



KubeCon



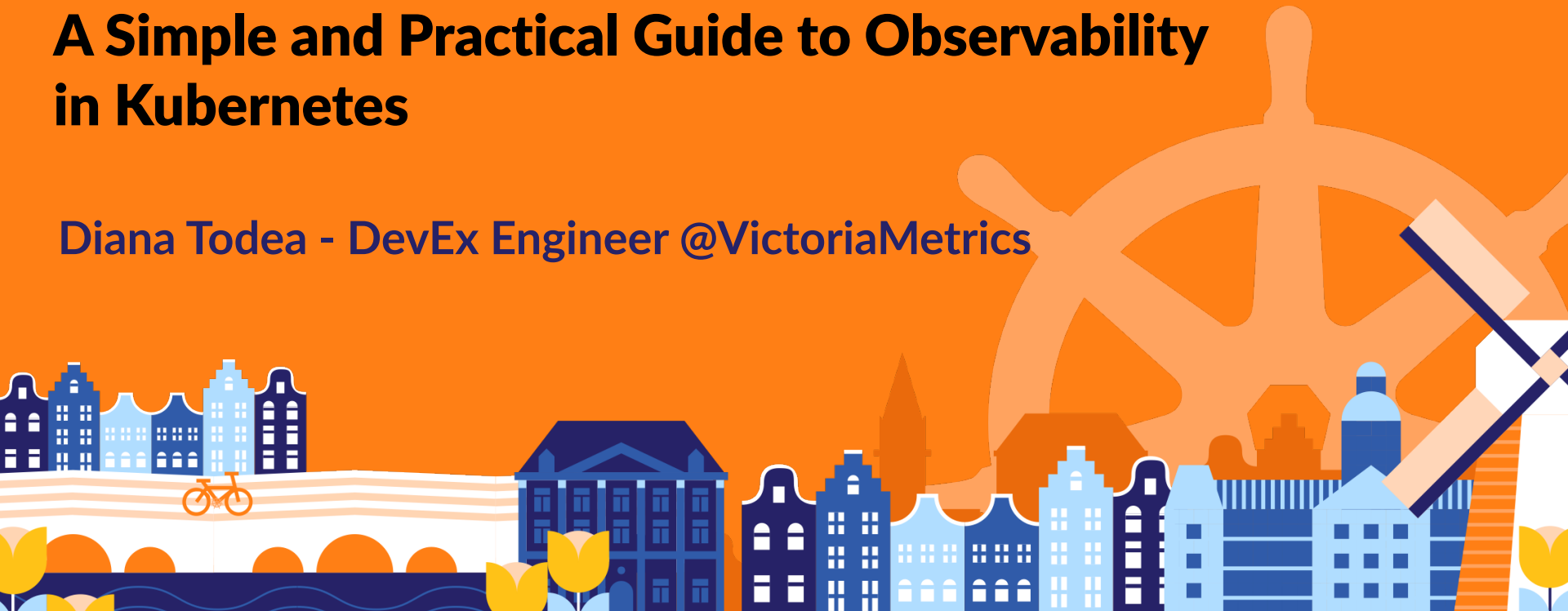
CloudNativeCon

Europe 2026

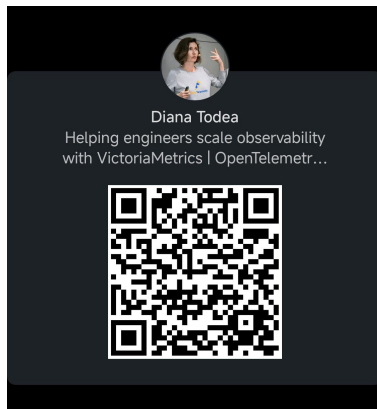
#KubeCon #CloudNativeCon

A Simple and Practical Guide to Observability in Kubernetes

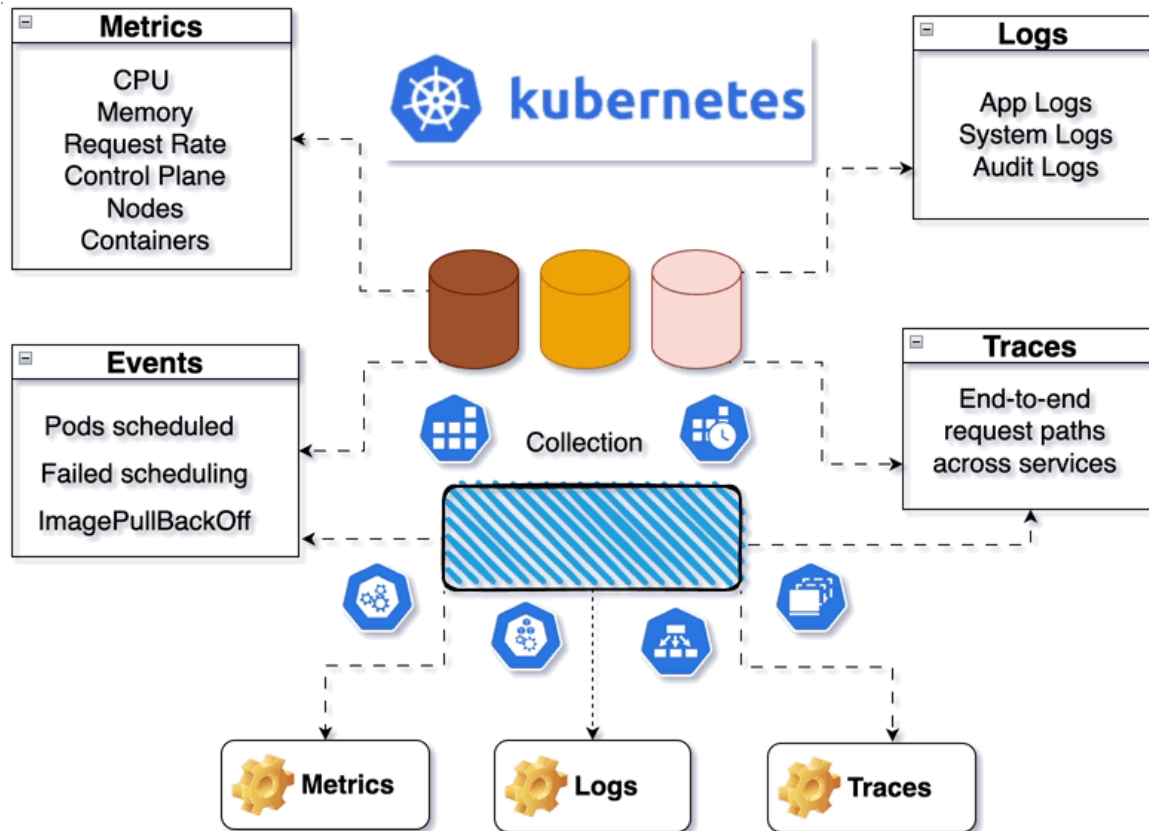
Diana Todea - DevEx Engineer @VictoriaMetrics



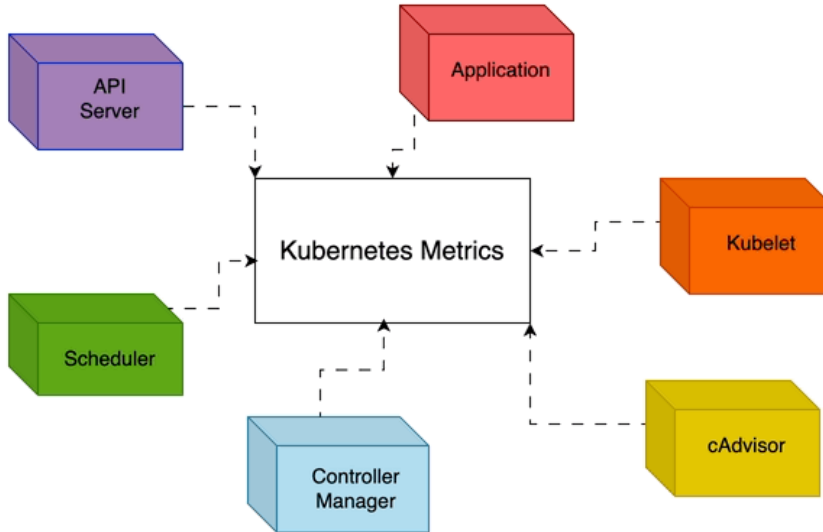
OpenTelemetry member and contributor
Co-lead CNCF Merge-Forward Neurodiversity
Cloud Native Days Romania organizer



Github: @DidiViking



Kubernetes metrics: who produces them?



Kubernetes metrics stability

- **Alpha** metrics have no stability guarantees; as such they can be modified or deleted at any time. At this time, all Kubernetes metrics implicitly fall into this category.
- **Stable** metrics can be guaranteed to not change, except that the metric may become marked deprecated for a future Kubernetes version.

