Exporting Kubernetes Event Objects for Better Observability

KubeCon/CloudNativeCon North America 2019 San Diego

MUSTAFA AKIN | AHMET ŞEKER | SRE @ ATLASSIAN OPSGENIE

Agenda

K8s Event API

Interesting Kubernetes Events

Proper Monitoring & Alerting

Event Exporter Tool

Kubernetes Event API

What is an Event and why they are thrown?

“Event is a report of an event somewhere in the cluster”

Kubernetes Source Code

“Event is a report of an event somewhere in the cluster”

Kubernetes Source Code

AN EVENT EXAMPLE

```yaml
apiVersion: v1
kind: Event
metadata:
 name: xxx.15d3018d822b6959
 namespace: default
count: 1
eventTime: null
type: Normal
message: pulling image "my-cool-app:0.1"
reason: Pulling
firstTimestamp: "2019-11-01T10:00:02Z"
involvedObject:
 apiVersion: v1
 fieldPath: spec.containers{container}
 kind: Pod
 name: my-app
 namespace: default
source:
 component: kubelet
 host: ip-10-35-44-212.us-west-2.compute.internal
```

Event Components

Message

A human-readable description of the status of this operation

Reason

Short, machine understandable string, in other words: Enum

Type

Currently holds only Normal & Warning, but custom type can be given if desired.

Involved Object

The object that this event is about, like Pod, Deployment, Node etc.

Source

The component reporting this event, short machine understandable string. i.e kube-scheduler

Count

The number of times the event has occurred

When are Events published?

Informational

Pod scheduled, images pulled, Node healthy, Deployment is updated, ReplicaSet is scaled, Container is killed

Warnings

Pods have errors, persistent volumes are not bound yet

Errors

Node is down, Persistent Volume is not found, Cannot create a LoadBalancer in the Cloud Provider

How can you publish custom events?

Directly

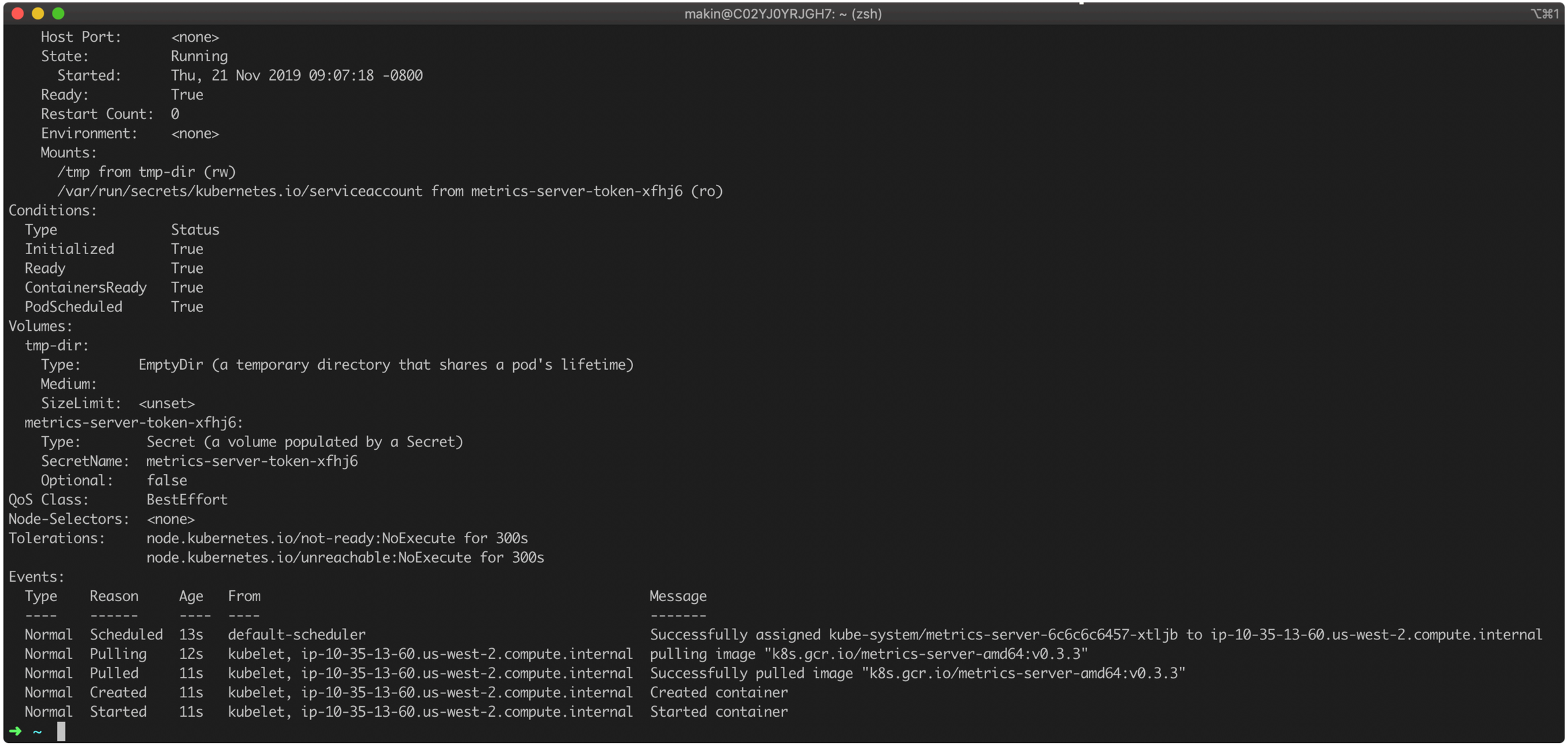
Use the REST API directly, or with a SDK (i.e. client-go) to create the Event Object with required fields.

