KubeCon | CloudNativeCon

Europe 2023

KubeCon | CloudNativeCon

Europe 2023

Smarter Golden Signals

Anusha Ragunathan & Venkata Gunapati, Intuit Inc

Agenda

- Background
- Cluster Golden Signals
- Anomaly Detection
- Numaproj
- Demo & Takeaways

KubeCon | CloudNativeCon
Europe 2023

Background

Background: Intuit and infra at a glance

INTUIT
turbotax

INTUIT
quickbooks

INTUIT
creditkarma

INTUIT
mailchimp

Background: Intuit and infra at a glance

275+
Clusters

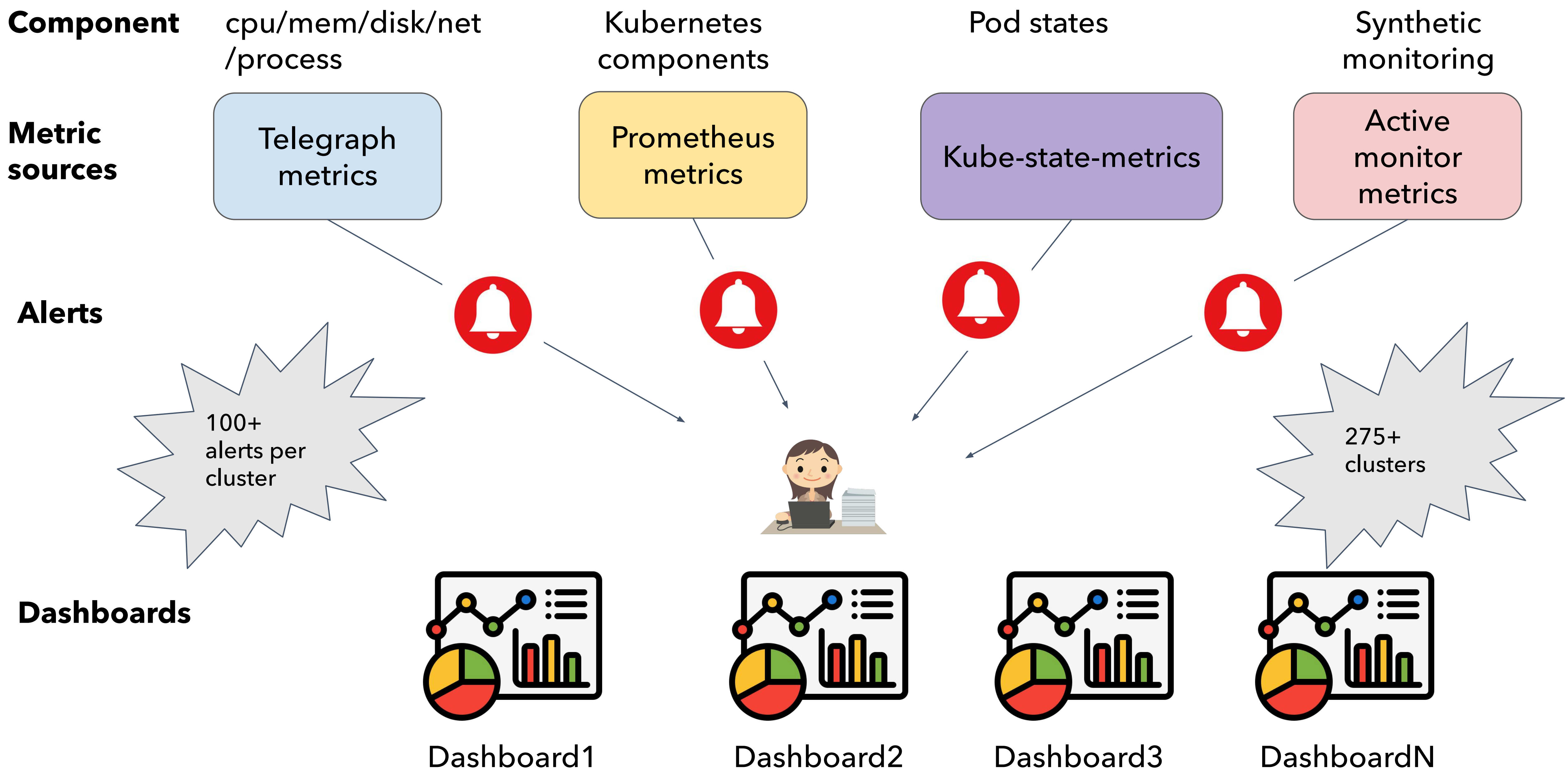
20500+
Namespaces

2500
production
Services

900+
Teams

6000+
Developers

Platform Engineer Woes - Part 1



Component

cpu/mem/disk/net /process

Kubernetes components

Pod states

Synthetic monitoring

Metric sources

Telegraph metrics

Prometheus metrics

Kube-state-metrics

Active monitor metrics

Alerts

100+ alerts per cluster

275+ clusters

Dashboards

Dashboard1

Dashboard2

Dashboard3

DashboardN

Platform Engineer Woes - Part 2

Incident

There are one of more impacted services in this incident. Are the clusters running them healthy? What alerts should I look at?

How can I quickly determine whether the issue is a service issue vs a cluster issue ?

