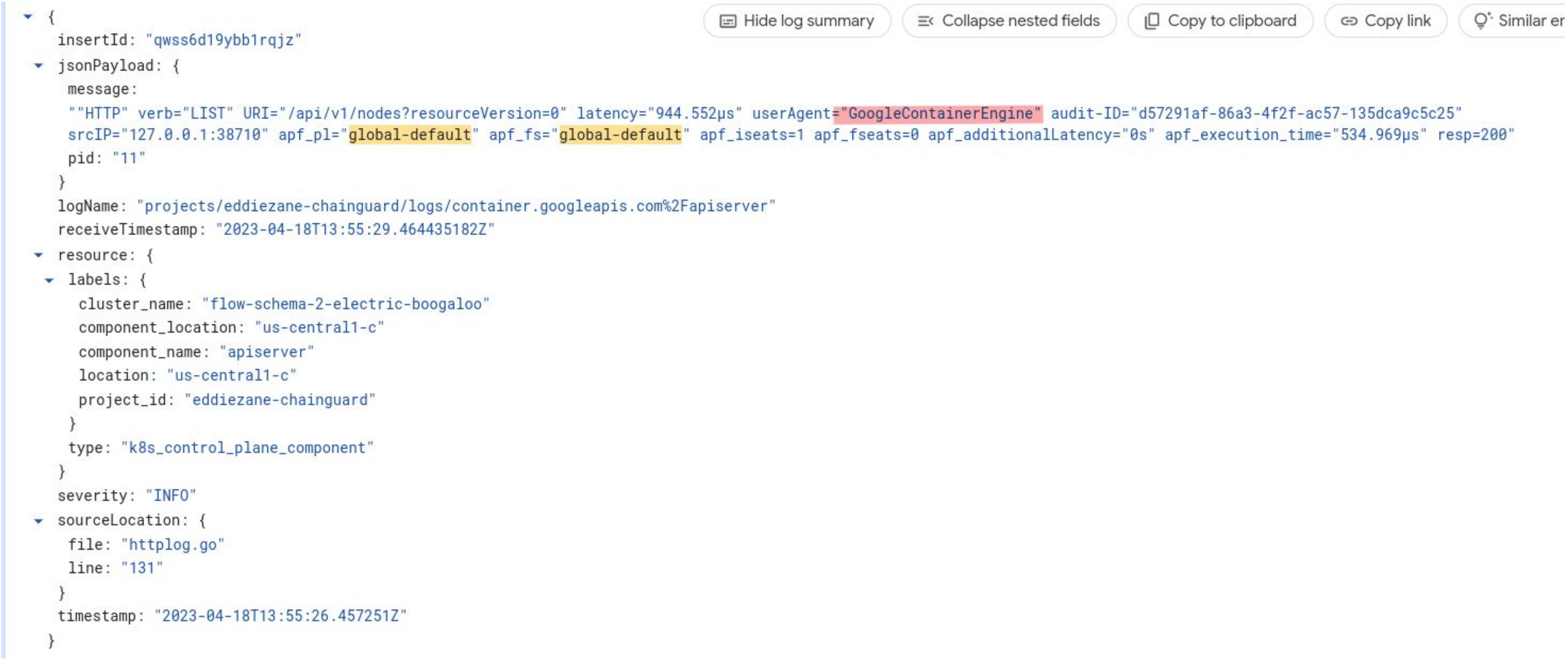
https://kubernetes.io/docs/concepts/cluster-administration/flow-control/#suggested-configuration-objects

# Hmm

- Why did the GKE Dashboard die?