Event Recorder

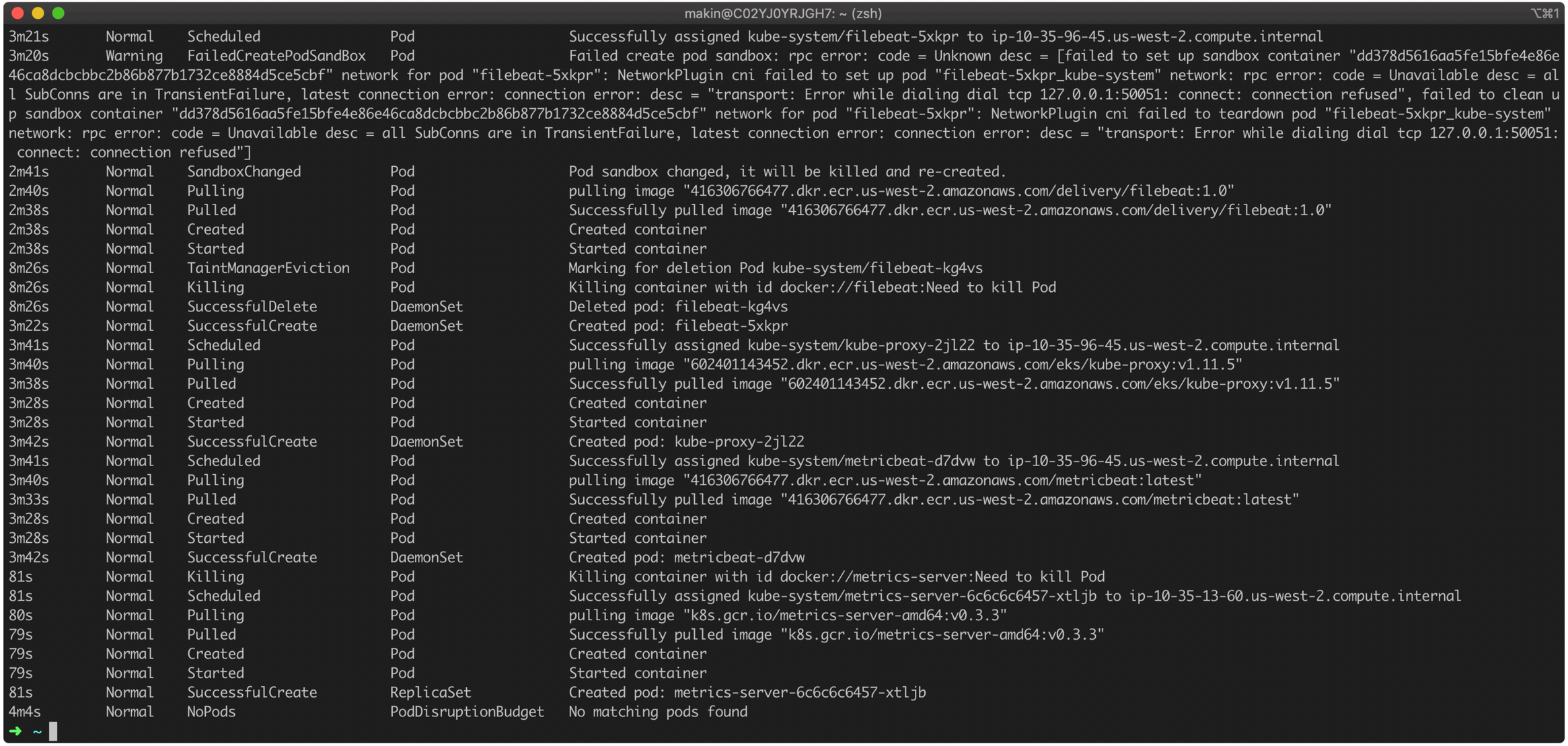
A helper for K8s controllers create events easier.

Also see another talk for more information:
Emitting, Consuming, and Presenting: The Event Lifecycle - Jesse Dearing, VMware
https://kccncna19.sched.com/event/UaY1

KUBECTL DESCRIBE POD



KUBECTL GET EVENTS



For scalability issues on etcd, events are stored only for 1-hour by default.

Usually a separate etcd cluster is deployed to reduce the load.

Interesting Kubernetes Events

If a tree falls in a forest, and no one is around to hear it, does it make a sound?

THE ONES YOU PROBABLY KNOW

Unhealthy

Pulled

Started

Scaled

Preempted

Starting

Failed

SuccesfulDelete

INFREQUENT EVENTS

FailedCreatePodSandBox

NetworkNotReady

LeaderElection

FailedAttachVolume

ScaleDownFailed

ImageGCFailed

FailedToUpdateEndpoint

TaintManagerEviction

SOME EVENTS FROM THE KUBERNETES SOURCE CODE

ProvisioningFailed

FailedToCreateEndpoint

ClusterIPNotValid

UpdateLoadBalancerFailed

CheckLimitsForResolvConf

PortRangeFull

InvalidDiskCapacity

ClusterIPAlreadyAllocated

InvalidEnvironmentVariableNames

WaitForFirstConsumer

FailedNeedsStart

FailedPlacementReason

MissingClusterDNS

TLSConfigChanged

CalculateExpectedPodCountFailed

2K

Events hourly for a 10-node not-so-busy stable cluster

700k

Events hourly for a 200-node for busy dev cluster

70

Unique Reasons

DISTRIBUTION OF EVENTS PER INVOLVED OBJECT

Mostly Pods

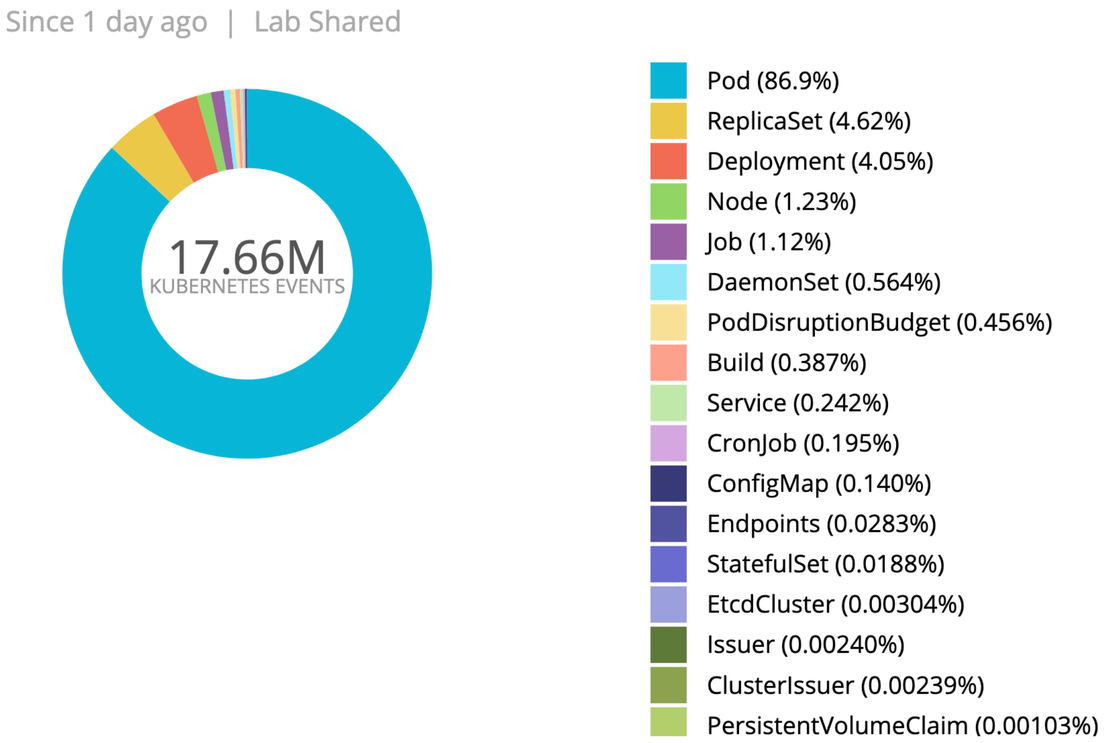
They are the unit of computation and there are probably lots of replicas

ReplicaSet & Deployments

Since they are main Pod controllers, they publish a lot of events

Node

They come and go with *cluster-autoscaler* and their health might fluctuate if you are not careful enough.



