KubeCon | CloudNativeCon

Europe 2023

KubeCon | CloudNativeCon Europe 2023

SIG Instrumentation Introduction and Deep Dive

Damien Grisonnet, Red Hat

About me

Damien Grisonnet

- Senior Software Engineer @ Red Hat
- Kubernetes SIG Instrumentation Tech Lead
- Maintainer of kube-state-metrics, metrics-server, and prometheus-adapter
- https://github.com/dgrisonnet
- https://linkedin.com/in/damien-grisonnet

Agenda

- What is SIG Instrumentation?
- SIG Subprojects
- Metrics
- Logs
- Traces
- How to contribute
- Where to find us

What do we do?

- **Charter**: To cover best practices for cluster observability across all Kubernetes components and develop relevant components.
- **Subprojects**:
  - kube-state-metrics
  - klog
  - metrics-server
  - and more!
- Metrics
- Logs and Events
- Traces

How do we do it?

- Triage and fix relevant instrumentation issues
  - All open SIG Instrumentation-labelled issues and pull requests
- Review all code changes for metrics
- Develop new features and enhancements
  - Kubernetes Enhancement Proposals (KEPs) for SIG Instrumentation
- Maintain and support subprojects
- Mentor new contributors

Subprojects

Subprojects

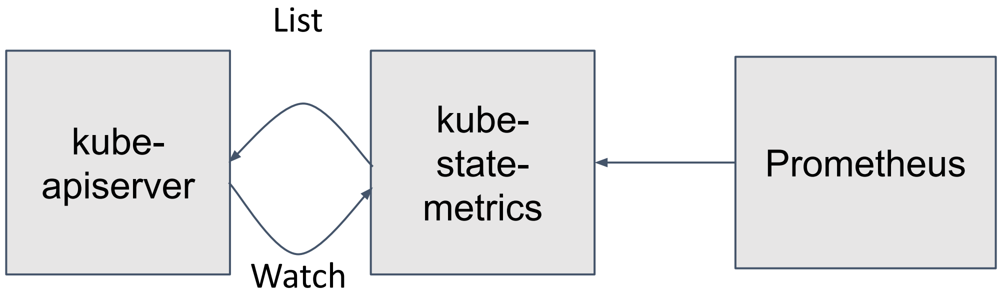
- kube-state-metrics
- metrics-server
- prometheus-adapter
- usage-metrics-collector

kube-state-metrics

- Generate Prometheus style metrics from Kubernetes API objects
- Pods, Deployments, StatefulSets, etc.

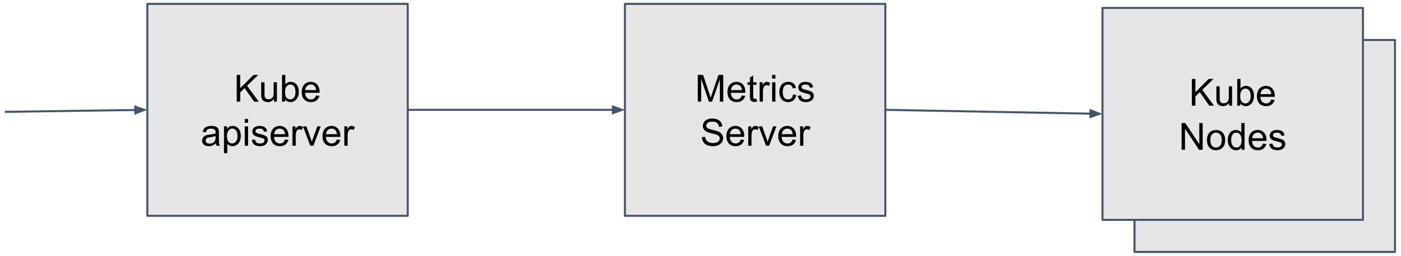
Example:

```
kube_deployment_spec_replicas
kube_deployment_status_replicas_updated
```



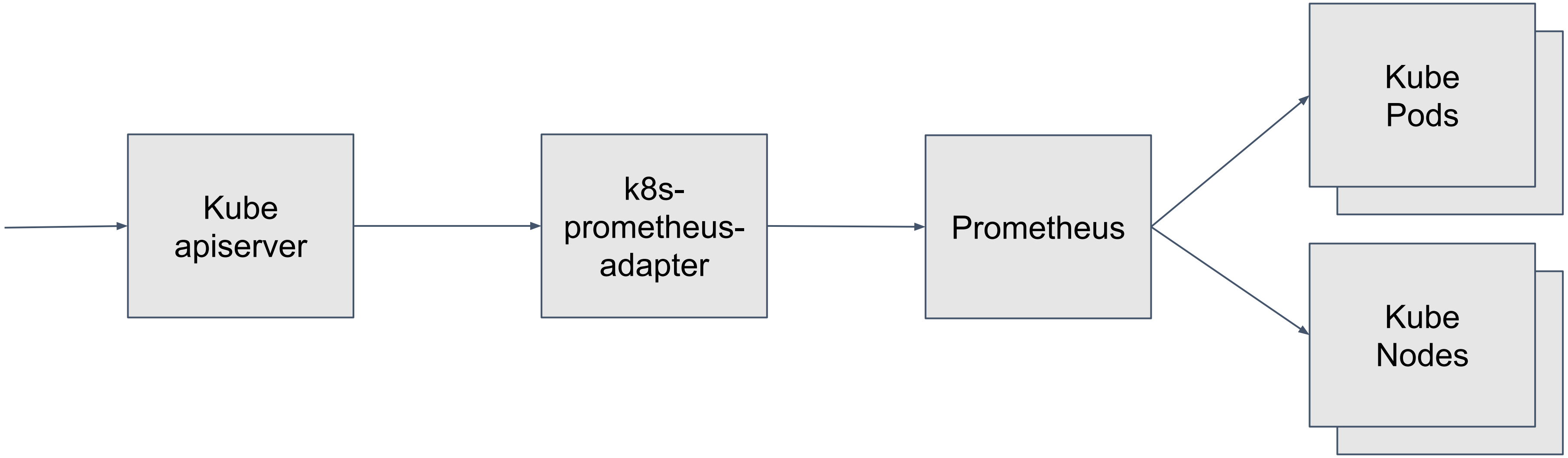
metrics-server

- Implementation of the resource metrics API
  - Source of `$ kubectl top`
  - Source of metrics for resource based autoscaling
- Repo: https://github.com/kubernetes-sigs/metrics-server