How do I determine the blast radius of the incident? How many other services or clusters could affected by the incident?

Platform Engineer Wants

I want to reduce the MTTD and MTTR during an incident!

I want less false positives and false negatives from alerts.

I want a few good quality signals from clusters.

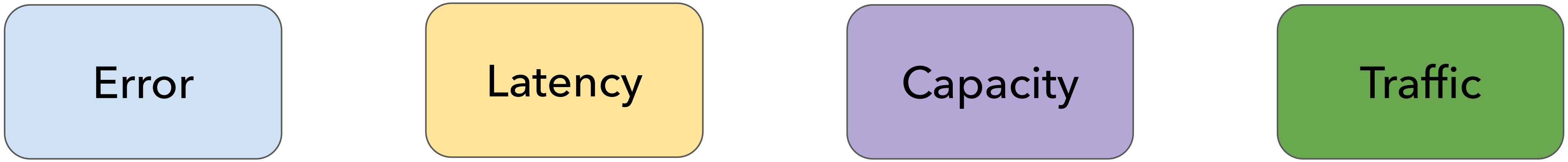
KubeCon | CloudNativeCon
Europe 2023

Cluster Golden Signals

Service Owner Concerns

P0 Platform requirements	Kubernetes components
Availability	Cluster Control Plane (ex: API server, etcd) Cluster Networking (ex:DNS, CNI)
Scale	Cluster AutoScaling HPA
Correctness	Cluster Authentication Cluster Networking (Packet Loss, Latency) Cluster Critical Addons

Cluster Golden Signals



Error

Latency

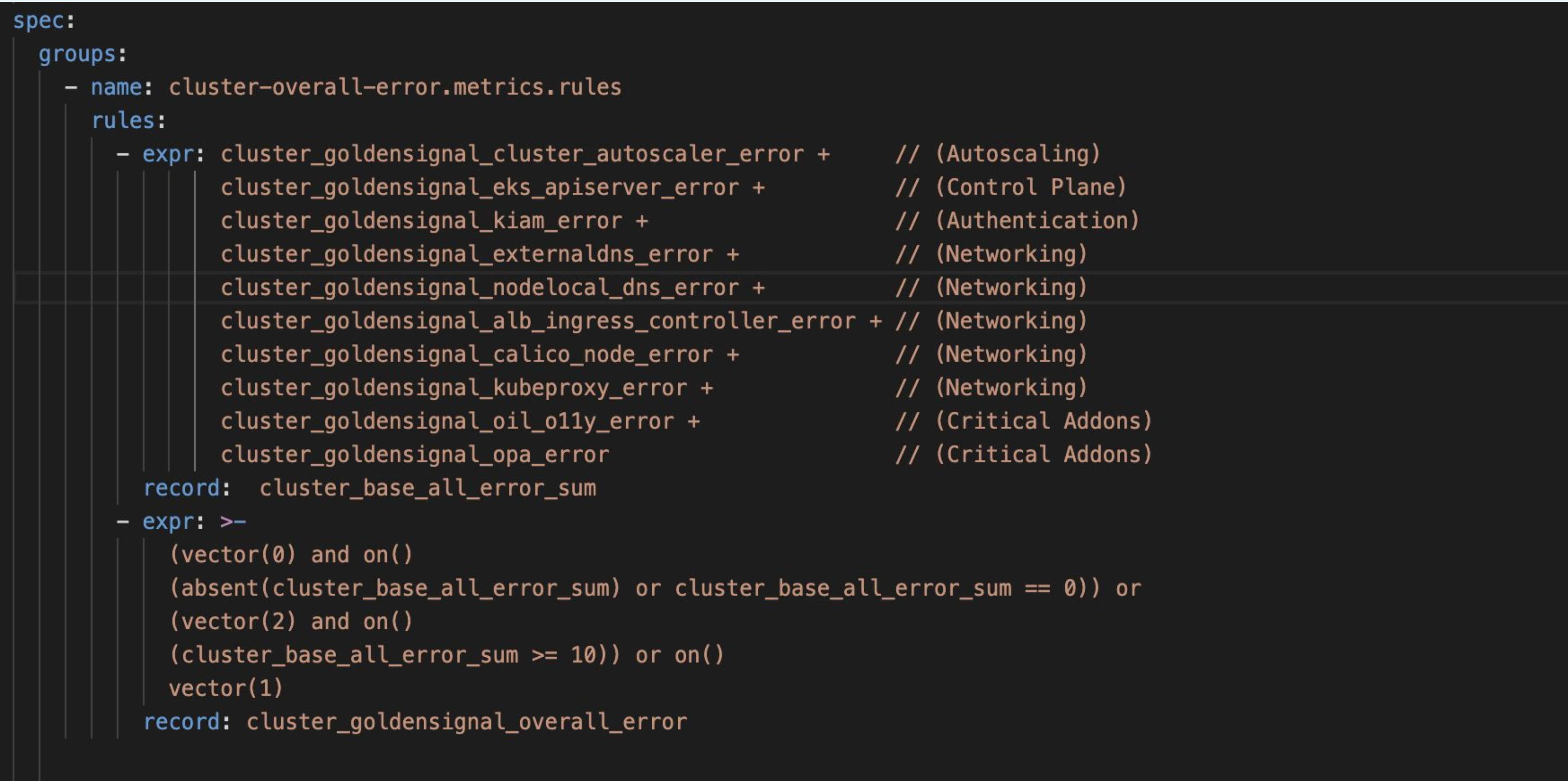
Capacity

Traffic

Each Cluster Golden Signal is reported and can have one of the following values:

