
KubeCon
CloudNativeCon
North America 2023

SIG Scalability: Intro + DeepDive

Wojciech Tyczynski, Google (SIG Scalability TL)
Marcel Zieba, Isovalent (Sig Scalability Chair)

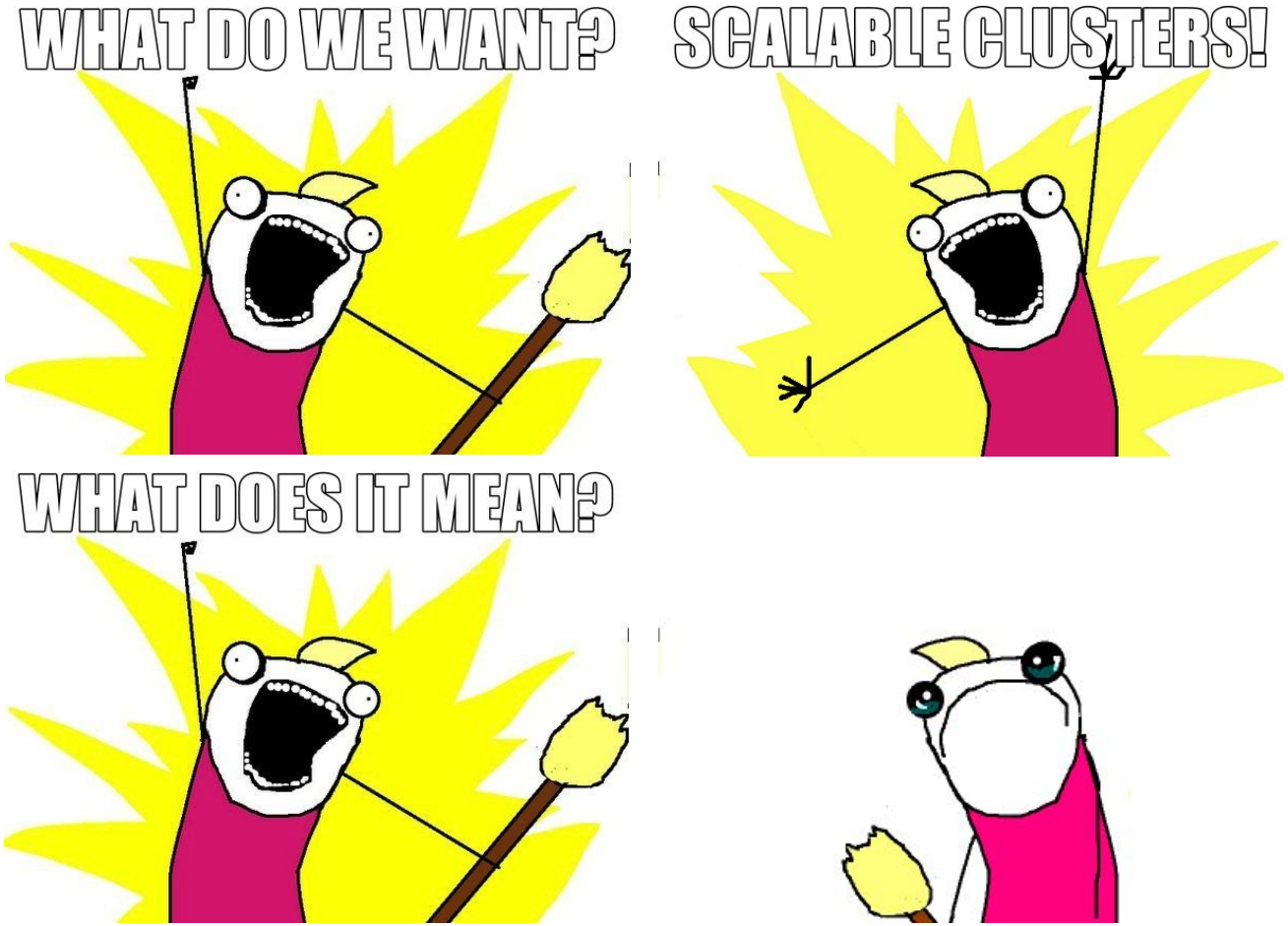
What do we do?