prometheus-adapter

- Implementation of the resource/custom/external metrics API
  - Use custom metrics for autoscaling
- Repo: https://github.com/kubernetes-sigs/prometheus-adapter



usage-metrics-collector

- New subproject given to the SIG by Apple in January 2023
- Prometheus metrics collector optimized for collecting kubernetes resource usage and capacity metrics.
    - High utilization metrics resolution (1s by default)
    - Performs aggregation of metrics at collection time
    - Does not require any promQL knowledge
- Repo: https://github.com/kubernetes-sigs/usage-metrics-collector/

usage-metrics-collector

Get p95 utilization (cpu and memory) using 1 second sampling intervals for all containers in each workload.

```
resources:
  "cpu": "cpu_cores" # get cpu metrics
  "memory": "memory_bytes" # get memory metrics
aggregations:
- sources:
    type: "container"
    container: [ "utilization" ] # export container utilization
  levels:
  - mask:
      name: "container"
      builtIn: # aggregate on these labels
        exported_container: true
        exported_namespace: true
        workload_name: true
        workload_kind: true
        workload_api_group: true
        workload_api_version: true
     operation: "p95" # take the 95th percentile sample
```

Resulting metrics:

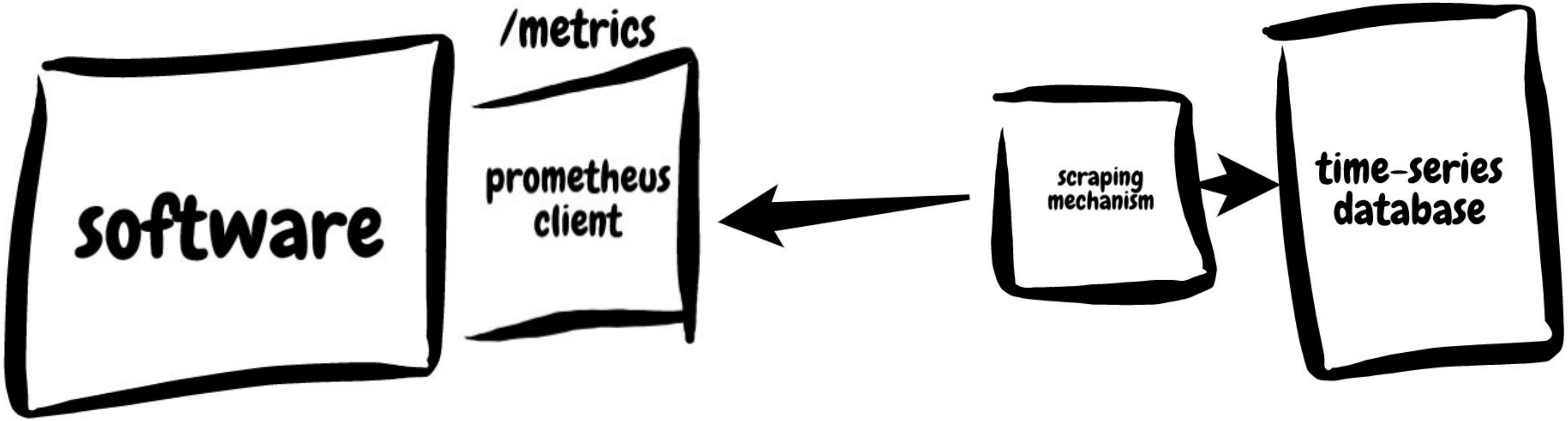
```
workload_p95_utilization_cpu_cores{exported_container="",exported_namespace="",workload_name="",
workload_kind="",workload_api_group="",workload_api_version=""}

workload_p95_utilization_memory_bytes{exported_container="",exported_namespace="",workload_name="",
workload_kind="",workload_api_group="",workload_api_version=""}
```