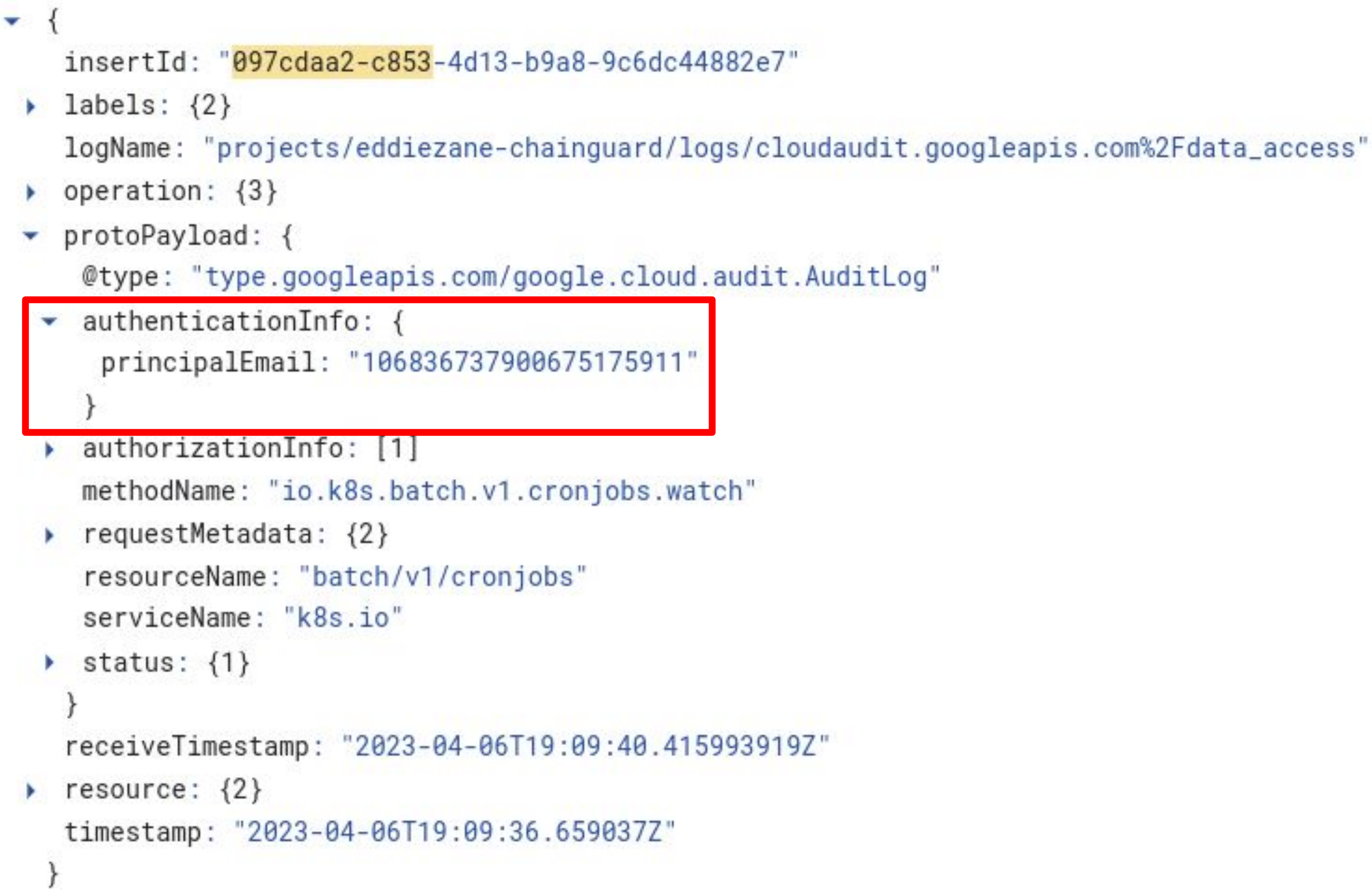
https://kubernetes.io/docs/concepts/cluster-administration/flow-control/#suggested-configuration-objects

# Ohhhhh



```
{
 insertId: "qwss6d19ybb1rqjz"
 jsonPayload: {
  message:
  ""HTTP" verb="LIST" URI="/api/v1/nodes?resourceVersion=0" latency="944.552µs" userAgent="GoogleContainerEngine" audit-ID="d57291af-86a3-4f2f-ac57-135dca9c5c25"
  srcIP="127.0.0.1:38710" apf_pl="global-default" apf_fs="global-default" apf_iseats=1 apf_fseats=0 apf_additionalLatency="0s" apf_execution_time="534.969µs" resp=200"
  pid: "11"
 }
 logName: "projects/eddiezane-chainguard/logs/container.googleapis.com%2Fapiserver"
 receiveTimestamp: "2023-04-18T13:55:29.464435182Z"
 resource: {
  labels: {
   cluster_name: "flow-schema-2-electric-boogaloo"
   component_location: "us-central1-c"
   component_name: "apiserver"
   location: "us-central1-c"
   project_id: "eddiezane-chainguard"
  }
  type: "k8s_control_plane_component"
 }
 severity: "INFO"
 sourceLocation: {
  file: "httplog.go"
  line: "131"
 }
 timestamp: "2023-04-18T13:55:26.457251Z"
}
```