- Define & Drive - scalability definition & goals
- Coordinate & Contribute - performance improvements
- Monitor & Measure - performance of the system
- Preserve & Protect - from scalability regressions
- Consult & Coach - community about scalability

Not to confuse with SIG Autoscaling!

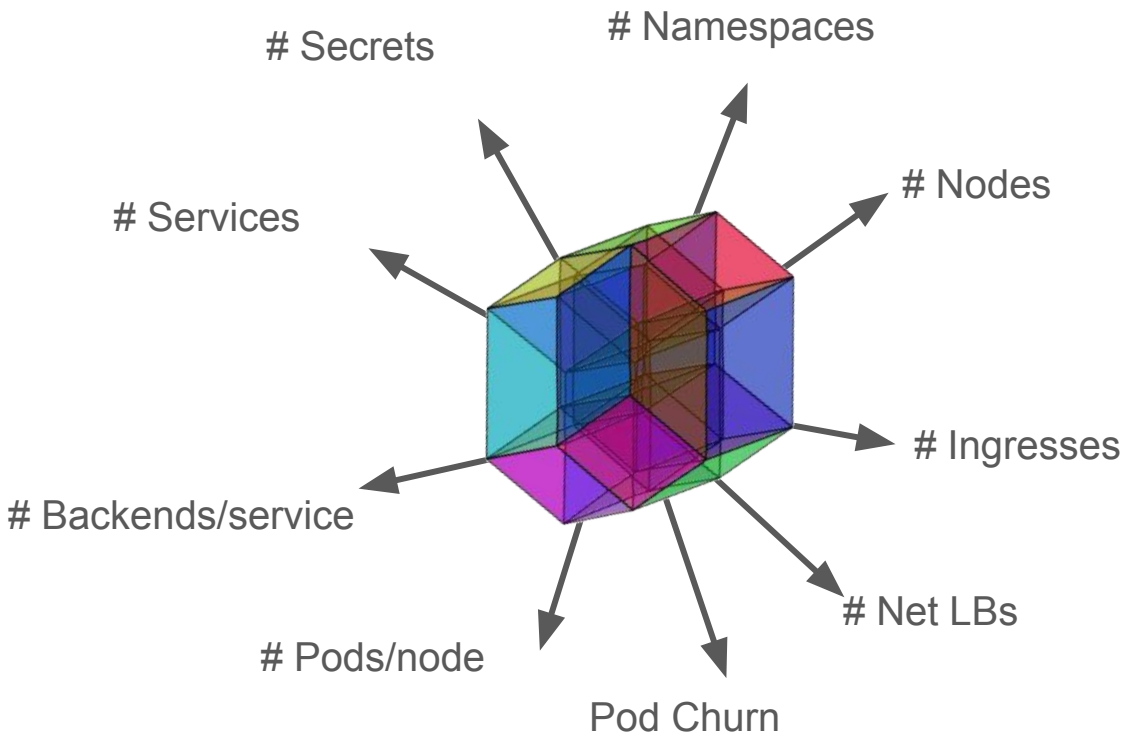
[Marcel]

What is Kubernetes Scalability?



What is Kubernetes Scalability?