DISTRIBUTION OF EVENTS PER REASON / STABLE CLUSTER

Pod Creation

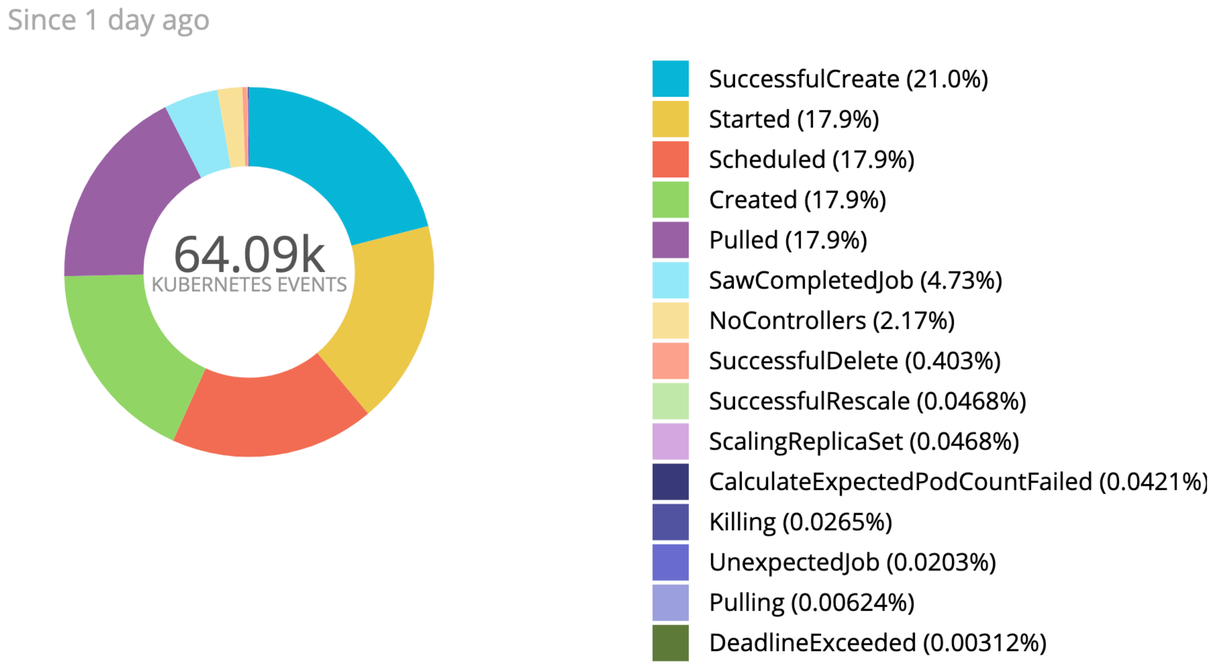
SuccessfulCreate, Started, Scheduled, Created, Pulled is all related to new Pods

Jobs

CronJob & Jobs also states are published as events.

Node

They come and go with *cluster-autoscaler* and their health might fluctuate if you are not careful enough.



DISTRIBUTION OF EVENTS PER REASON / DEV CLUSTER

Unhealthy & Readiness

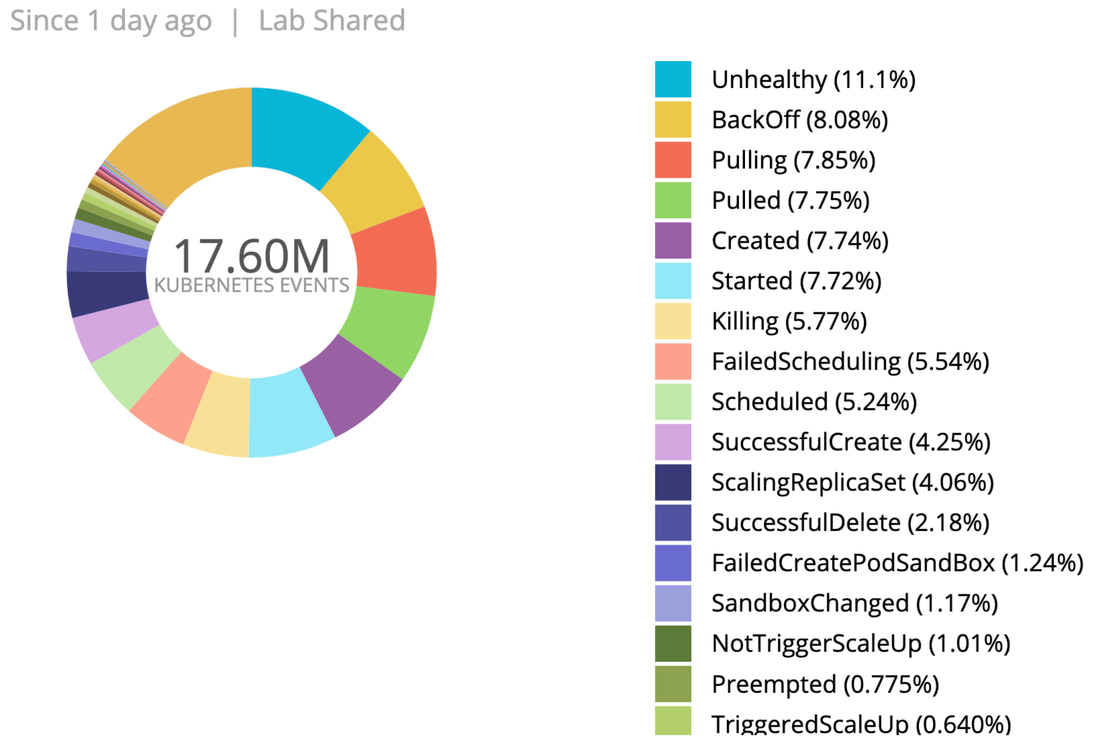
Readiness probes might need tweaking in dev environments.

Schedule Fails & Pre-emption

High-volume pod schedule and keeping costs

Sandbox

CNI, Docker & Kubelet bugs happen at scale and you can miss them.



Proper Monitoring and Alerting

What should be an alert and notification from those K8s Events?

EVENTS –> ALERT

Alert should be designed for human consumption.

EVENTS –> ALERT

They should be structured, precise, actionable and noise-free.

“Back-off Restarting Failed Container”

Involved Object: Pod name

This event has happened for a Pod and should be included in the message.

Namespace

We utilize namespaces for organization and segregation and it can specify importance.

Labels

Events normally do not have labels, but we fetch them in our tool for embedding more information, so they can be routed correctly.

MONITORING & OBSERVABILITY

We can also extract information and metrics from the events for extra observability.