- Healthy: All components are healthy
- Degraded: At least one component is degraded
- Critical: At least one component is critical

Cluster Golden Signals: Errors, a closer look

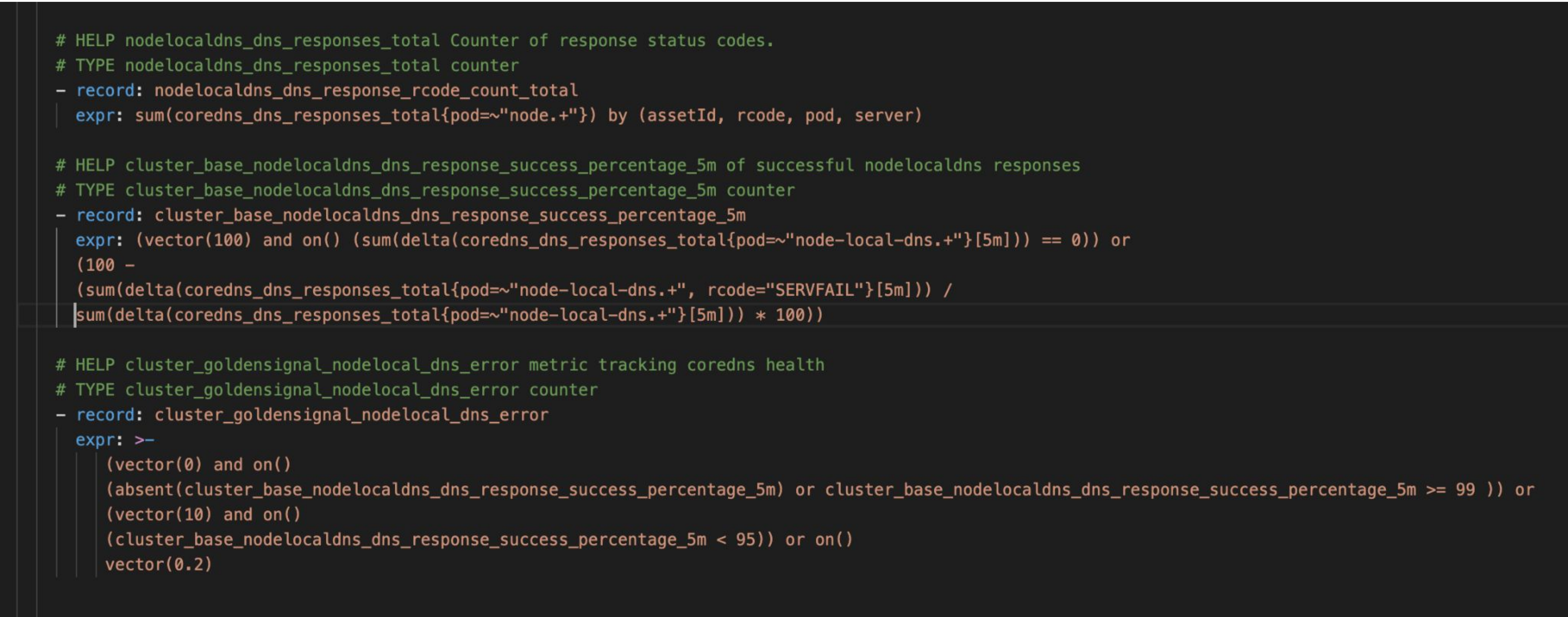


```
spec:
  groups:
    - name: cluster-overall-error.metrics.rules
      rules:
        - expr: cluster_goldensignal_cluster_autoscaler_error +     // (Autoscaling)
                cluster_goldensignal_eks_apiserver_error +          // (Control Plane)
                cluster_goldensignal_kiam_error +                   // (Authentication)
                cluster_goldensignal_externaldns_error +            // (Networking)
                cluster_goldensignal_nodelocal_dns_error +          // (Networking)
                cluster_goldensignal_alb_ingress_controller_error + // (Networking)
                cluster_goldensignal_calico_node_error +            // (Networking)
                cluster_goldensignal_kubeproxy_error +              // (Networking)
                cluster_goldensignal_oil_o11y_error +               // (Critical Addons)
                cluster_goldensignal_opa_error                      // (Critical Addons)
          record:  cluster_base_all_error_sum
        - expr: >-
            (vector(0) and on()
            (absent(cluster_base_all_error_sum) or cluster_base_all_error_sum == 0)) or
            (vector(2) and on()
            (cluster_base_all_error_sum >= 10)) or on()
            vector(1)
          record: cluster_goldensignal_overall_error
```

Cluster Golden Signals: Errors, a closer look

Error rate of a component is usually calculated based on:

- Success rate SLA over a preset time window
- An example success rate SLA calculations for ‘node-local-dns’ may look like this:

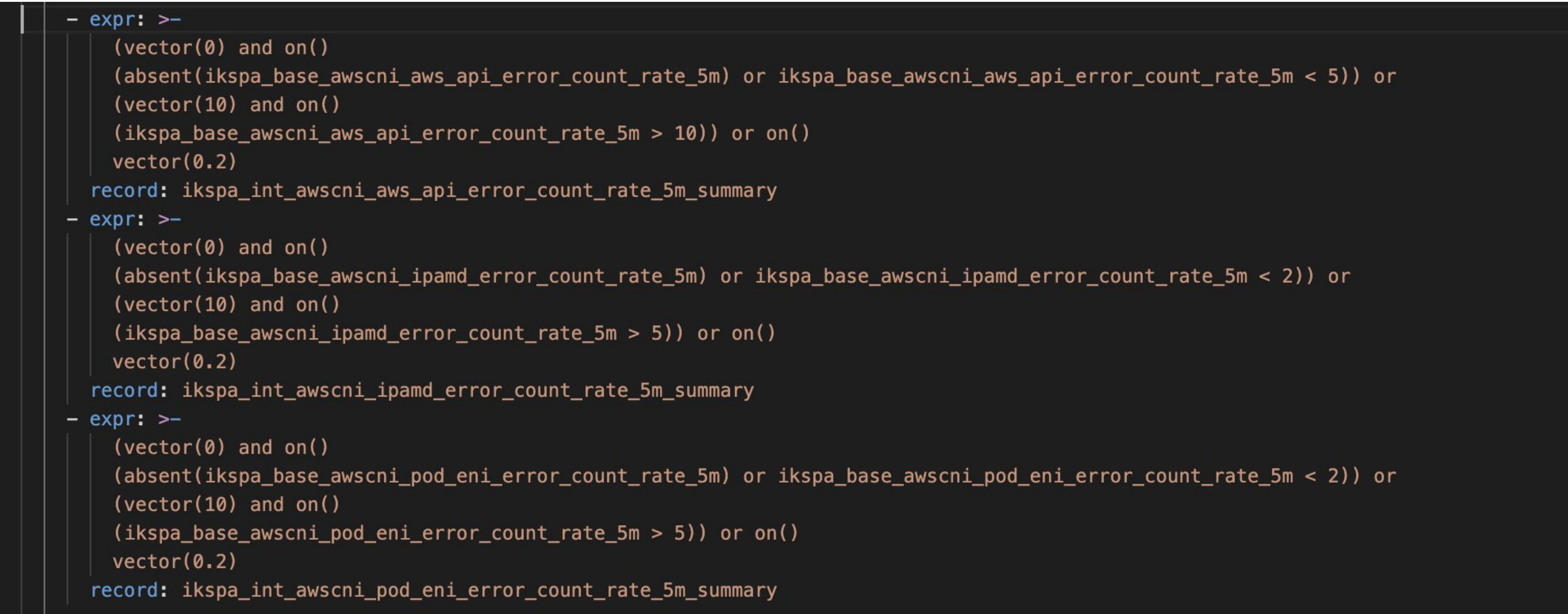


```
# HELP nodelocaldns_dns_responses_total Counter of response status codes.
# TYPE nodelocaldns_dns_responses_total counter
- record: nodelocaldns_dns_response_rcode_count_total
  expr: sum(coredns_dns_responses_total{pod=~"node.+"}) by (assetId, rcode, pod, server)

# HELP cluster_base_nodelocaldns_dns_response_success_percentage_5m of successful nodelocaldns responses
# TYPE cluster_base_nodelocaldns_dns_response_success_percentage_5m counter
- record: cluster_base_nodelocaldns_dns_response_success_percentage_5m
  expr: (vector(100) and on() (sum(delta(coredns_dns_responses_total{pod=~"node-local-dns.+"}[5m])) == 0)) or
  (100 -
  (sum(delta(coredns_dns_responses_total{pod=~"node-local-dns.+", rcode="SERVFAIL"}[5m])) /
  sum(delta(coredns_dns_responses_total{pod=~"node-local-dns.+"}[5m])) * 100))

# HELP cluster_goldensignal_nodelocal_dns_error metric tracking coredns health
# TYPE cluster_goldensignal_nodelocal_dns_error counter
- record: cluster_goldensignal_nodelocal_dns_error
  expr: >-
     (vector(0) and on()
     (absent(cluster_base_nodelocaldns_dns_response_success_percentage_5m) or cluster_base_nodelocaldns_dns_response_success_percentage_5m >= 99 )) or
     (vector(10) and on()
     (cluster_base_nodelocaldns_dns_response_success_percentage_5m < 95)) or on()
     vector(0.2)
```