~~Scalability = # nodes~~



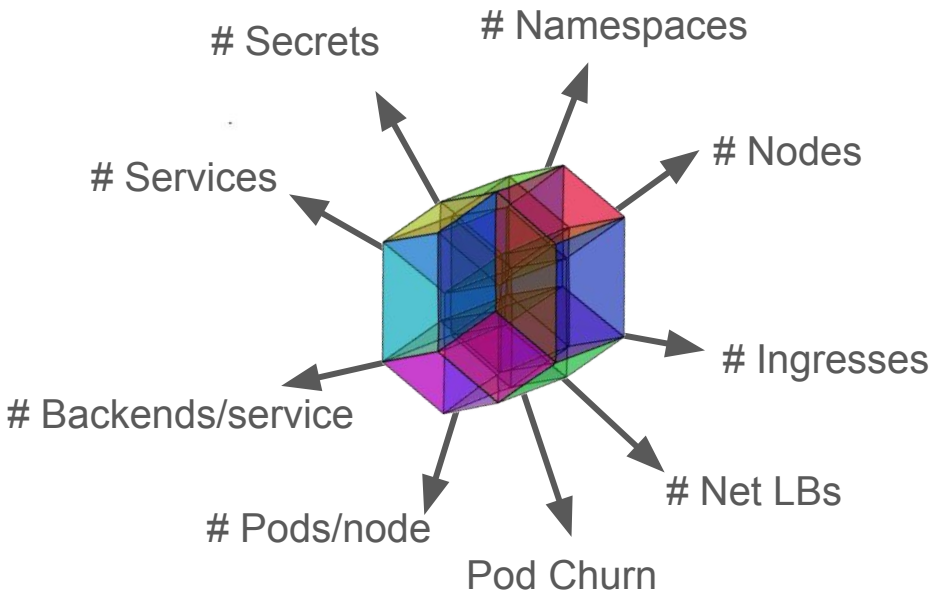
Scalability Envelope

Source of hypercube image:
http://www.gregegan.net/APPLETS/29/29.html

What is Kubernetes Scalability?

?

Scalability Envelope - a *safe zone*, within which your cluster is *happy*.



ALL (scalability)
SLOs are satisfied

Source of hypercube image:
http://www.gregegan.net/APPLETS/29/29.html

Kubernetes Scalability SLIs/SLOs

SLI - Service Level Indicator

SLO - Service Level Objective

1. API Call Latency
2. Pod Startup Latency
3. In-Cluster Network Programming Latency
4. DNS Programming Latency
5. In-Cluster Network Latency
6. DNS Latency

More at github.com/kubernetes/community/tree/master/sig-scalability/slos

SLIs/SLOs - case study

2015: (blog post)

SLO: “99% of all our API calls return in less than 1 second”

2023: (definition)

SLI: Latency of processing mutating API calls for single objects for every (resource, verb) pair, measured as 99th percentile over last 5 minutes

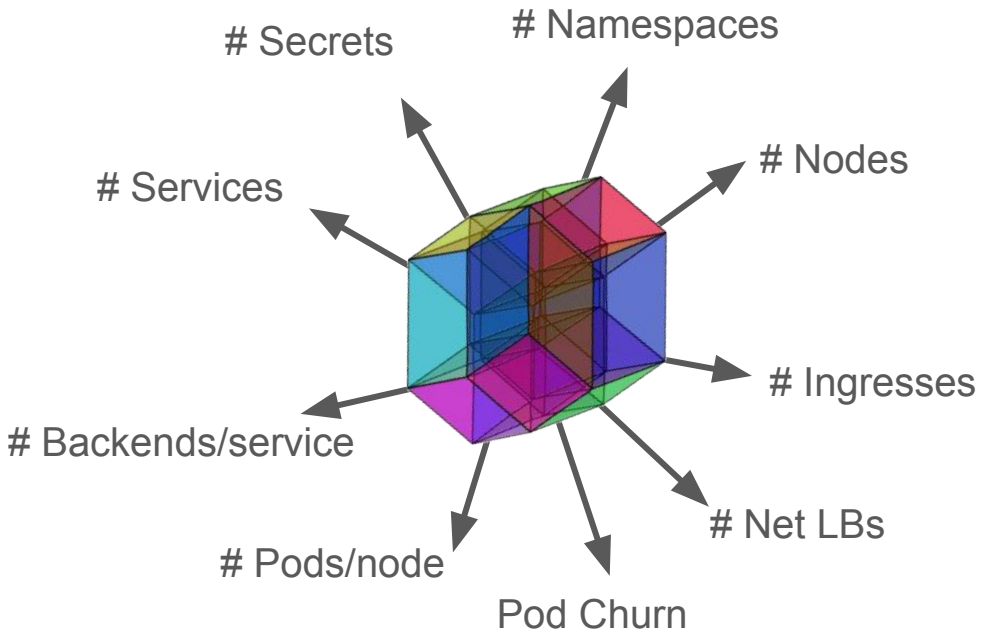
SLO: In default Kubernetes installation, for every (resource, verb) pair, excluding virtual and aggregated resources, 99th percentile per cluster-day <= 1s

Scalability limits

Precise definition of **scalability envelope** is ~impossible.