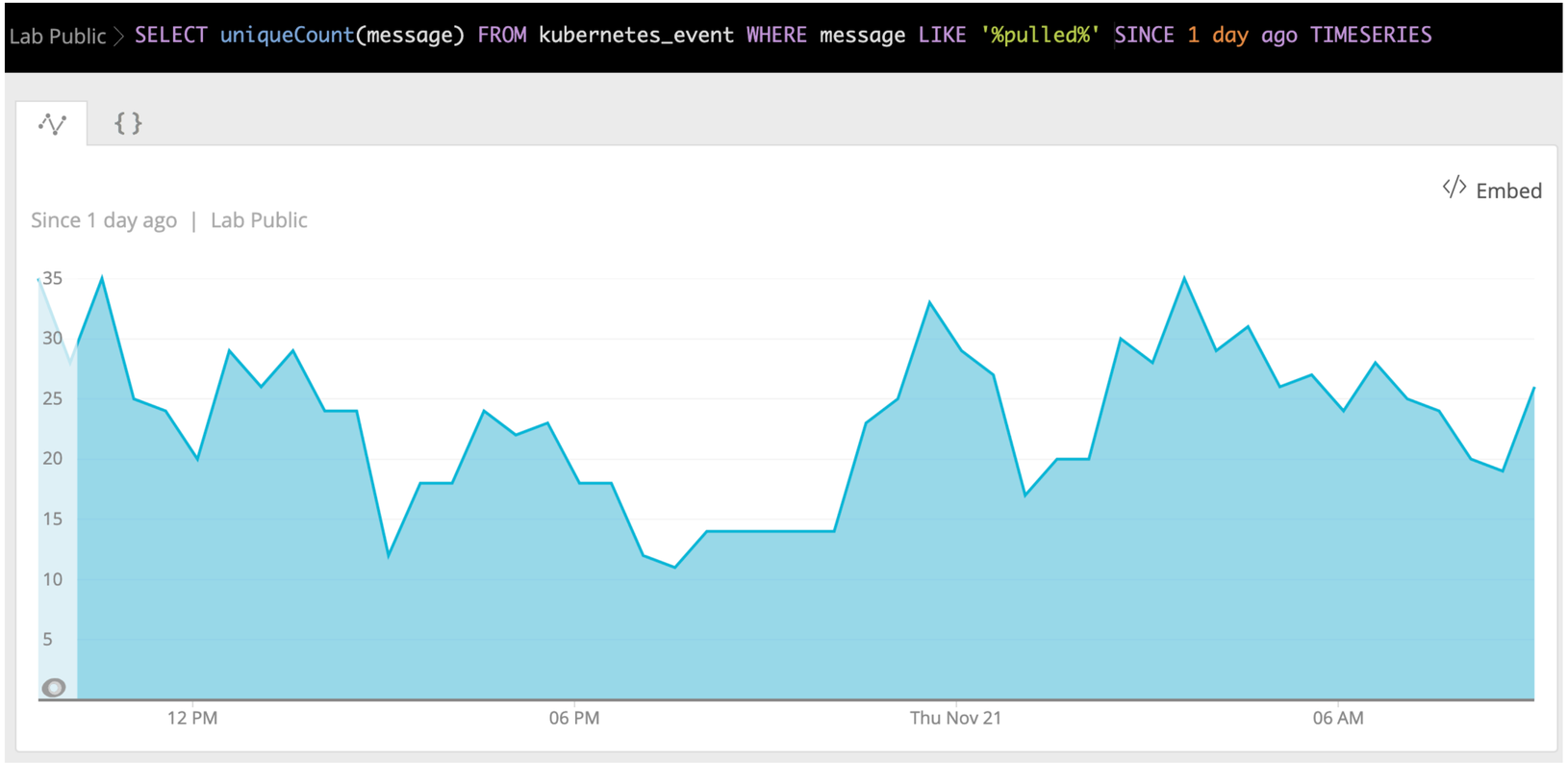
Alerts

Usually Warnings and some Information can be transformed into alerts for human consumption.

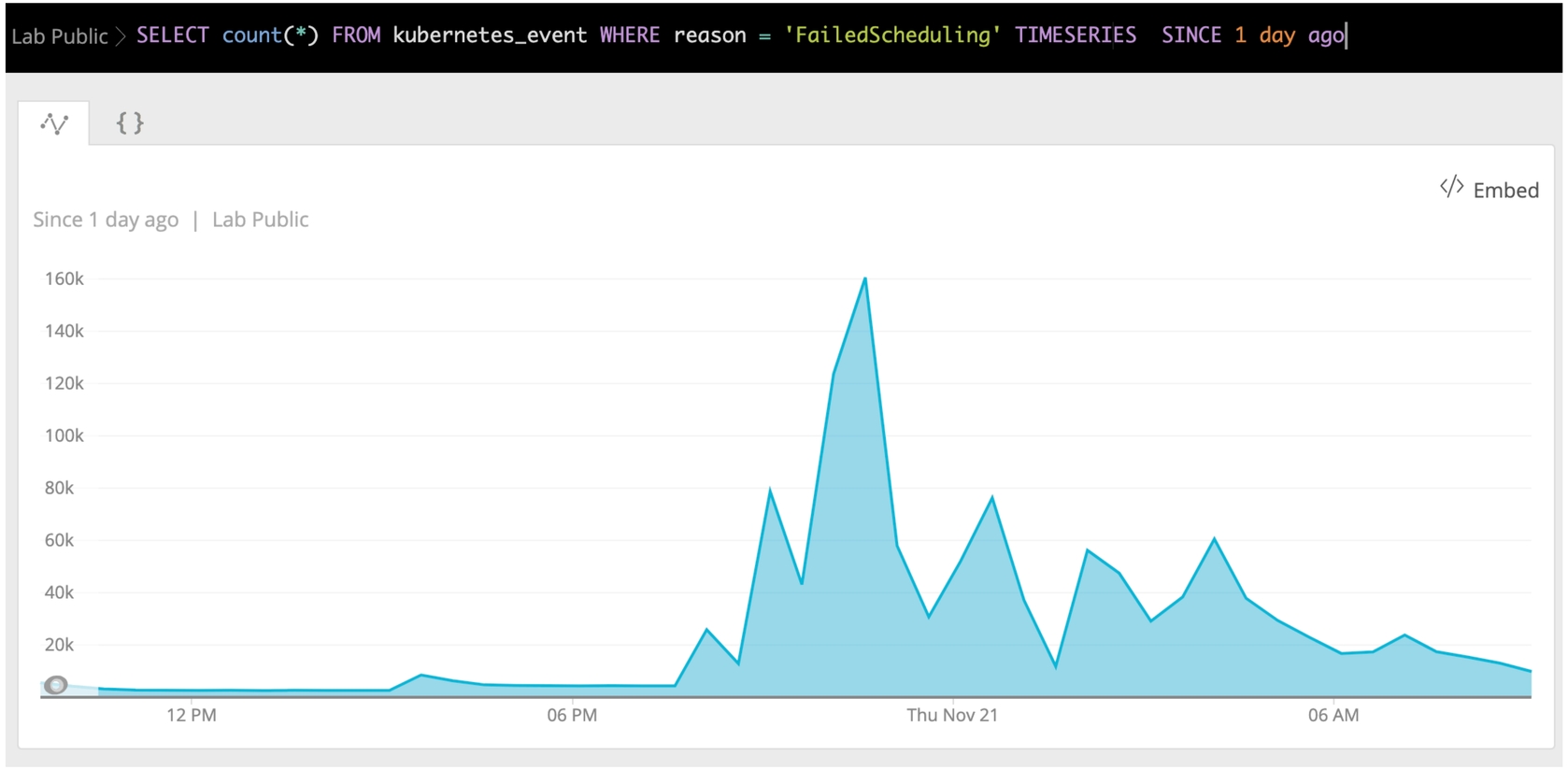
Monitoring

Aggregation & filtering of many events over a time-series can give you uncharted information about the state of cluster.