Cluster Golden Signals: Errors, a closer look

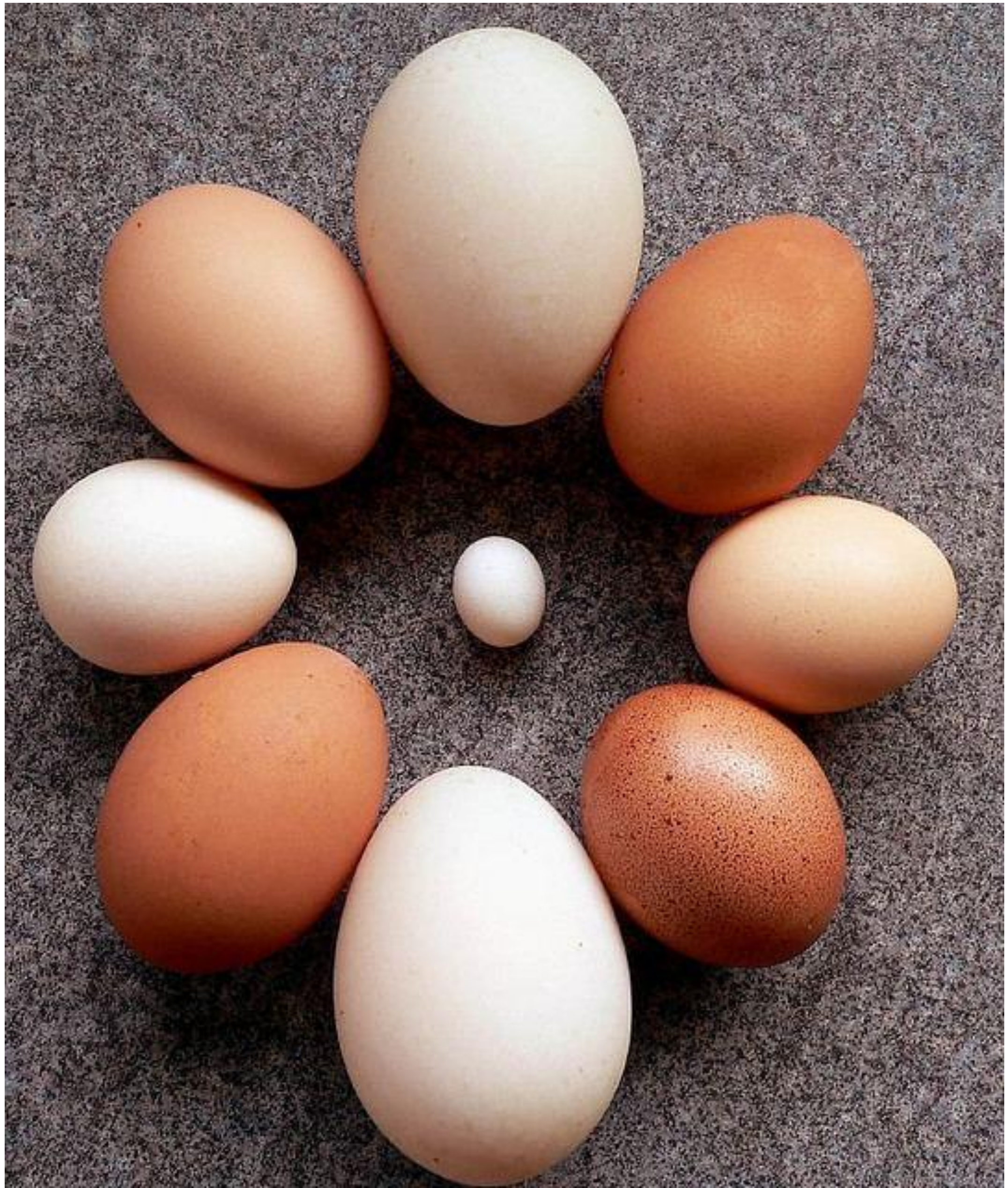
Error rate of a component is also calculated based on:

- Error count SLA over a preset time window
- An example error count SLA calculations for ‘aws-cni’ may look like this:



```
- expr: >-
    (vector(0) and on()
    (absent(ikspa_base_awscni_aws_api_error_count_rate_5m) or ikspa_base_awscni_aws_api_error_count_rate_5m < 5)) or
    (vector(10) and on()
    (ikspa_base_awscni_aws_api_error_count_rate_5m > 10)) or on()
    vector(0.2)
  record: ikspa_int_awscni_aws_api_error_count_rate_5m_summary
- expr: >-
    (vector(0) and on()
    (absent(ikspa_base_awscni_ipamd_error_count_rate_5m) or ikspa_base_awscni_ipamd_error_count_rate_5m < 2)) or
    (vector(10) and on()
    (ikspa_base_awscni_ipamd_error_count_rate_5m > 5)) or on()
    vector(0.2)
  record: ikspa_int_awscni_ipamd_error_count_rate_5m_summary
- expr: >-
    (vector(0) and on()
    (absent(ikspa_base_awscni_pod_eni_error_count_rate_5m) or ikspa_base_awscni_pod_eni_error_count_rate_5m < 2)) or
    (vector(10) and on()
    (ikspa_base_awscni_pod_eni_error_count_rate_5m > 5)) or on()
    vector(0.2)
  record: ikspa_int_awscni_pod_eni_error_count_rate_5m_summary
```