Fortunately, we have reasonable approximation:

- #Nodes <= 5,000
- #Services <= 10,000
- …



More at [sig-scalability/configs-and-limits/thresholds.md]

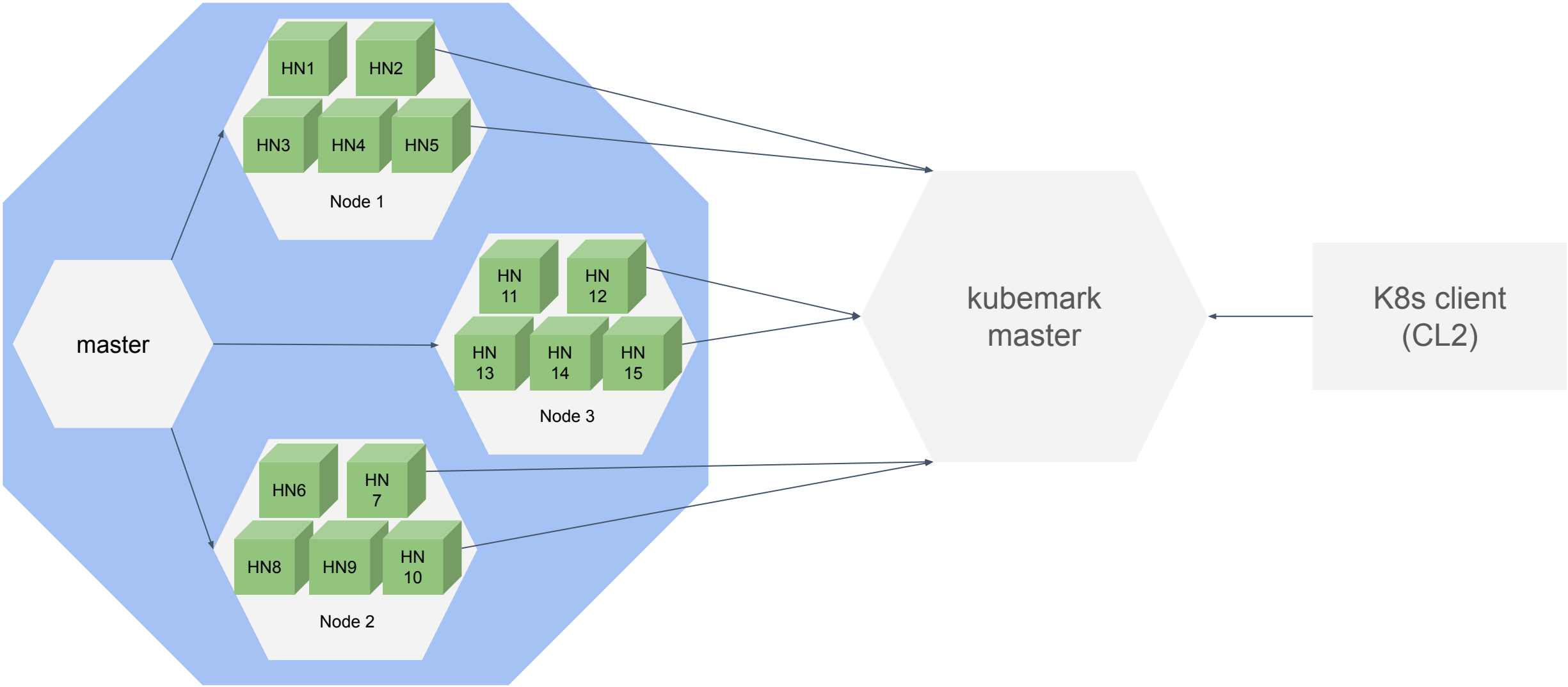
Source of hypercube image:
http://www.gregegan.net/APPLETS/29/29.html