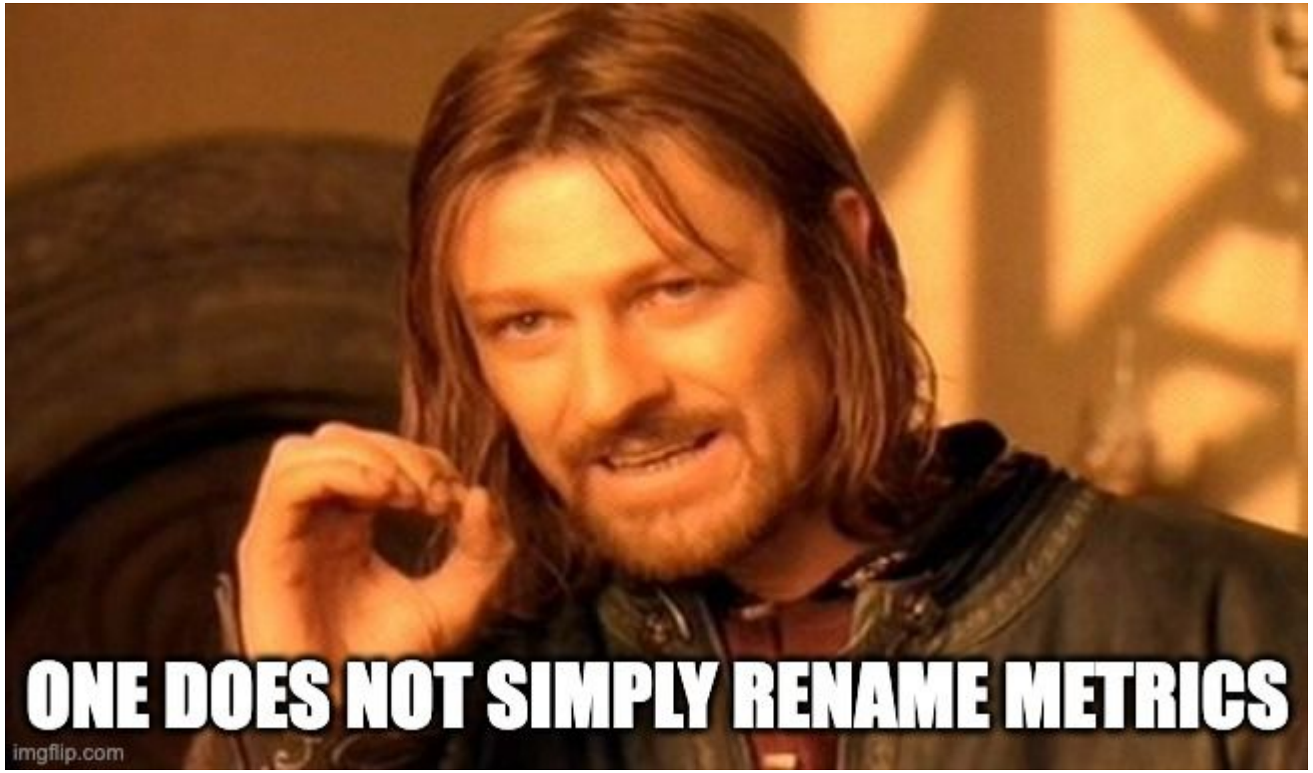
Metrics

Metrics

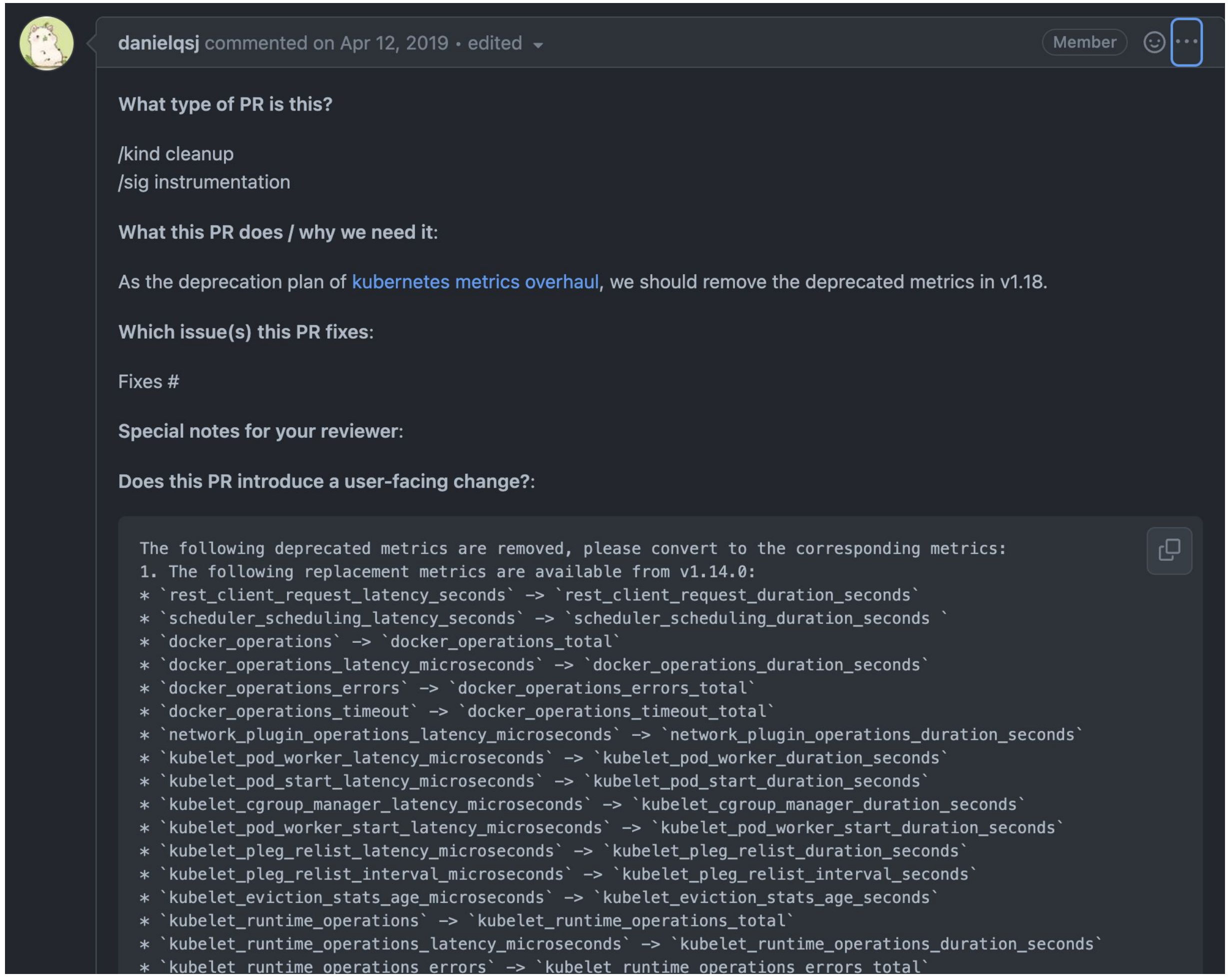
Kubernetes components integrate with Prometheus, a time-series based monitoring and alerting toolkit.



Metrics



Metrics Overhaul



danielqsj commented on Apr 12, 2019 • edited ▾ Member

What type of PR is this?

/kind cleanup
/sig instrumentation

What this PR does / why we need it:

As the deprecation plan of kubernetes metrics overhaul, we should remove the deprecated metrics in v1.18.