# Sidequest

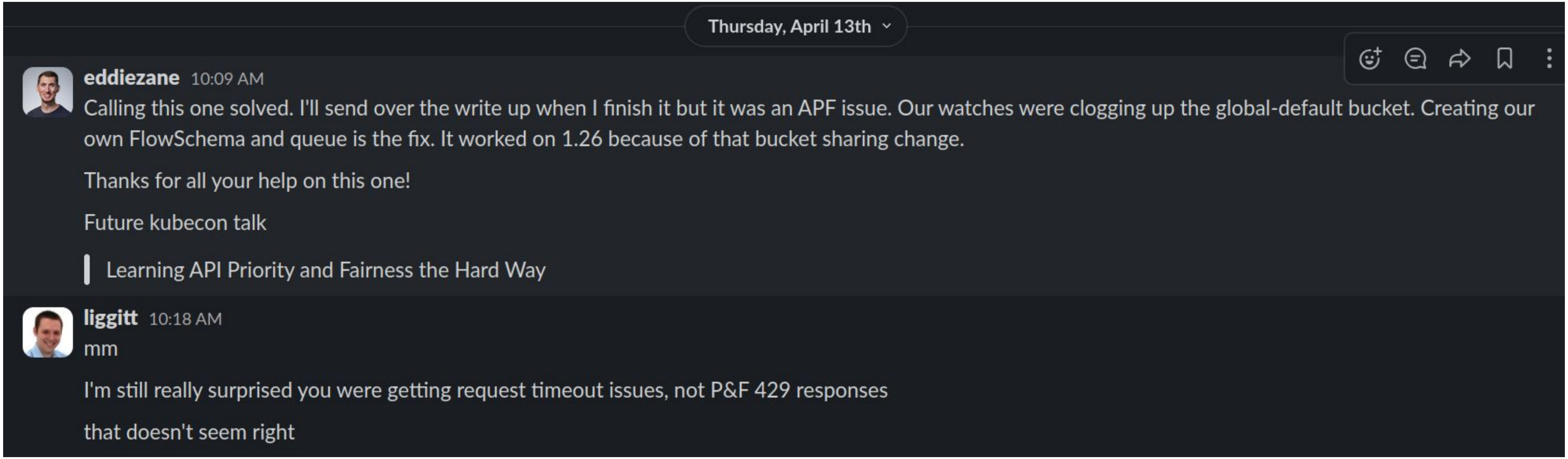


```
▾ {
    insertId: "097cdaa2-c853-4d13-b9a8-9c6dc44882e7"
  ▸ labels: {2}
    logName: "projects/eddiezane-chainguard/logs/cloudaudit.googleapis.com%2Fdata_access"
  ▸ operation: {3}
  ▾ protoPayload: {
      @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    ▾ authenticationInfo: {
        principalEmail: "106836737900675175911"
      }
    ▸ authorizationInfo: [1]
      methodName: "io.k8s.batch.v1.cronjobs.watch"
    ▸ requestMetadata: {2}
      resourceName: "batch/v1/cronjobs"
      serviceName: "k8s.io"
    ▸ status: {1}
    }
    receiveTimestamp: "2023-04-06T19:09:40.415993919Z"
  ▸ resource: {2}
    timestamp: "2023-04-06T19:09:36.659037Z"
  }
```

# More More Lessons

- Being clever means you need to do more research
- Need people to test new things and give feedback
- Read the docs and release notes
- Get involved upstream!!

# To Be Continued?



Thursday, April 13th

**eddiezane** 10:09 AM

Calling this one solved. I'll send over the write up when I finish it but it was an APF issue. Our watches were clogging up the global-default bucket. Creating our own FlowSchema and queue is the fix. It worked on 1.26 because of that bucket sharing change.

Thanks for all your help on this one!

Future kubecon talk

> Learning API Priority and Fairness the Hard Way

**liggitt** 10:18 AM

mm

I'm still really surprised you were getting request timeout issues, not P&F 429 responses

that doesn't seem right

# Shoutouts

- Jordan Liggitt (Google)
- Mike Danese (Google)
- Billy Lynch (Chainguard)

# Thank You!

- Questions?
- @eddiezane

**Please scan the QR Code ->
to leave feedback on this session**