Scalability Testing Infrastructure

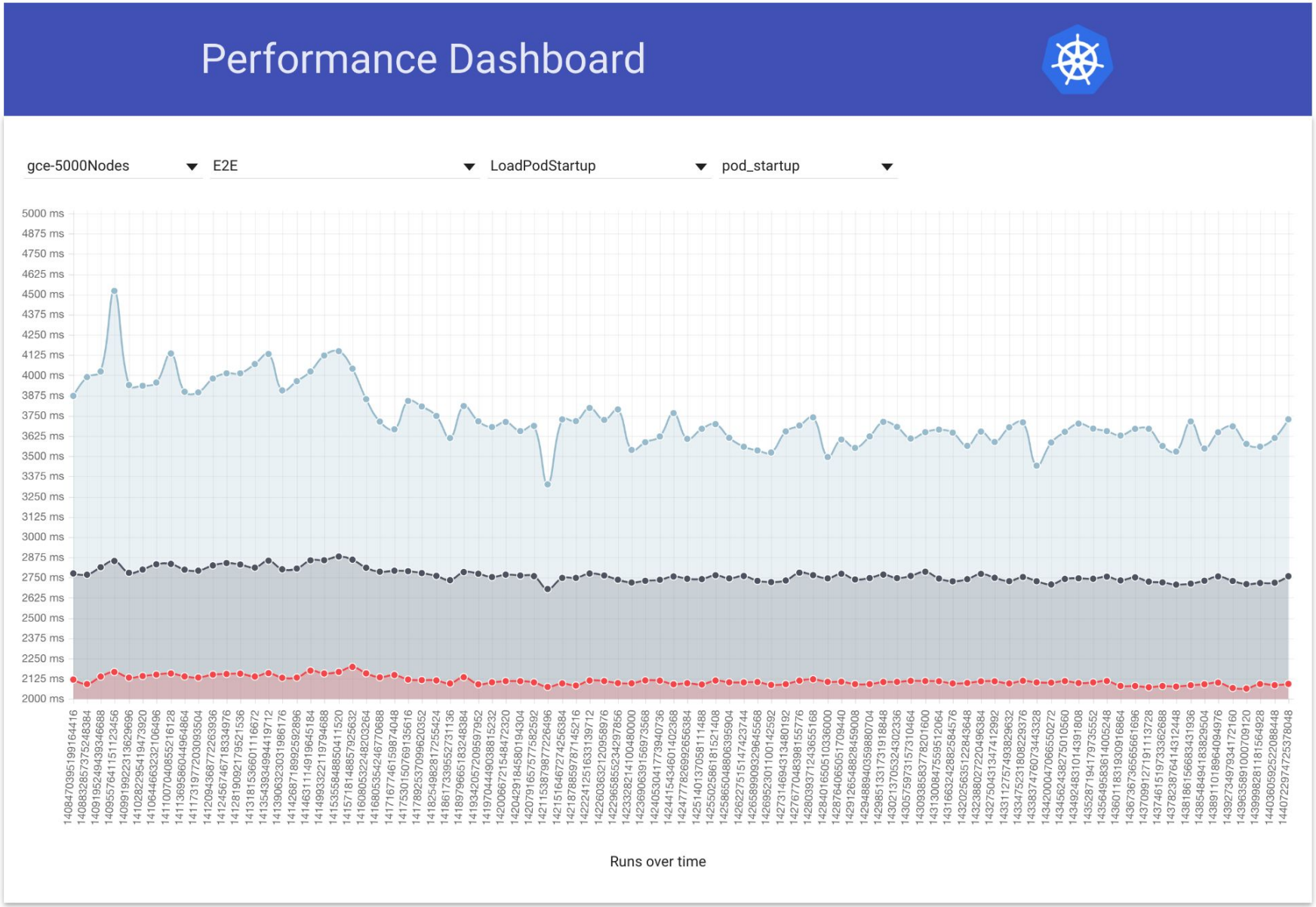
Test framework - ClusterLoader2

- **A “bring your own yaml” test framework**
  - User (semi) declaratively describes desired state of the cluster
  - ClusterLoader brings the cluster to state
  - At the same time verifying Scalability SLOs
- **Designed for easy extensibility**
  - Provides extra observability
  - And a bunch of extra features
  - See kubernetes/perf-tests/clusterloader2 for more details