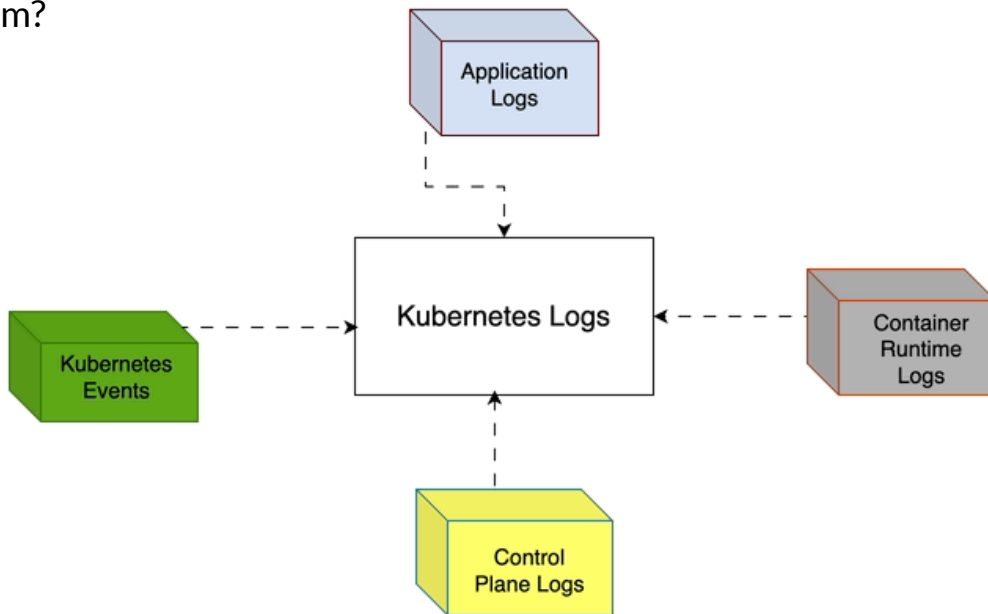
What metrics actually matter?

- Control Plane: e.g. API server request rate, API server latency, API errors
- Nodes and workloads: CPU usage, memory usage, resource requests vs limits, pod restarts
- Scheduling: scheduling latency, pending pods

The common metrics problem

- Inconsistent naming and labels
- High cardinality
- Duplicated metrics
- Dashboards with no purpose

Where do Kubernetes logs come from?



The common logs problem

- Logs are not standardized
- Control plane → structured JSON
- Application logs → anything
- Events → structured objects
- Audit logs → strict JSON schema
- Kubernetes logs are generally structured.
- Your application logs are your responsibility.