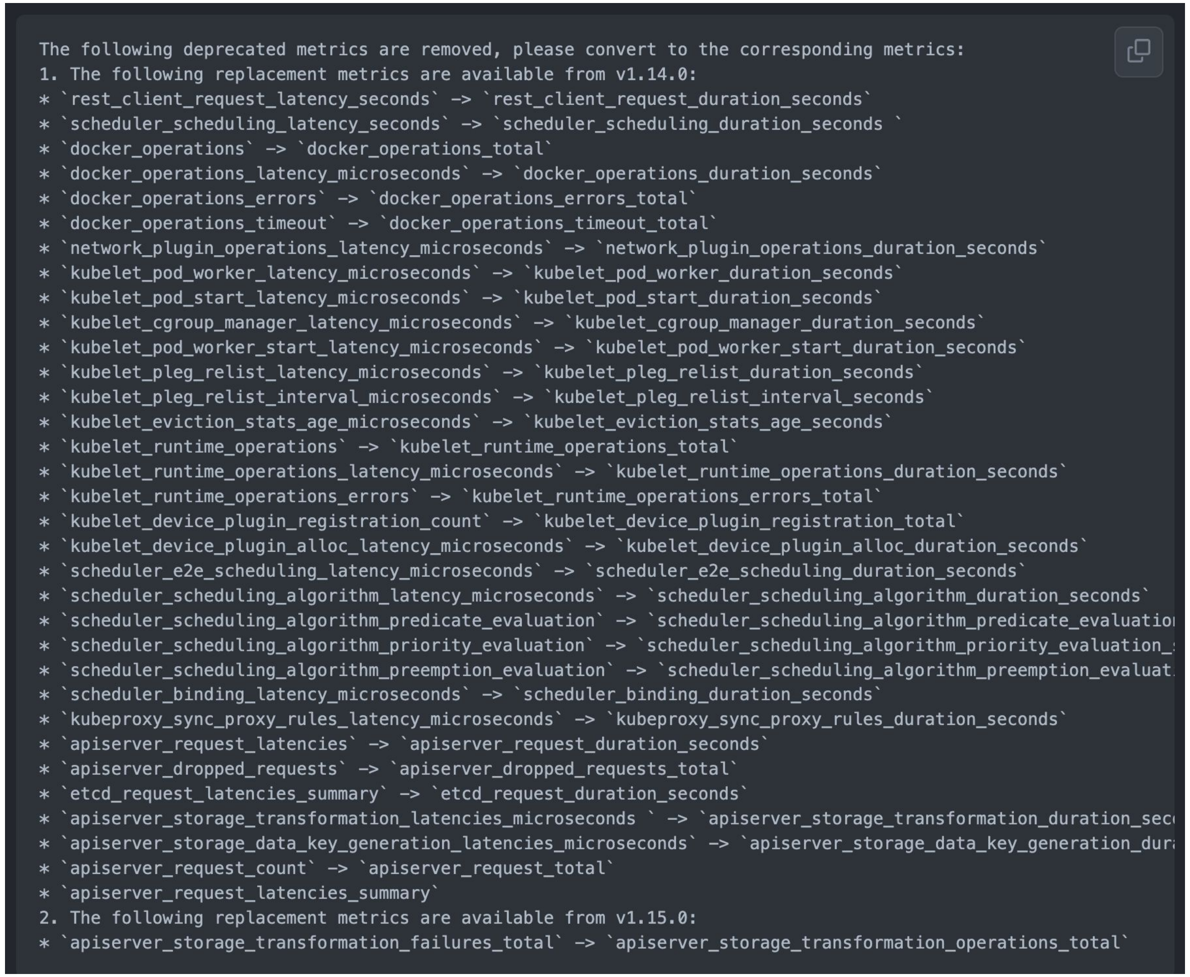
Which issue(s) this PR fixes:

Fixes #

Special notes for your reviewer:

Does this PR introduce a user-facing change?:

```
The following deprecated metrics are removed, please convert to the corresponding metrics:
1. The following replacement metrics are available from v1.14.0:
* `rest_client_request_latency_seconds` -> `rest_client_request_duration_seconds`
* `scheduler_scheduling_latency_seconds` -> `scheduler_scheduling_duration_seconds `
* `docker_operations` -> `docker_operations_total`
* `docker_operations_latency_microseconds` -> `docker_operations_duration_seconds`
* `docker_operations_errors` -> `docker_operations_errors_total`
* `docker_operations_timeout` -> `docker_operations_timeout_total`
* `network_plugin_operations_latency_microseconds` -> `network_plugin_operations_duration_seconds`
* `kubelet_pod_worker_latency_microseconds` -> `kubelet_pod_worker_duration_seconds`
* `kubelet_pod_start_latency_microseconds` -> `kubelet_pod_start_duration_seconds`
* `kubelet_cgroup_manager_latency_microseconds` -> `kubelet_cgroup_manager_duration_seconds`
* `kubelet_pod_worker_start_latency_microseconds` -> `kubelet_pod_worker_start_duration_seconds`
* `kubelet_pleg_relist_latency_microseconds` -> `kubelet_pleg_relist_duration_seconds`
* `kubelet_pleg_relist_interval_microseconds` -> `kubelet_pleg_relist_interval_seconds`
* `kubelet_eviction_stats_age_microseconds` -> `kubelet_eviction_stats_age_seconds`
* `kubelet_runtime_operations` -> `kubelet_runtime_operations_total`
* `kubelet_runtime_operations_latency_microseconds` -> `kubelet_runtime_operations_duration_seconds`
* `kubelet_runtime_operations_errors` -> `kubelet_runtime_operations_errors_total`
```



```
The following deprecated metrics are removed, please convert to the corresponding metrics:
1. The following replacement metrics are available from v1.14.0:
* `rest_client_request_latency_seconds` -> `rest_client_request_duration_seconds`
* `scheduler_scheduling_latency_seconds` -> `scheduler_scheduling_duration_seconds `
* `docker_operations` -> `docker_operations_total`
* `docker_operations_latency_microseconds` -> `docker_operations_duration_seconds`
* `docker_operations_errors` -> `docker_operations_errors_total`
* `docker_operations_timeout` -> `docker_operations_timeout_total`
* `network_plugin_operations_latency_microseconds` -> `network_plugin_operations_duration_seconds`
* `kubelet_pod_worker_latency_microseconds` -> `kubelet_pod_worker_duration_seconds`
* `kubelet_pod_start_latency_microseconds` -> `kubelet_pod_start_duration_seconds`
* `kubelet_cgroup_manager_latency_microseconds` -> `kubelet_cgroup_manager_duration_seconds`
* `kubelet_pod_worker_start_latency_microseconds` -> `kubelet_pod_worker_start_duration_seconds`
* `kubelet_pleg_relist_latency_microseconds` -> `kubelet_pleg_relist_duration_seconds`
* `kubelet_pleg_relist_interval_microseconds` -> `kubelet_pleg_relist_interval_seconds`
* `kubelet_eviction_stats_age_microseconds` -> `kubelet_eviction_stats_age_seconds`
* `kubelet_runtime_operations` -> `kubelet_runtime_operations_total`
* `kubelet_runtime_operations_latency_microseconds` -> `kubelet_runtime_operations_duration_seconds`
* `kubelet_runtime_operations_errors` -> `kubelet_runtime_operations_errors_total`
* `kubelet_device_plugin_registration_count` -> `kubelet_device_plugin_registration_total`
* `kubelet_device_plugin_alloc_latency_microseconds` -> `kubelet_device_plugin_alloc_duration_seconds`
* `scheduler_e2e_scheduling_latency_microseconds` -> `scheduler_e2e_scheduling_duration_seconds`
* `scheduler_scheduling_algorithm_latency_microseconds` -> `scheduler_scheduling_algorithm_duration_seconds`
* `scheduler_scheduling_algorithm_predicate_evaluation` -> `scheduler_scheduling_algorithm_predicate_evaluatio
* `scheduler_scheduling_algorithm_priority_evaluation` -> `scheduler_scheduling_algorithm_priority_evaluation_
* `scheduler_scheduling_algorithm_preemption_evaluation` -> `scheduler_scheduling_algorithm_preemption_evaluat
* `scheduler_binding_latency_microseconds` -> `scheduler_binding_duration_seconds`
* `kubeproxy_sync_proxy_rules_latency_microseconds` -> `kubeproxy_sync_proxy_rules_duration_seconds`
* `apiserver_request_latencies` -> `apiserver_request_duration_seconds`
* `apiserver_dropped_requests` -> `apiserver_dropped_requests_total`
* `etcd_request_latencies_summary` -> `etcd_request_duration_seconds`
* `apiserver_storage_transformation_latencies_microseconds ` -> `apiserver_storage_transformation_duration_sec
* `apiserver_storage_data_key_generation_latencies_microseconds` -> `apiserver_storage_data_key_generation_dur
* `apiserver_request_count` -> `apiserver_request_total`
* `apiserver_request_latencies_summary`
2. The following replacement metrics are available from v1.15.0:
* `apiserver_storage_transformation_failures_total` -> `apiserver_storage_transformation_operations_total`
```