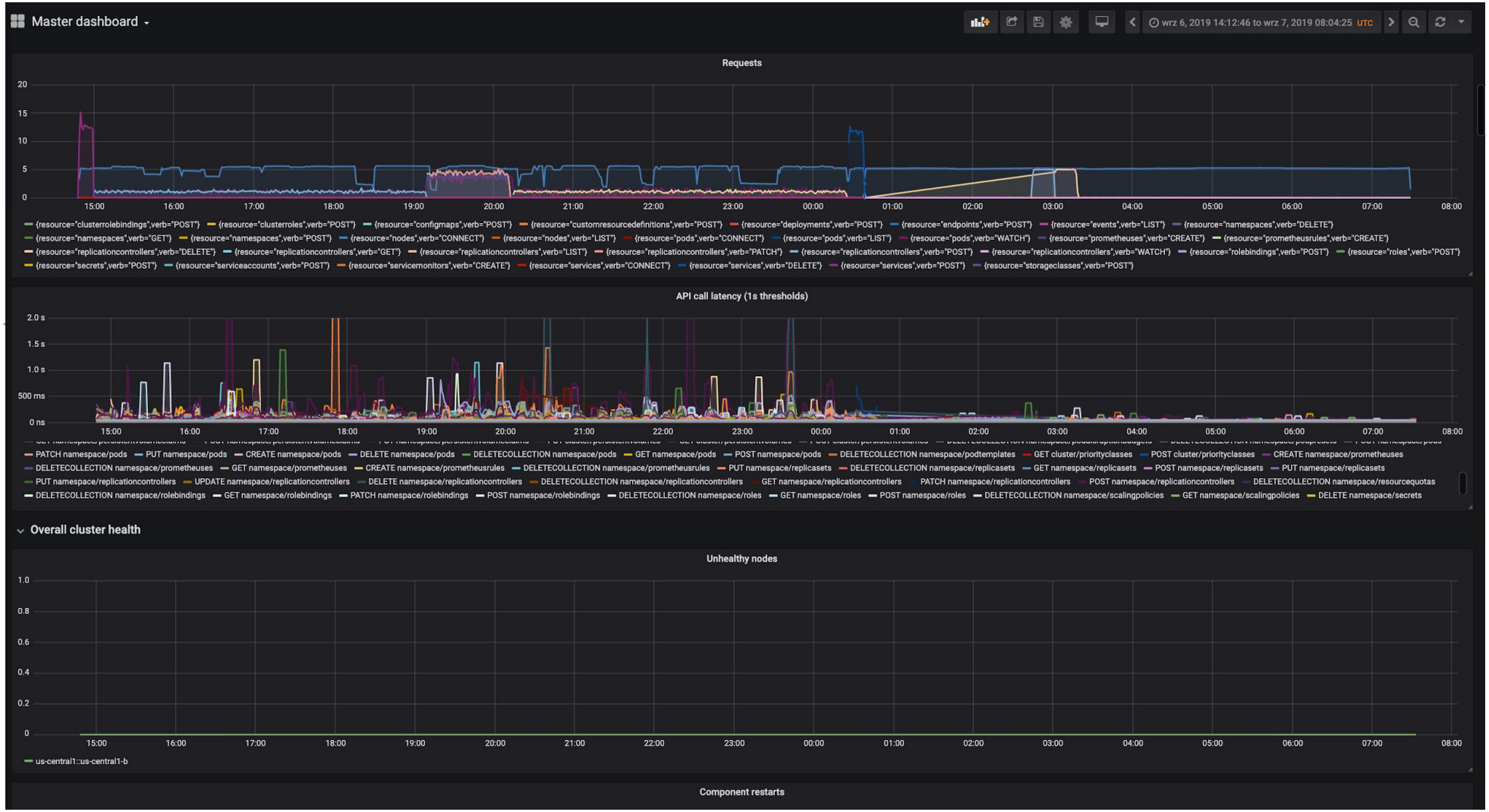
Cluster simulation - Kubemark