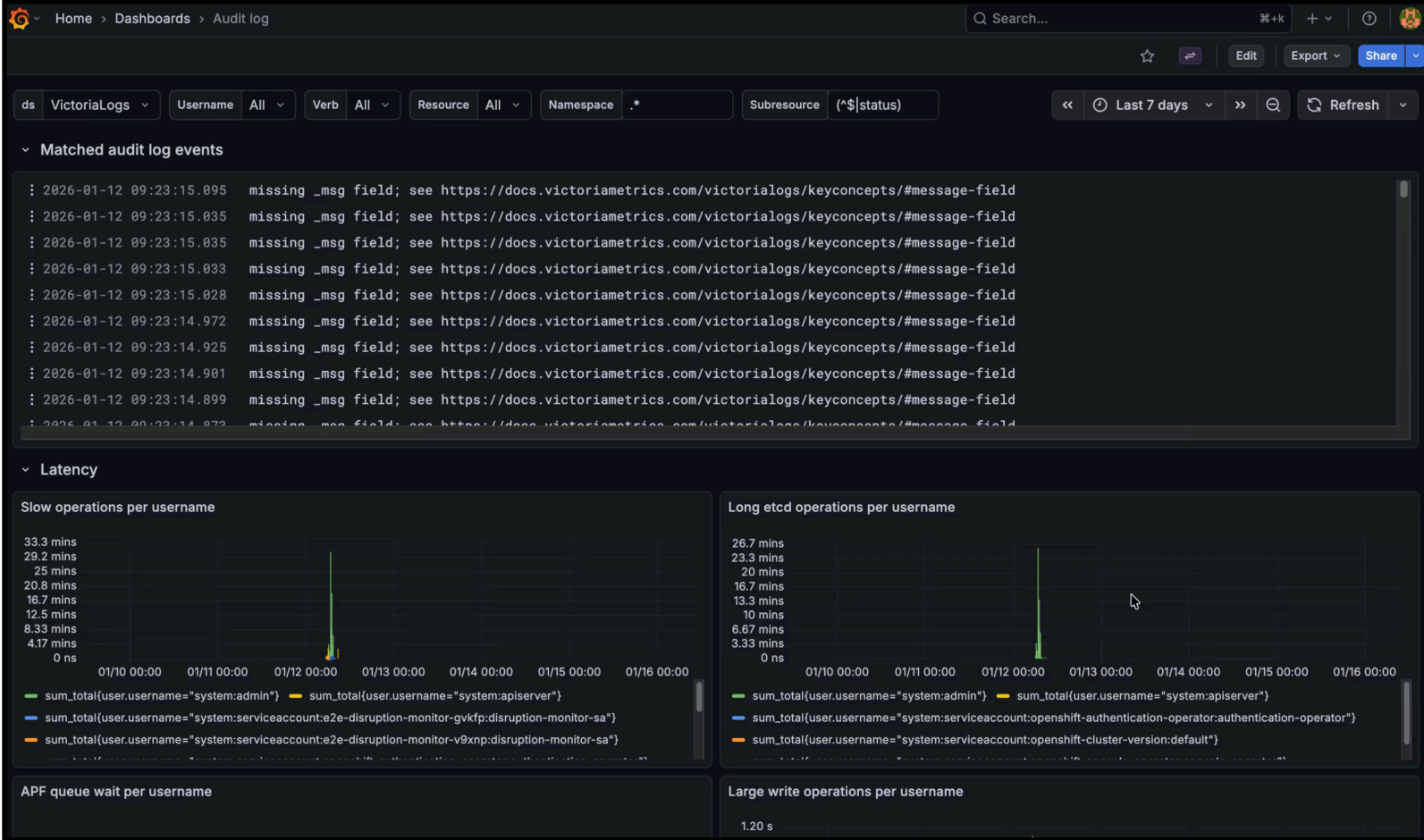
Kubernetes media

- Events are extremely useful: <https://github.com/resmoio/kubernetes-event-exporter>
- Audit logs tell you WHO did WHAT
- Audit logs contain sensitive info - IPs, role names etc.
- Example: kubernetes-event-exporter

Demo

- Audit Logs Stats app parses Kubernetes audit log files and sends them to VictoriaLogs. A Grafana dashboard is used to render statistics derived from these logs, helping to identify noisy applications or requests that take too much time.
- Source code: <https://github.com/vrutkovs/audit-log-stats>





Advice for developers:

- Use structured logs
- Instrumentation from day zero
- Focus on alerts! Which alerts would you like to see? This will dictate with metrics do you want to have. Dashboards are tied to alerts.

Tracing in Kubernetes

- <https://kubernetes.io/docs/concepts/cluster-administration/system-traces/>
- Benefits of tracing: vendor-neutral standard, common tools to work with
- Currently tracing is grouped by stability, but we also want them grouped by criticality
- Definitely something that we want SIG-Instrumentation to focus on!

What SIG-Instrumentation is working on

- Better metric documentation
- Group metrics by severity
- Improve dashboards & alerts
- Improve structured logging
- Observability guides
- Team consulting on best practices

Contributing to SIG-Instrumentation

- Further contribution to SIG-instrumentation is needed for end users:
- <https://github.com/kubernetes/website/issues/53168>
- Adding the contributions to SIG-instrumentation
- Once SIG-instrumentation finalizes work
- Create tutorials, documentation for end-users

Improve Kubernetes Instrumentation DevEx

- One idea would be to classify the metrics for particular use cases ranging from beginner, intermediate to advanced.
- Improve standard recording rules, alerts and dashboards at <https://github.com/kubernetes-monitoring/kubernetes-mixin>
- Create guides on how to interpret dashboards

Kubernetes Metrics Reference

Details of the metric data that Kubernetes components export.

Metrics (v1.35)

This page details the metrics that different Kubernetes components export. You can query the metrics endpoint for these components using an HTTP scrape, and fetch the current metrics data in Prometheus format.

List of Stable Kubernetes Metrics

Stable metrics observe strict API contracts and no labels can be added or removed from stable metrics during their lifetime.

`apiserver_admission_controller_admission_duration_seconds`

Admission controller latency histogram in seconds, identified by name and broken out for each operation and API resource and type (validate or admit).

- **Stability Level:** STABLE
- **Type:** Histogram
- **Labels:** `name` `operation` `rejected` `type`

`apiserver_admission_step_admission_duration_seconds`

Admission sub-step latency histogram in seconds, broken out for each operation and API resource and step type (validate or admit).

- **Stability Level:** STABLE
- **Type:** Histogram

Resources:

- Kubernetes Slack: #sig-instrumentation
- <https://github.com/vrutkovs/audit-log-stats>
- <https://github.com/kubernetes-monitoring/kubernetes-mixin>
- <https://github.com/resmoio/kubernetes-event-exporter>
- <https://github.com/kubernetes/community/sig-instrumentation>
- <https://kubernetes.io/docs/reference/instrumentation/metrics/>
- <https://kubernetes.io/docs/concepts/cluster-administration/system-logs/#structured-logging>



KubeCon



CloudNativeCon

Europe 2026

Thank you!