Kubernetes Metrics Framework

- Provide a framework to express metric stability guarantees
- Provide automation around stability levels
- Provide a mechanism to centralize instrumentation related code and instrumentation processes
- https://bit.ly/metrics-stability

Stability Levels

- **Internal** (experimental) - *does **not** have* any stability guarantees
- **Alpha** - does not have any stability guarantees
- **Beta** (experimental) - *likely to have* stability guarantees
- **Stable** - has stability guarantees

What we are working on

- establishing the guarantees for the new stability levels
  - https://bit.ly/extending-stability
- auto-generated documentation for metrics
  - https://kubernetes.io/docs/reference/instrumentation/metrics/
- meta-metrics about registered metrics
- integrate native histograms

Logs

Structured Logs: What?

Before:

```
I0902 03:19:16.663200    1862 kubelet.go:1856] SyncLoop (ADD, "api"): "busybox-user-0-fd7df2b0-44be-44d3-8263-c63607950d99_security-context-test-4141(10a830b4-bdff-4671-a947-451346fe13fe)"
```

After (text):

```
I0902 21:38:49.937907    1821 kubelet.go:2053] "SyncLoop ADD" source="api" pods=[security-context-test-832/busybox-privileged-false-77fb495d-3037-4597-868d-d4b0e7a3eafd]
```

After (JSON):

```
{"ts":1630623419364.0852,"caller":"kubelet/kubelet.go:2053","msg":"SyncLoop ADD","v":2,"source":"api","pods":[{"name":"security-context-test-832","namespace":"busybox-privileged-false-77fb495d-3037-4597-868d-d4b0e7a3eafd"}]}
```

Structured Logs: When?

- Fully migrated Kubelet in 1.21 (#98976), kube-scheduler in 1.24 release (#105841), and became a stable feature in 1.26.
  - Includes static analysis to prevent regressions
- Deprecated klog-specific flags in Kubernetes components in 1.23, removal in 1.26
  - Reduce maintenance burden and complexity
  - Reduce number of flags needed to be supported by JSON and other formats
  - kubernetes/enhancements#2845

Structured Logs: With Context

- New in Kubernetes 1.24 as alpha feature:
  contextual logging (kubernetes/enhancements#3077)
- Logging through logger from call chain:
  - Attach key/value pairs and/or prefix to all log entries
  - Per-test output in unit test
- Implemented through new API in klog v2,
  fully interoperable with previous usage of klog.
- Future code migration will change to structured, contextual logging.