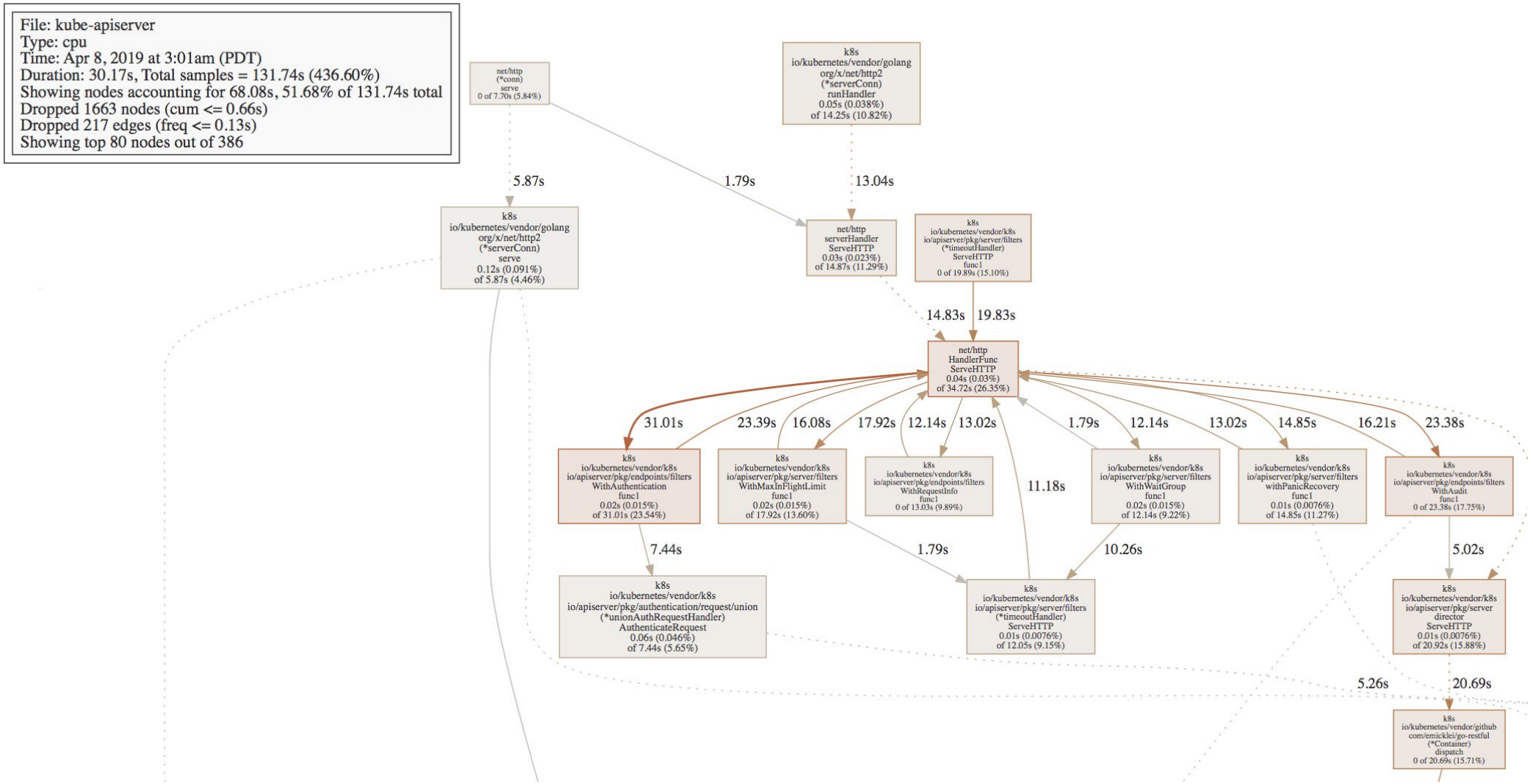
Observability & debuggability - Perfdash