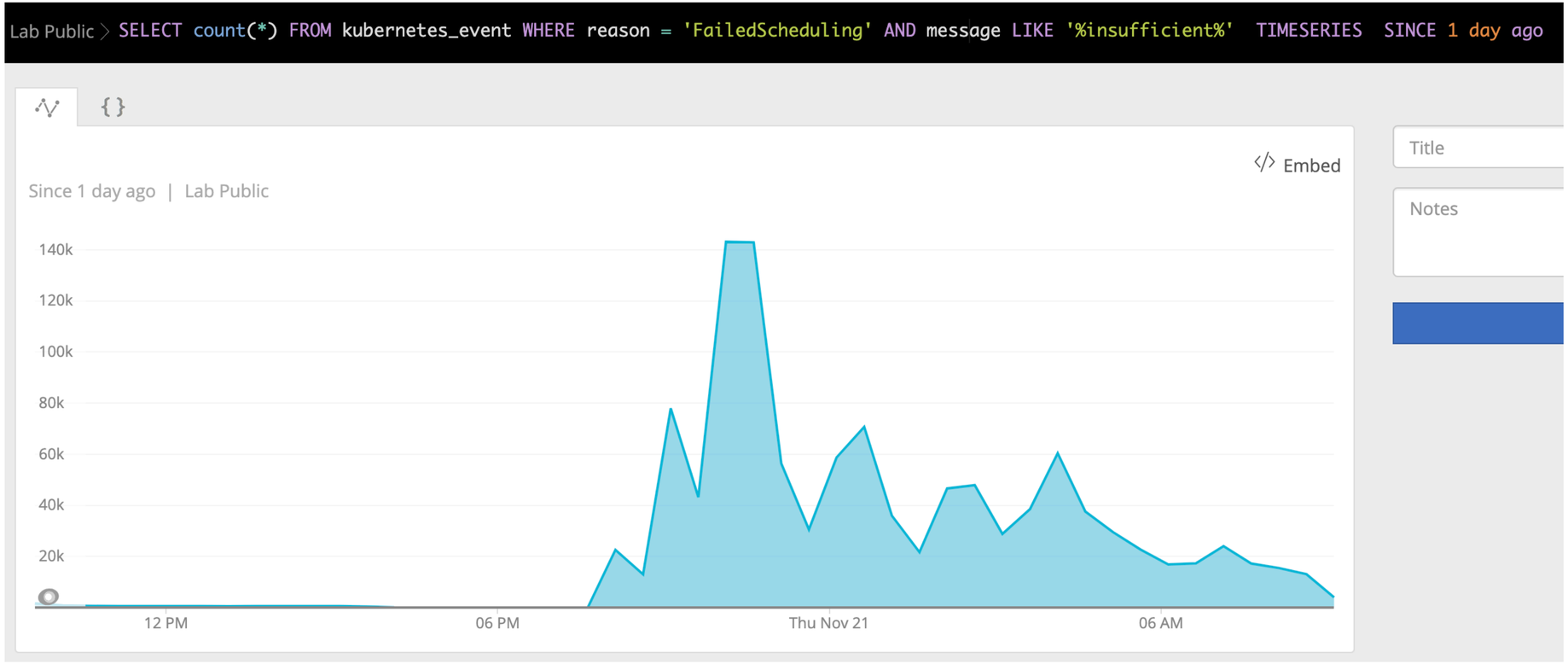
How many different images are pulled hourly?



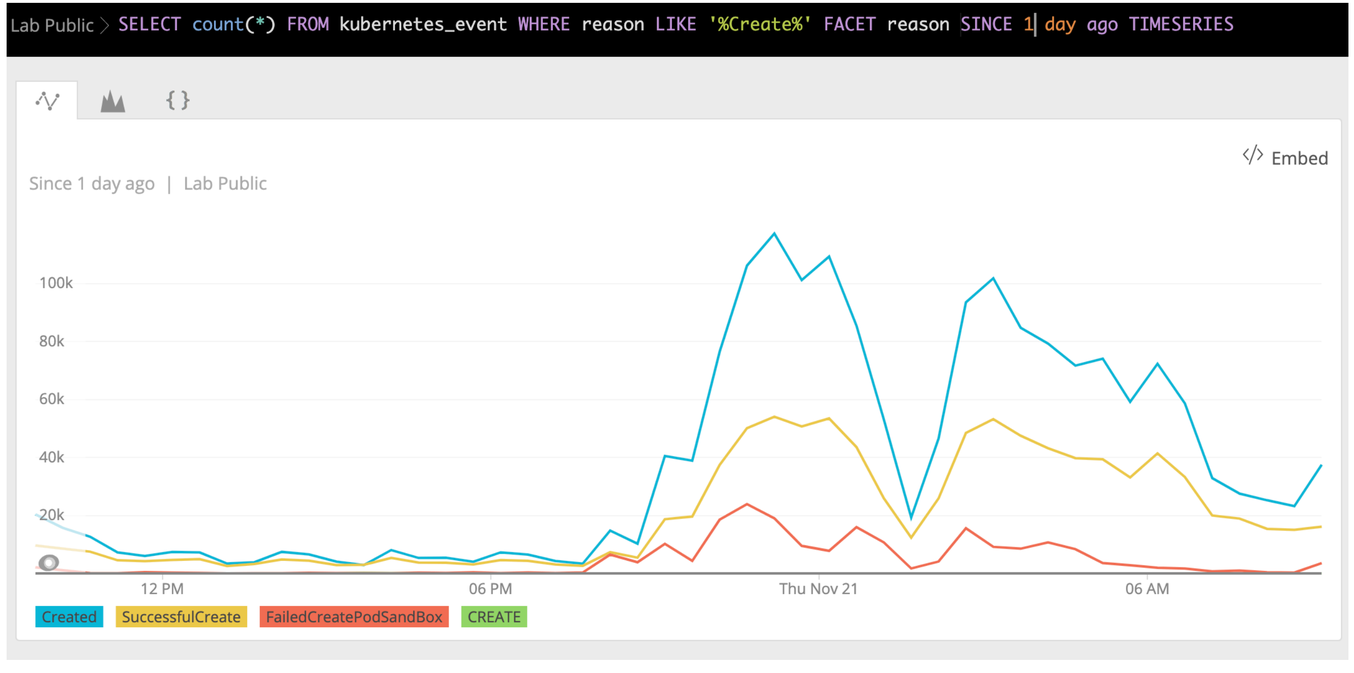
How often pods cannot be scheduled?



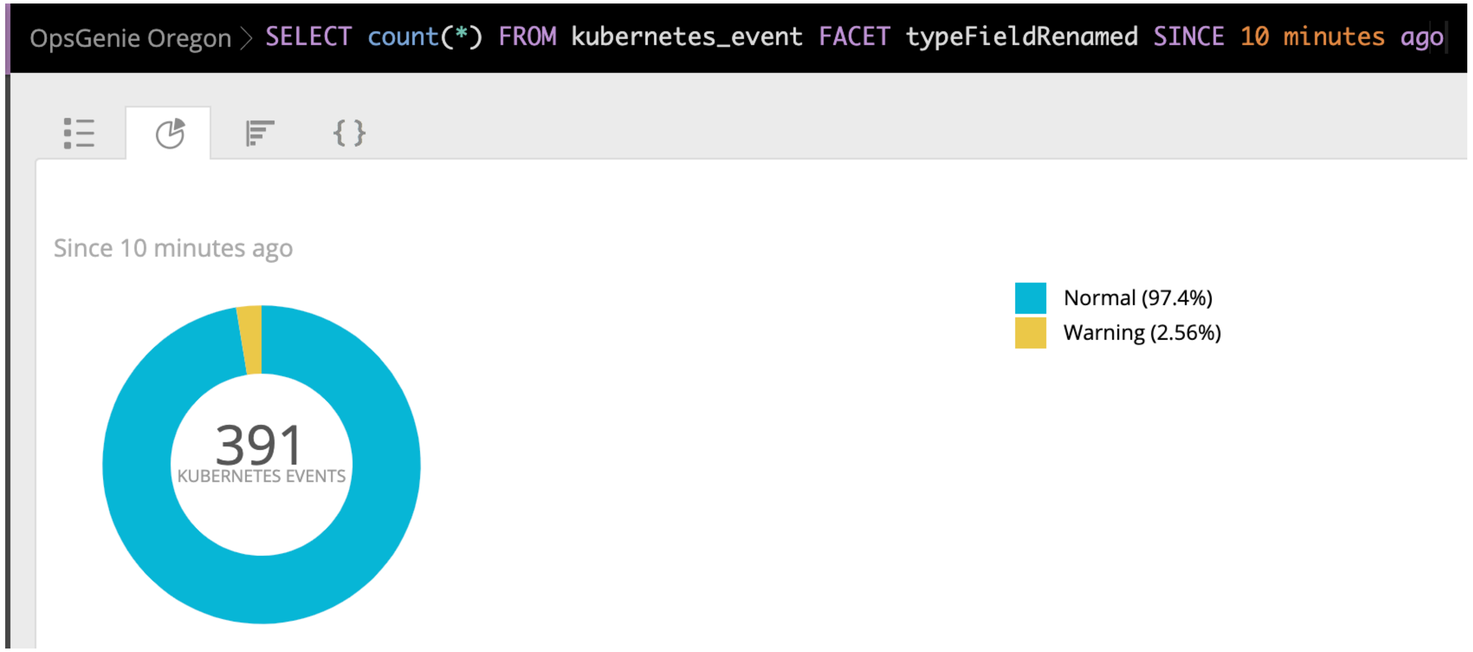
Are they related to capacity errors?