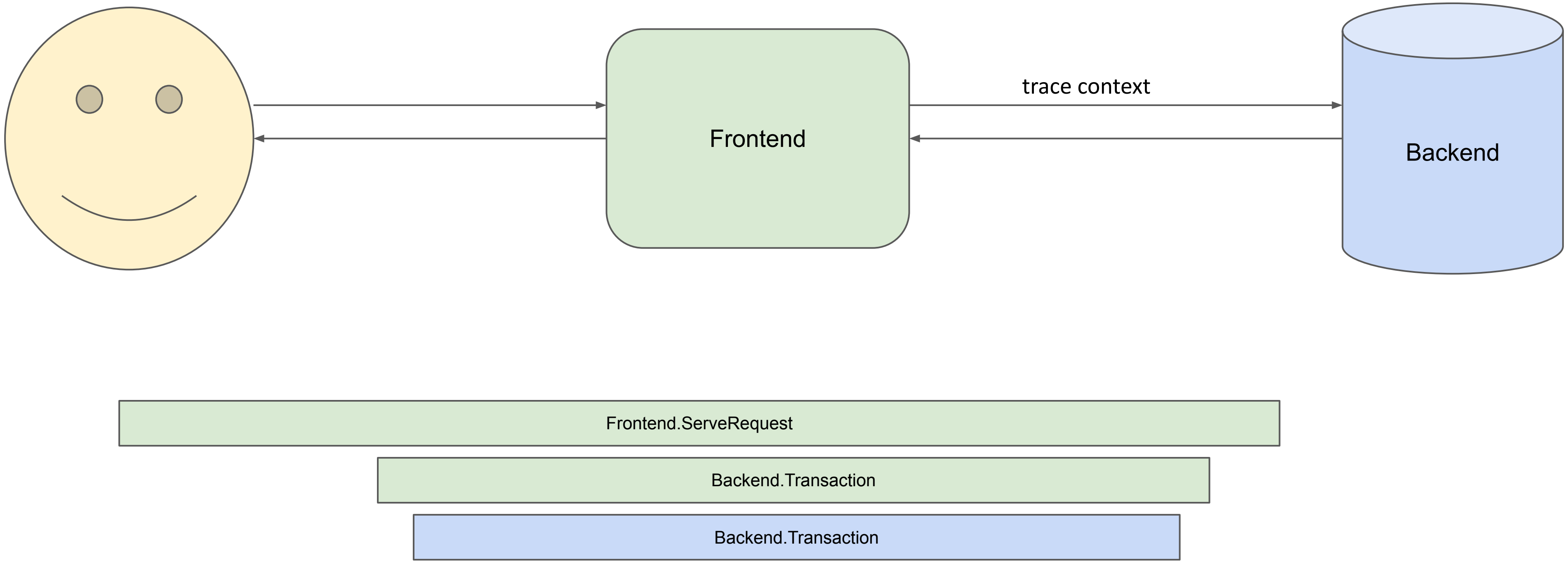
Structured Logs: Who?

- Spun off new **WG Structured Logging** to manage the structured log migration
  - **Organizers:**
    - Marek Siarkowicz (@serathius), Google
    - Patrick Ohly (@pohly), Intel
  - **Slack channel:** #wg-structured-logging
  - **Charter:** kubernetes/community/wg-structured-logging
  - **Biweekly meetings:** Thursdays at 15:30 British Time
- Need your help!

Traces

Traces

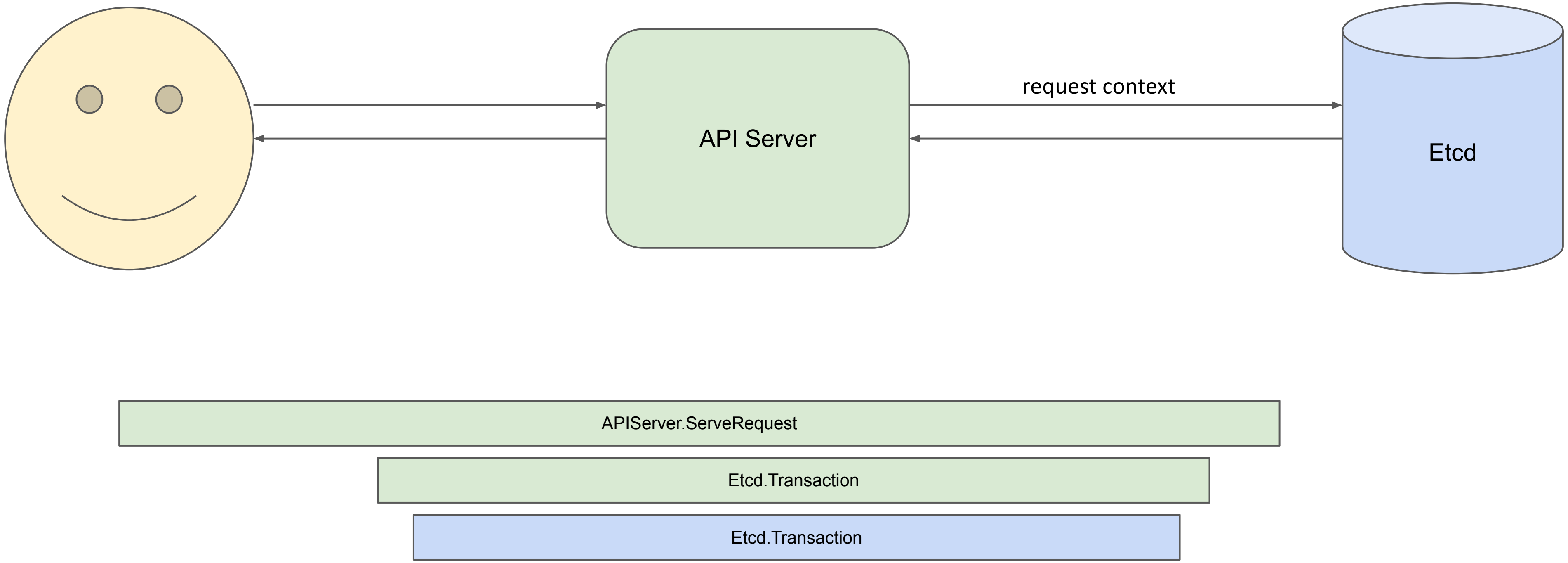
What is Distributed Tracing?



Traces

Tracing in Kubernetes

OpenTelemetry



Traces

API Server Tracing:

- **Beta** in 1.27
- Trace requests from the API Server to Etcd.