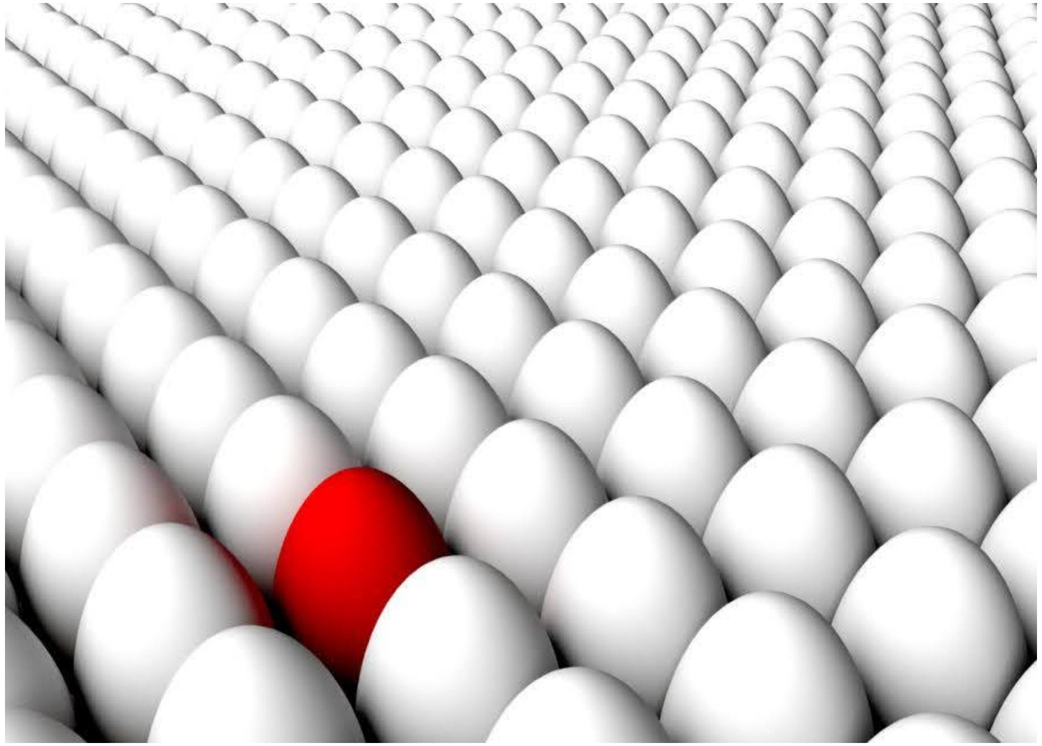
Static threshold Conundrum



KubeCon | CloudNativeCon Europe 2023

Anomaly Detection

Z-Scores For Anomaly Detection



- Z-score is a popular statistical measure to calculate outliers in a normal distribution. It takes historical data for a particular duration and calculates a z-score for new data based on that.

- General calculation is:
  z-score_metric = (current_metric_value - average_over_time) / standard_deviation_over_time

- Anomaly detection:
  Z-Scores from **1 to -1** map as healthy
  Z-Scores outside **2 to -2** map as degraded
  Z-Scores outside **3 to -3** map to critical

Z-Score: Pros & Cons

Pros

- **Well known** statistical approach.
- Provides **cluster specific anomaly detection.**
- Simple and **built-in Prometheus** rules**.**

Cons

- Z-order is a mean based approach. So detecting anomalies when there are spikes on a **downward spike is difficult**.
- Z-score implicitly assumes that the data follows a **Gaussian distribution**. Any real data may not be strictly Gaussian, and in those cases this fails.
- Z-score is very **sensitive to outliers** in the training data. Even a point anomaly with a huge spike can make detection less sensitive. As a result, we need **more data (at least a few weeks worth data)** in order to get a good signal from Z-Score.