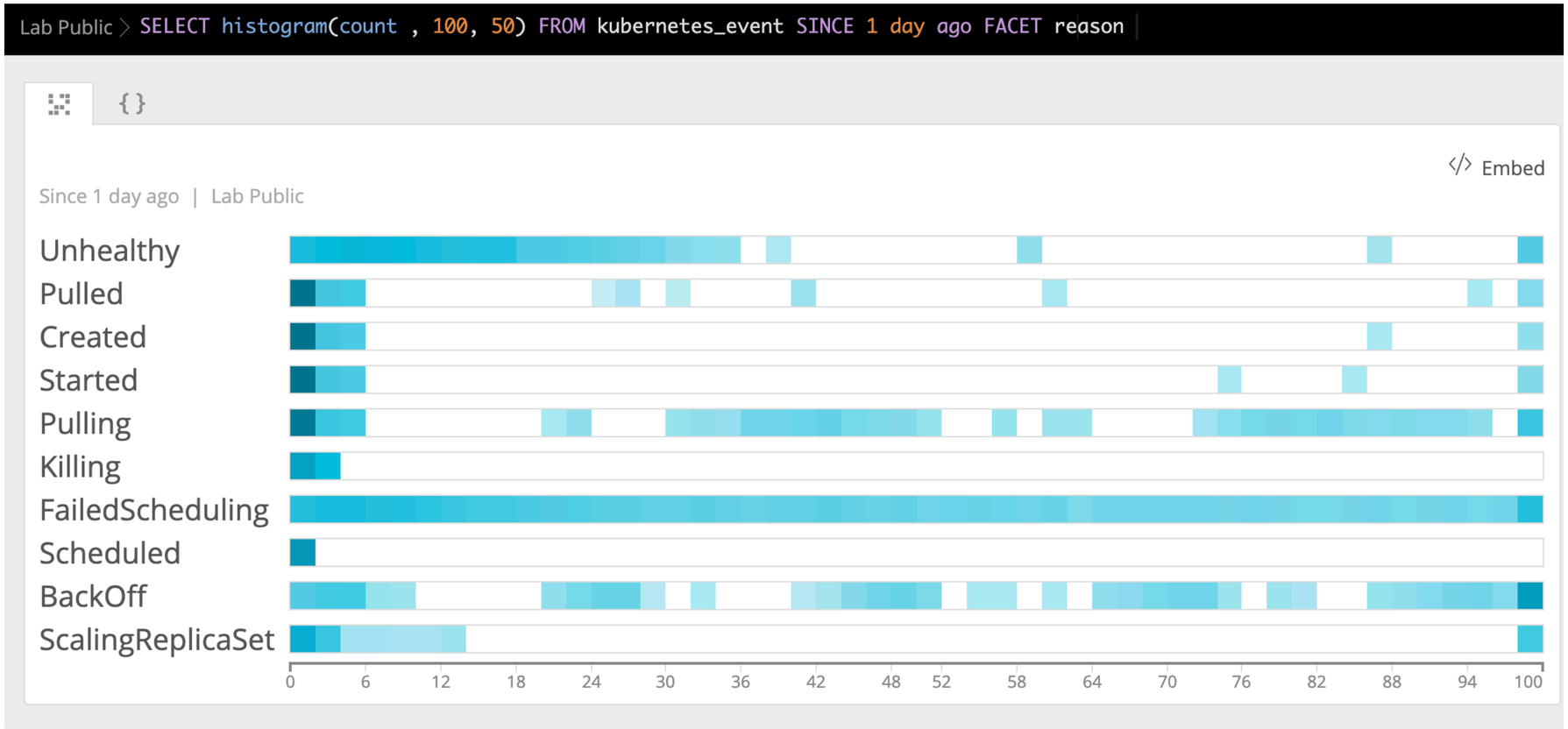
What is the distribution of Pods, Deployments Created/ Updated through out the day?



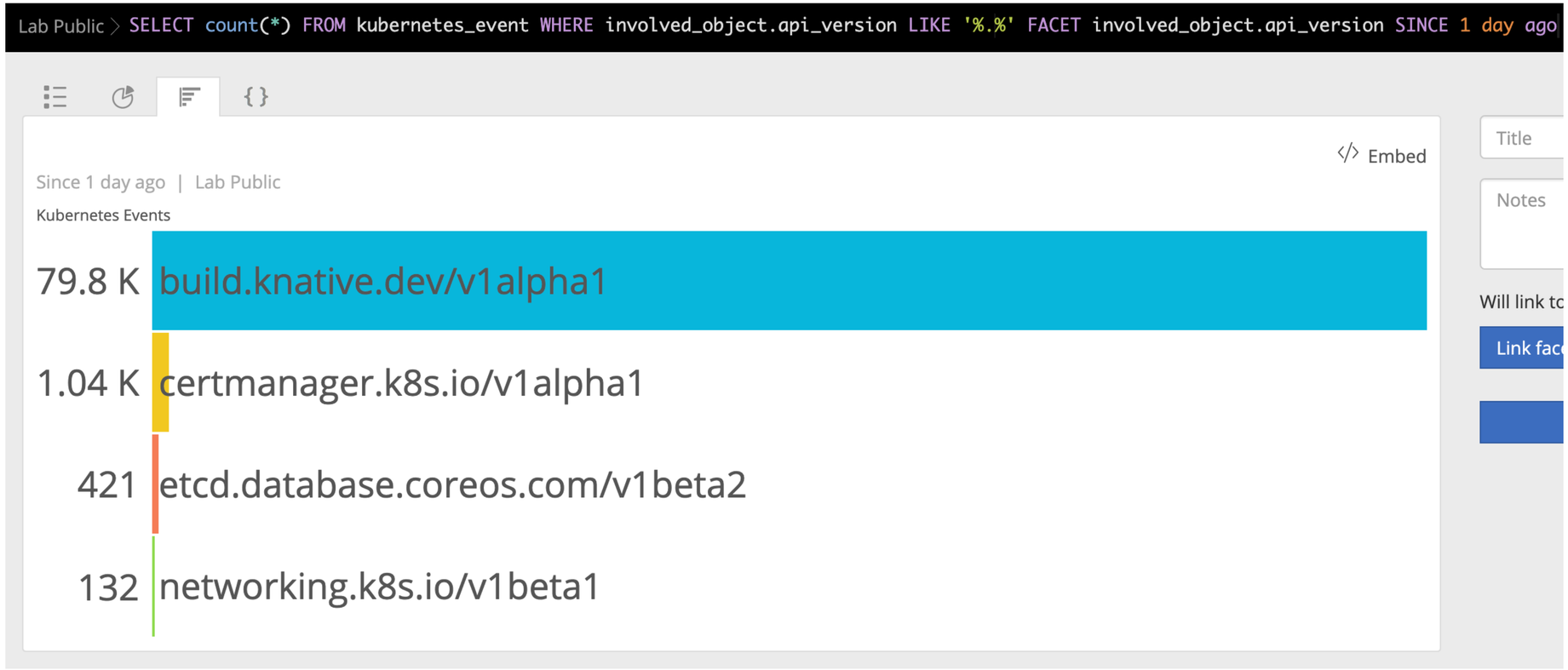
Are there significant amount of ‘Warning’ events right now?