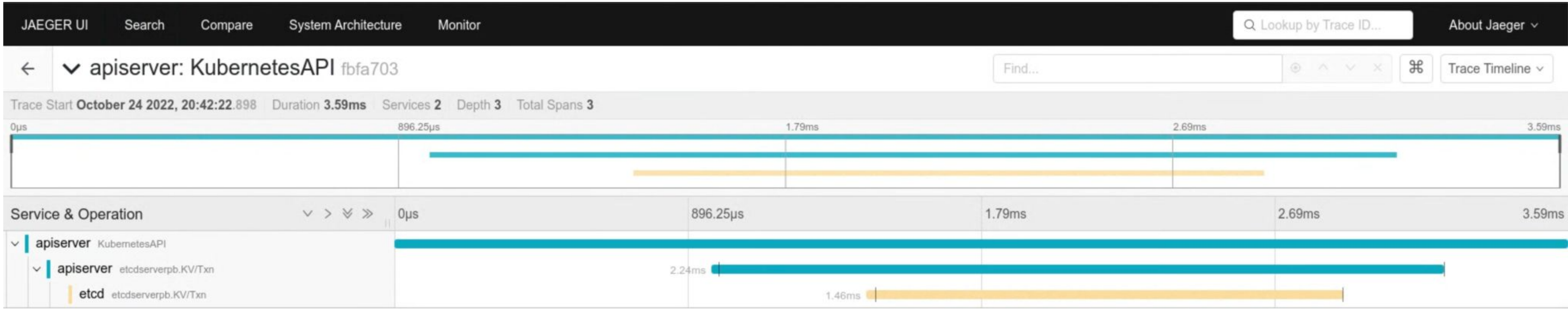
Kubelet Tracing

- **Beta** in 1.27
- Trace requests from the Kubelet to the Container Runtime

OpenTelemetry dependency updated to 1.0+ in K8s 1.26

Traces: API Server + Etcd

Kubernetes 1.22



apiserver: KubernetesAPI fbfa703

Trace Start **October 24 2022, 20:42:22**.898 Duration **3.59ms** Services **2** Depth **3** Total Spans **3**

apiserver KubernetesAPI

apiserver etcdserverpb.KV/Txn

etcd etcdserverpb.KV/Txn

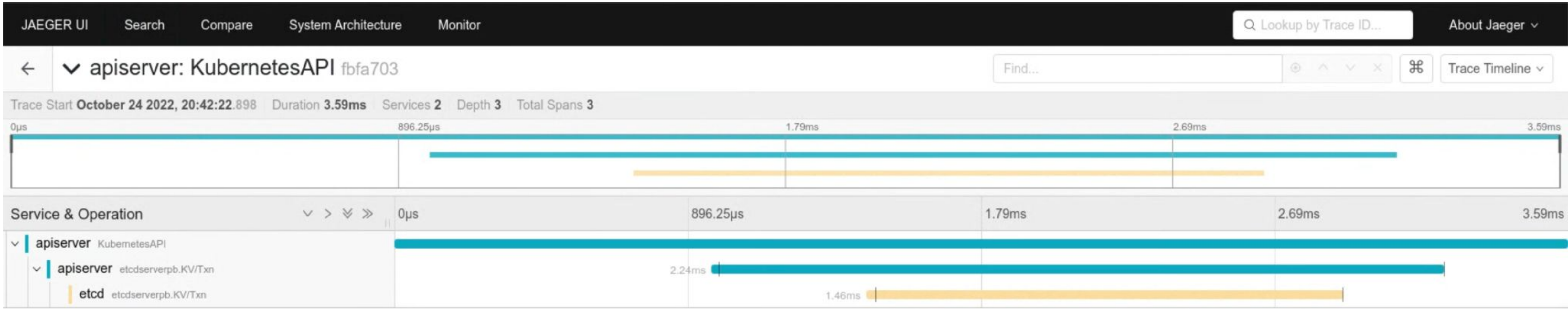
Traces: API Server + Etcd

Log-Based “Tracing”

```
Trace[1395114870]: "Create" url:/api/v1/nodes,user-agent:tracing.test/v0.0.0 (linux/amd64)
kubernetes/$Format,audit-id:c0282104-8068-44b1-a088-756b1253326d,client:127.0.0.1,accept:application/vnd.kubernetes.protobuf, */*,protocol:HTTP/2.0
(28-Oct-2022 13:42:35.876) (total time: 2ms):
Trace[1395114870]: ---"limitedReadBody succeeded" len:86 0ms (13:42:35.876)
Trace[1395114870]: ---"About to convert to expected version" 0ms (13:42:35.876)
Trace[1395114870]: ---"Conversion done" 0ms (13:42:35.876)
Trace[1395114870]: ---"About to store object in database" 0ms (13:42:35.876)
Trace[1395114870]: ["Create etcd3" audit-id:c0282104-8068-44b1-a088-756b1253326d,key:/minions/fake,type:*core.Node,resource:nodes 2ms (13:42:35.876)
Trace[1395114870]:  ---"About to Encode" 0ms (13:42:35.876)
Trace[1395114870]:  ---"Encode succeeded" len:177 0ms (13:42:35.876)
Trace[1395114870]:  ---"TransformToStorage succeeded" 0ms (13:42:35.876)
Trace[1395114870]:  ---"Txn call succeeded" 1ms (13:42:35.878)
Trace[1395114870]:  ---"decode succeeded" len:177 0ms (13:42:35.878)]
Trace[1395114870]: ---"Write to database call succeeded" len:86 0ms (13:42:35.878)
Trace[1395114870]: ---"About to write a response" 0ms (13:42:35.878)
Trace[1395114870]: ---"Writing http response done" 0ms (13:42:35.878)
Trace[1395114870]: [2.771143ms] [2.771143ms] END
```