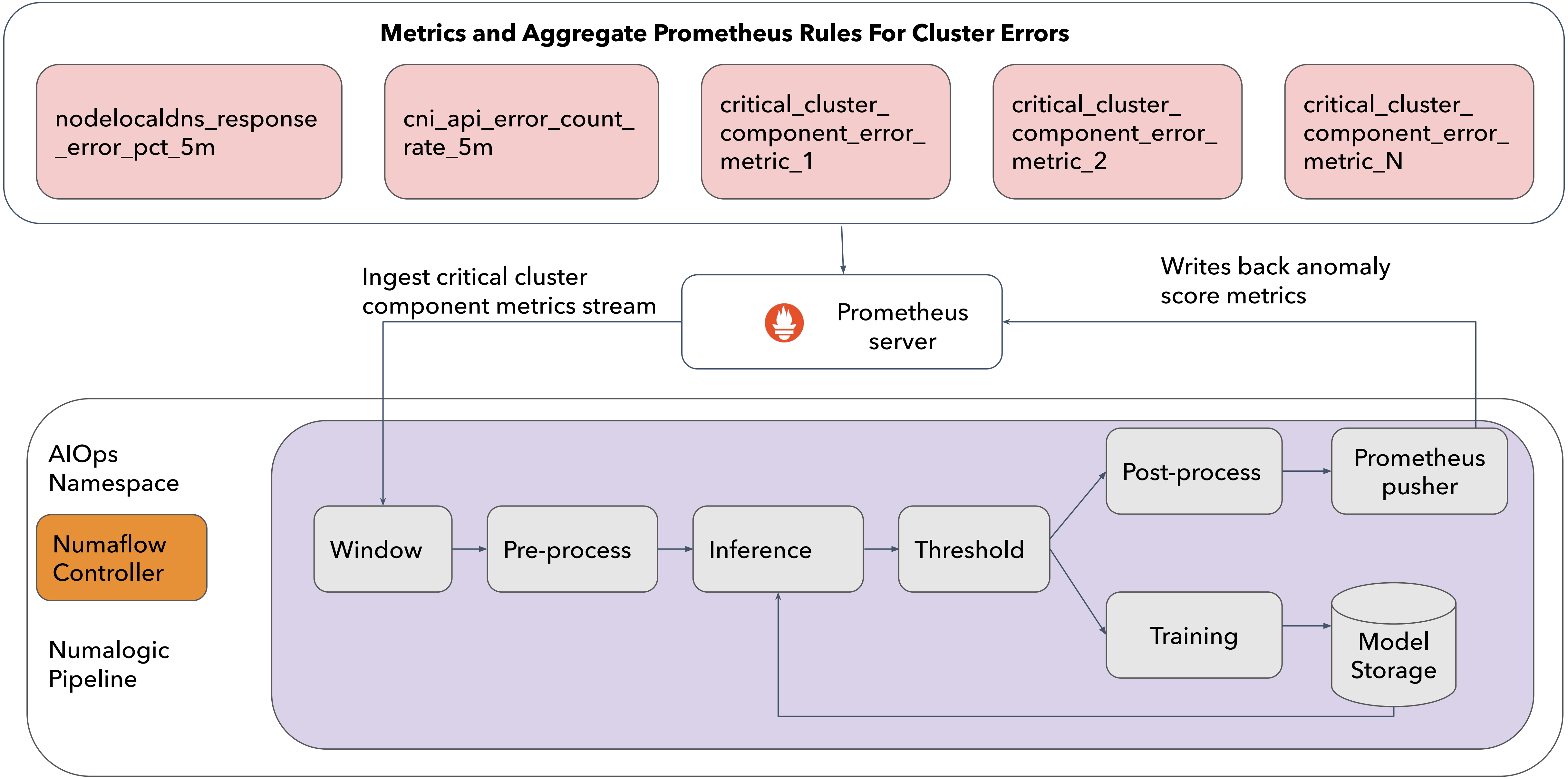
Numaproj

Numaproj - Real-Time Analytics & AIOps on Kubernetes

Numaflow
Massively parallel, real-time data and stream processing engine

Numalogic
ML models and libraries for real-time operational AIOps

AIOps Pipeline



Metrics and Aggregate Prometheus Rules For Cluster Errors

nodelocaldns_response_error_pct_5m

cni_api_error_count_rate_5m

critical_cluster_component_error_metric_1

critical_cluster_component_error_metric_2

critical_cluster_component_error_metric_N

Ingest critical cluster component metrics stream

Prometheus server

Writes back anomaly score metrics

AIOps Namespace

Numaflow Controller

Numalogic Pipeline

Window

Pre-process

Inference

Threshold

Post-process

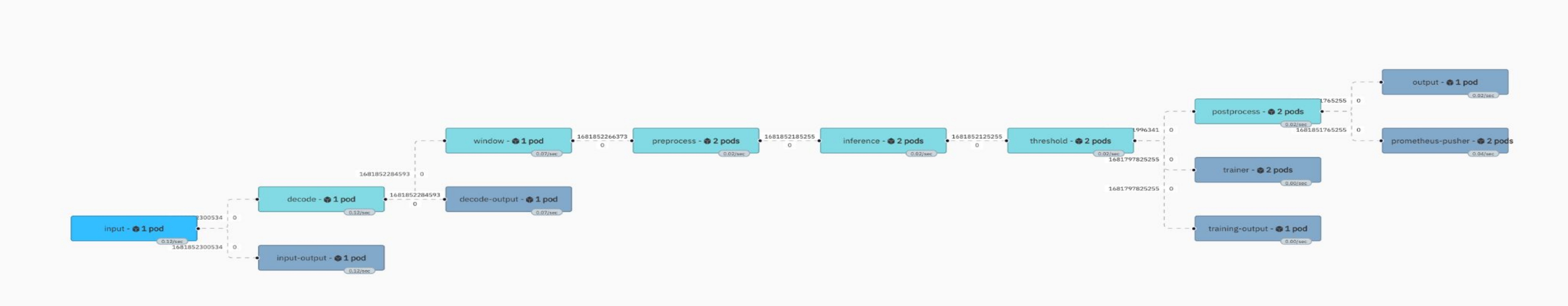
Prometheus pusher

Training

Model Storage

Numaflow

- https://github.com/numaproj/numaflow
- Numaflow **Pipeline, Vertex** and **InterstepBufferServices** are implemented as a Kubernetes custom resource and support the different aspects of the pipeline
- Simple deployment. Setup ready in minutes!
- Effective for Cluster anomaly detection
- Numaflow UI available (see below) with a simple port-fwd. You can look through the pipeline and debug using logs and timestamps (if needed)



Numalogic: FAQ For Kubernetes Engineers

Q: Do I need to be an ML engineer to use the project?
A: No. But, it's definitely worth checking out the project at https://github.com/numaproj/numalogic

Q: Tell me a bit about the ML model. How does it detect anomalies?
A: The ML model is an "auto-encoder" model. It tries to learn the normal behavior of the data. An anomalous data point will produce higher reconstruction error, since the model cannot reconstruct those outlier points properly.