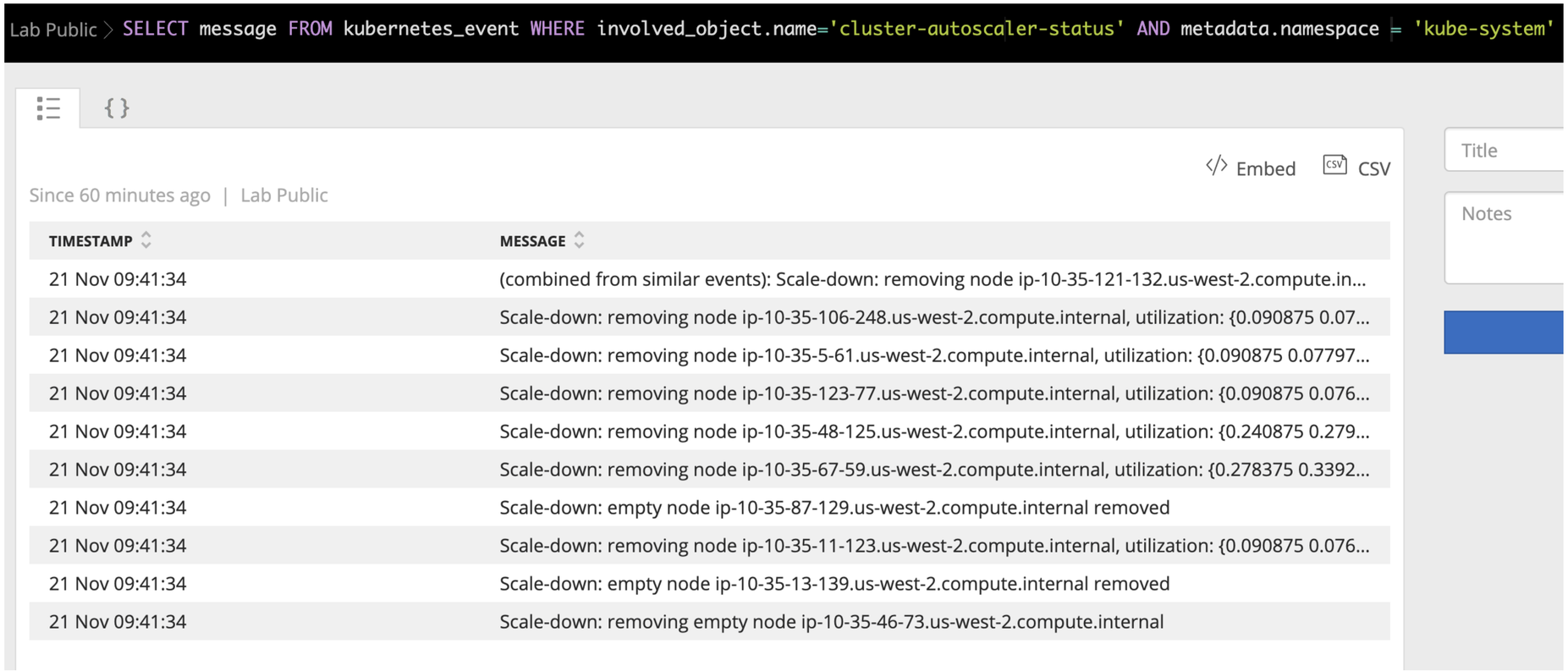
What does the distribution of event counts per reason?



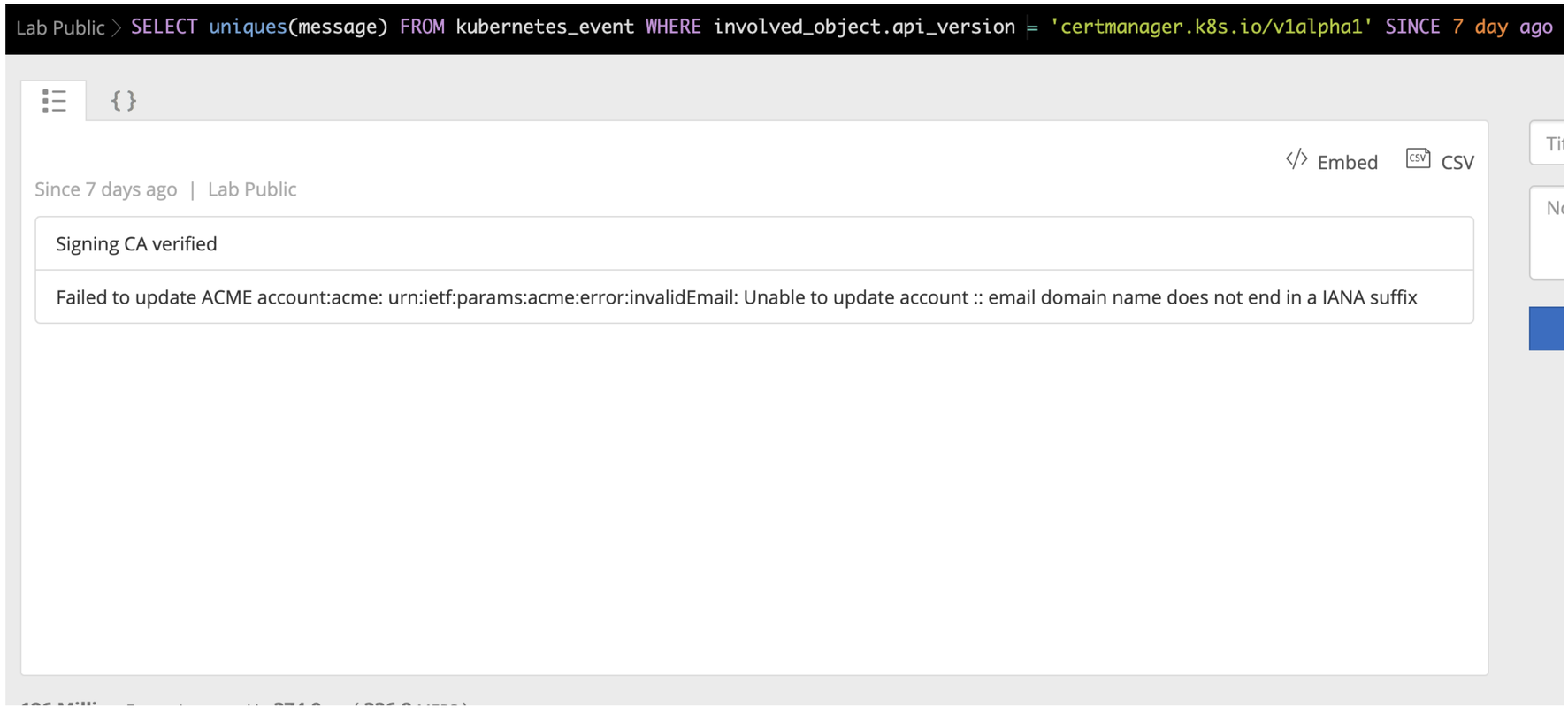
How many times did your *Custom Event* occur?



Is Cluster Autoscaler publishing interesting events?



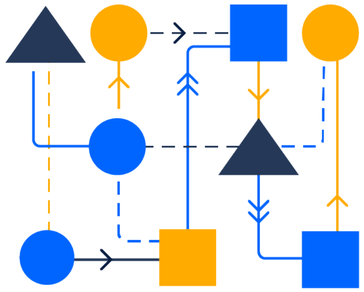
Is cert-manager able to renew certificates properly?