Observability & debuggability - Grafana



Observability & debuggability - Profiling



Protecting Kubernetes Scalability

Scalability tests

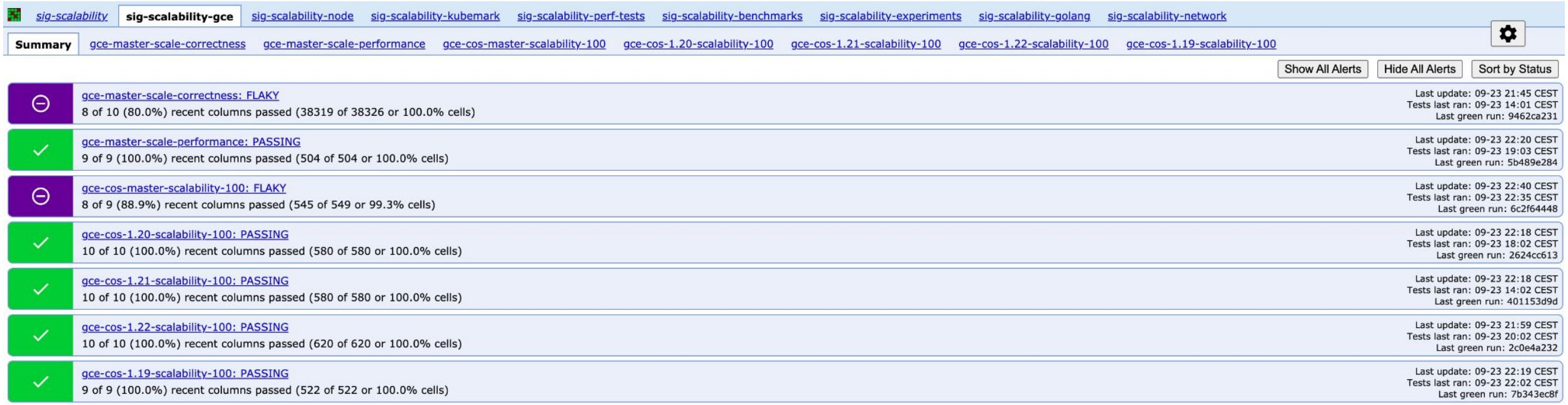
Periodic tests:

- Release blocking tests:
  - Performance 100 nodes
  - Performance 5000 nodes
  - Correctness 5000 nodes
- Non-release blocking
  - Kubemark
  - Storage
  - Benchmarks
  - …

Optional Presubmit tests:

- Performance 100 nodes
  - kubernetes
  - Perf-tests
- Performance 5000 nodes

Scalability Tests Infrastructure



Scalability regressions

- Scalability is *sensitive*
- We've seen regressions come from pretty much everywhere:
  - Golang
  - Operating System
  - Controllers
  - API machinery
  - Scheduler
  - Etcd
  - Kubelet
  - …
- We often debug/fix them ourselves, or triage to relevant SIGs

Driving Scalability Improvements

Scalability improvements

Two categories of work:

- Improving reliability at scale
  - Scalability can be thought of as reliability at scale
- Pushing the limits

[Most joined with SIG API-machinery]

Scalability improvements

Improving reliability [at scale]:

- API Priority & Fairness - #1040
  - GA-ing in 1.29
- Improved upgrades experience
  - Graceful shutdown - e.g. #114925
- API streaming lists - #3157
  - Second Alpha in 1.29

Scalability improvements

Pushing the limits

- Improved CRD scalability:
  - Binary encoding for CRDs - #4222
    - Pre-alpha in 1.29
  - Efficient event serialization - #120300

- Localized kube-apiserver improvements
  - e.g. faster compression - #112296
  - e.g. page size progressing - #108569

- Improvements towards higher throughput

Justification is critical for complexity vs ROI tradeoff

How to get involved?

Where to find us?

- Home page: README
- Public Meetings: Thursdays 17.30 UTC (bi-weekly)
- Slack channel: #sig-scalability
- Mailing List: #kubernetes-sig-scale

How to get involved?

- kubernetes/perf-tests help-wanted
- kubernetes/kubernetes help-wanted

Please scan the QR Code above
to leave feedback on this session