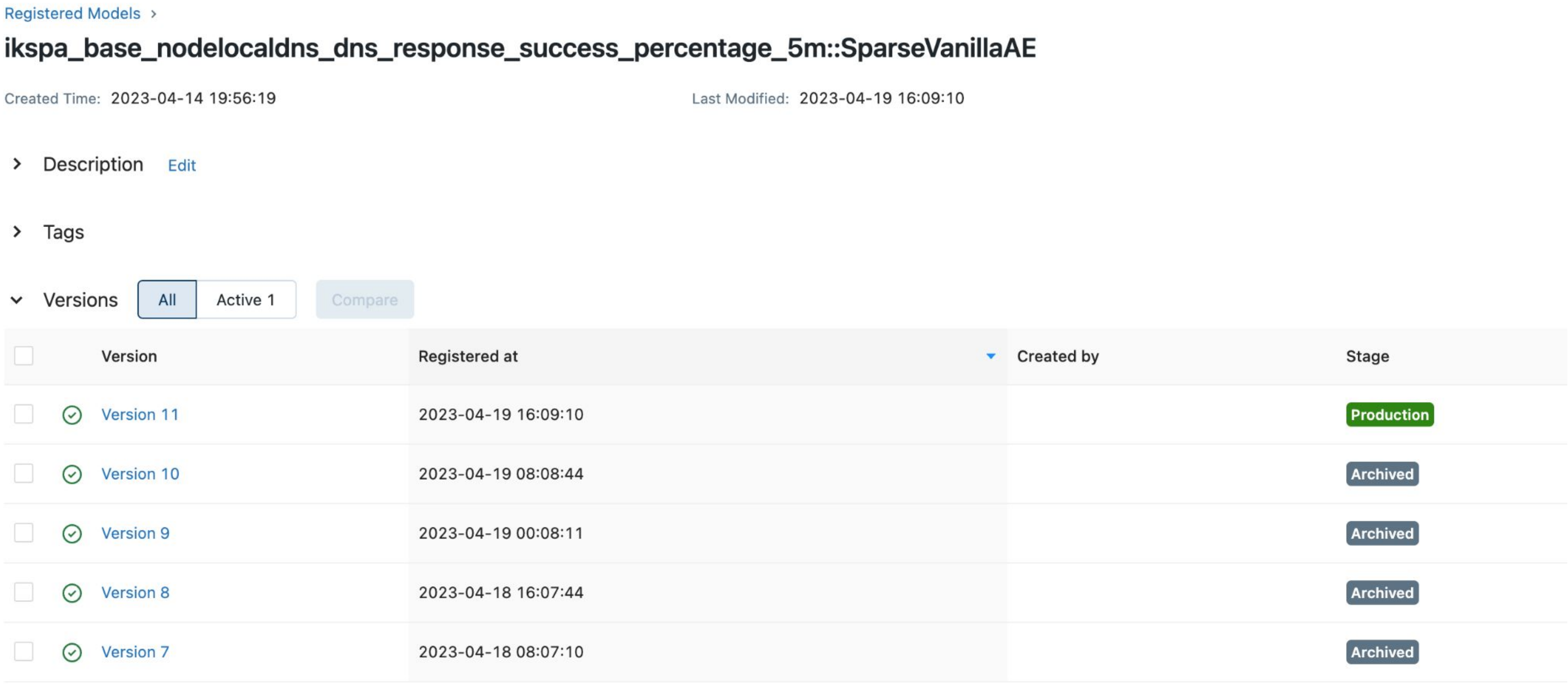
Q. What's the purpose of retaining models? How many models are retained?
A: Automating an ML training process is a hard problem. There can be times when the models could be trained on a bad data, or maybe very less data. Previous models serve as a backup. By default, we retain upto 5 models. This can be configured.

Q. What is the model training frequency?
A: The default that we use for Intuit AIOps systems is 8 hours. This means that after this interval, the models will be retrained on the fresher data. Data drift can happen from time to time, so retraining models will address that.

Numalogic: FAQ For Kubernetes Engineers

Q: Is there a UI to observe and configure the ML models?

A: Yes. You can access the ML models UI with a simple port-forward. You can look at the different ML models, retrigger training, etc using the UI. See below.



Registered Models ›

ikspa_base_nodelocaldns_dns_response_success_percentage_5m::SparseVanillaAE

Created Time: 2023-04-14 19:56:19

Last Modified: 2023-04-19 16:09:10

› Description Edit

› Tags

⌄ Versions All Active 1 Compare

□		Version	Registered at	Created by	Stage
□	✓	Version 11	2023-04-19 16:09:10		Production
□	✓	Version 10	2023-04-19 08:08:44		Archived
□	✓	Version 9	2023-04-19 00:08:11		Archived
□	✓	Version 8	2023-04-18 16:07:44		Archived
□	✓	Version 7	2023-04-18 08:07:10		Archived

Demo & Takeaways

Demo

Takeaways

- Implementing Cluster Golden Signals will help reduce operational burden for your platform engineers.

- Anomaly Detection using NumaProj is promising!

- Check out:
  - github.com/numaproj/numaflow
  - github.com/numaproj/numalogic
  - github.com/numaproj/numalogic-prometheus

KubeCon | CloudNativeCon

Europe 2023

Thank You!