Event Exporter Tool

The implementation details and output types

Features



Event Routing & Filtering

Events are received from a single endpoint and filtered and routed based on their fields with regexes to route relevant events.

Multiple Outputs

Each output has different use cases, so tool allows using all of them with the routing rules to avoid deploying multiple instances .

Payload Customization

The pushed data can be customized to fit custom needs so that it can be easily embedded in the monitoring stack of many users.

Outputs

Reporting

When you want to export more data

elasticsearch

APACHE kafka

Google BigQuery

splunk>

amazon KINESIS FIREHOSE

Alerting

The critical events for the eyes of the on-call

ATLASSIAN Opsgenie

Notification

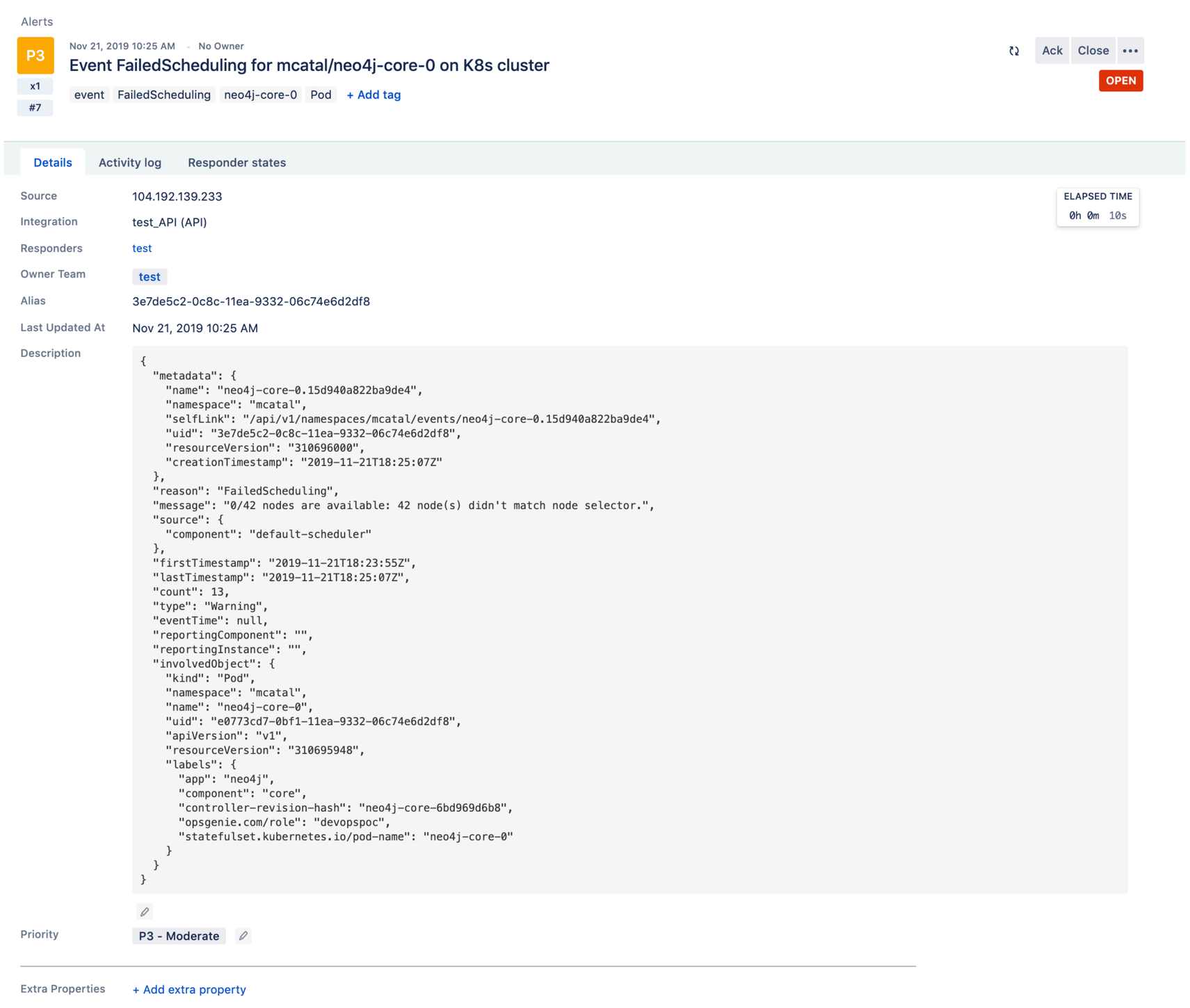
Push some of events for notification, extra processing

slack

amazon SNS

webhooks

An example Alert in Opsgenie



Implementation

Watcher

Writing a Kubernetes resource watcher requires some care

Generic Client

Events are enriched with objects labels to be used in routing and filtering

Output buffering

Many types of outputs, we've tried to utilized goroutines efficiently.

Configuration

Routing

Outputs

Payload Customization

```
route:
  match:
    - receiver: dump
 routes:
    - drop:
        - namespace: test*
        - type: Normal
    - match:
        - receiver: slack
          kind: Pod
        - receiver: alert
          kind: Pod
          namespace: prod
          reason: “Failed*"
```

Configuration

Routing

Outputs

Payload Customization

```
- name: personal-message
  slack:
    apiKey: "xoxo-12345"
    channel: "{{ .InvolvedObject.Labels.Owner }}"
    message: "Your pod has a msg {{ .InvolvedObject.Name }}”

- name: dump
  elasticsearch:
    addresses:
      - http://localhost:9200
    index: kuberenetes-events

- name: high-priority-alert
  opsgenie:
    apikey: xxx
    priority: "P3"
    message: "Event {{ .Reason }} for {{ .InvolvedObject.Namespace }}/
{{ .InvolvedObject.Name }} on K8s cluster"
    alias: "{{ .UID }}"
    description: "<pre>{{ toPrettyJson . }}</pre>"
    tags:
      - "event"
      - "{{ .Reason }}"
      - "{{ .InvolvedObject.Kind }}"
      - "{{ .InvolvedObject.Name }}"
```

Configuration

Routing

Outputs

Payload Customization

```
- name: appMetric
  kinesis:
    region: us-west-2
    streamname: applicationMetric
    layout:
      region: "us-west-2"
      eventType: "kubeevent"
      createdAt: "{{ .GetTimestampMs }}"
      details:
        message:   "{{ .Message }}"
        reason:    "{{ .Reason }}"
        type:      "{{ .Type }}"
        count:     "{{ .Count }}"
        kind:      "{{ .InvolvedObject.Kind }}"
        name:      "{{ .InvolvedObject.Name }}"
        namespace: "{{ .Namespace }}"
        component: "{{ .Source.Component }}"
        host:      "{{ .Source.Host }}"
        labels:    "{{ toJson .InvolvedObject.Labels}}"
```

Where did this project come from?

Attended KubeCon ’19 Barcelona

We loved everyone sharing experiences and their tooling in an open and welcoming environment

Open Source an In-House Project

We already have many tools to improve our own observability and wanted to share our experience with the whole world as a generic tool.

More to Come

We loved open-sourcing our stuff to share with the community, and we are working on sharing our more internal projects.

Next: *Alternative Kubernetes Dashboard, Golang Batching Library*

Thanks for joining us!
Any questions, comments?

https://github.com/opsgenie/kubernetes-event-exporter

Meet us at the booth!

ATLASSIAN BOOTH G20