Traces: API Server + Etcd

Kubernetes 1.22



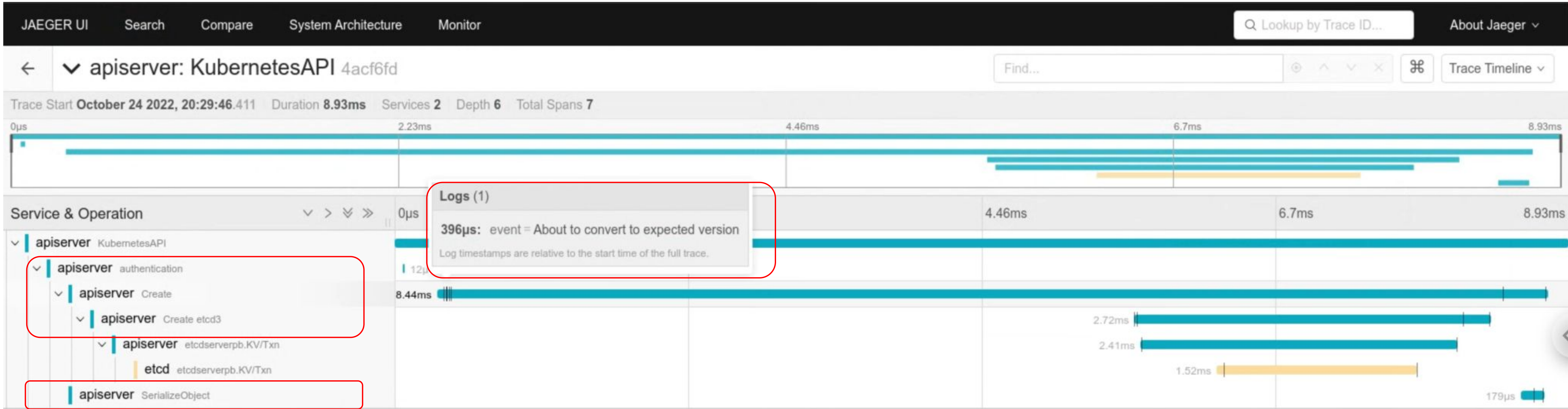
apiserver: KubernetesAPI fbfa703

Trace Start **October 24 2022, 20:42:22**.898 Duration **3.59ms** Services **2** Depth **3** Total Spans **3**

Service & Operation

- apiserver KubernetesAPI
  - apiserver etcdserverpb.KV/Txn — 2.24ms
    - etcd etcdserverpb.KV/Txn — 1.46ms

Kubernetes 1.26



apiserver: KubernetesAPI 4acf6fd

Trace Start **October 24 2022, 20:29:46**.411 Duration **8.93ms** Services **2** Depth **6** Total Spans **7**

Service & Operation

- apiserver KubernetesAPI
  - apiserver authentication — 12μ
  - apiserver Create — 8.44ms
    - apiserver Create etcd3 — 2.72ms
      - apiserver etcdserverpb.KV/Txn — 2.41ms
        - etcd etcdserverpb.KV/Txn — 1.52ms
  - apiserver SerializeObject — 179μs

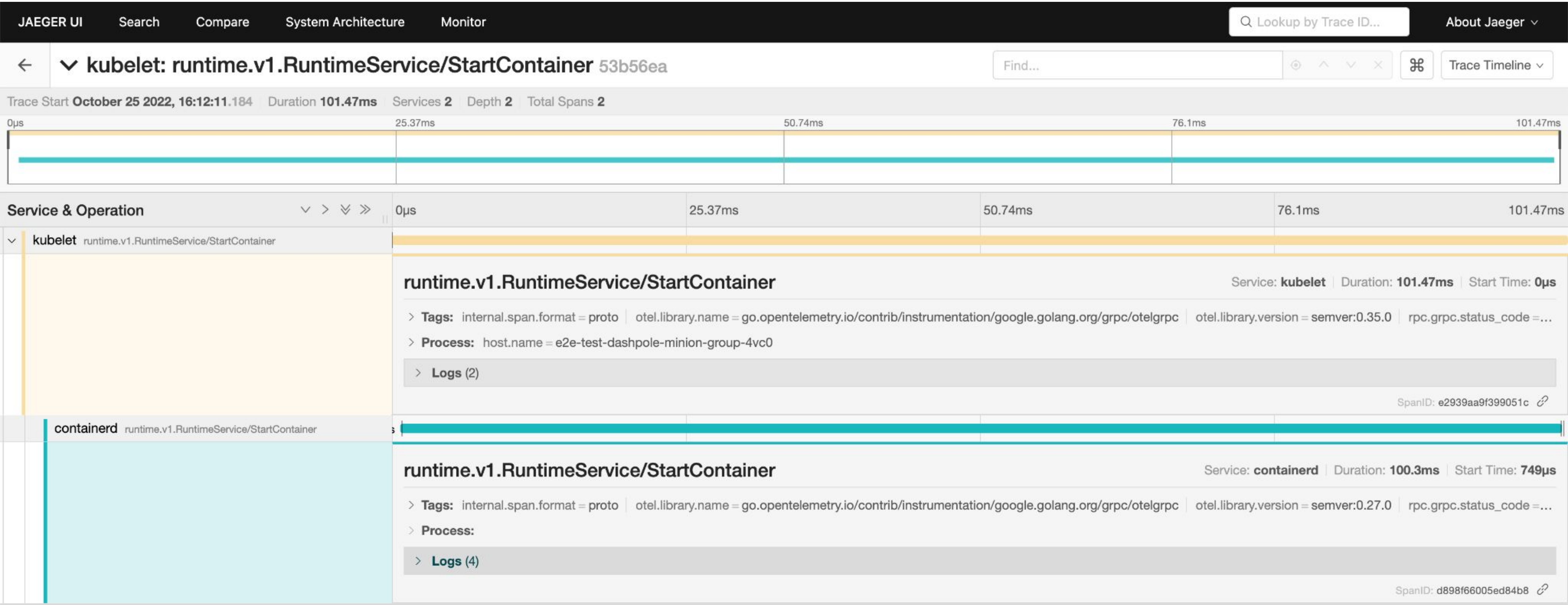
Logs (1)

396μs: event = About to convert to expected version

Log timestamps are relative to the start time of the full trace.

Traces: Kubelet + Container Runtime

Alpha: CRI Traces



kubelet: runtime.v1.RuntimeService/StartContainer 53b56ea

Trace Start October 25 2022, 16:12:11.184 | Duration 101.47ms | Services 2 | Depth 2 | Total Spans 2

kubelet runtime.v1.RuntimeService/StartContainer

runtime.v1.RuntimeService/StartContainer Service: kubelet | Duration: 101.47ms | Start Time: 0μs

> Tags: internal.span.format = proto | otel.library.name = go.opentelemetry.io/contrib/instrumentation/google.golang.org/grpc/otelgrpc | otel.library.version = semver:0.35.0 | rpc.grpc.status_code =...

> Process: host.name = e2e-test-dashpole-minion-group-4vc0

> Logs (2)

SpanID: e2939aa9f399051c

containerd runtime.v1.RuntimeService/StartContainer

runtime.v1.RuntimeService/StartContainer Service: containerd | Duration: 100.3ms | Start Time: 749μs

> Tags: internal.span.format = proto | otel.library.name = go.opentelemetry.io/contrib/instrumentation/google.golang.org/grpc/otelgrpc | otel.library.version = semver:0.27.0 | rpc.grpc.status_code =...

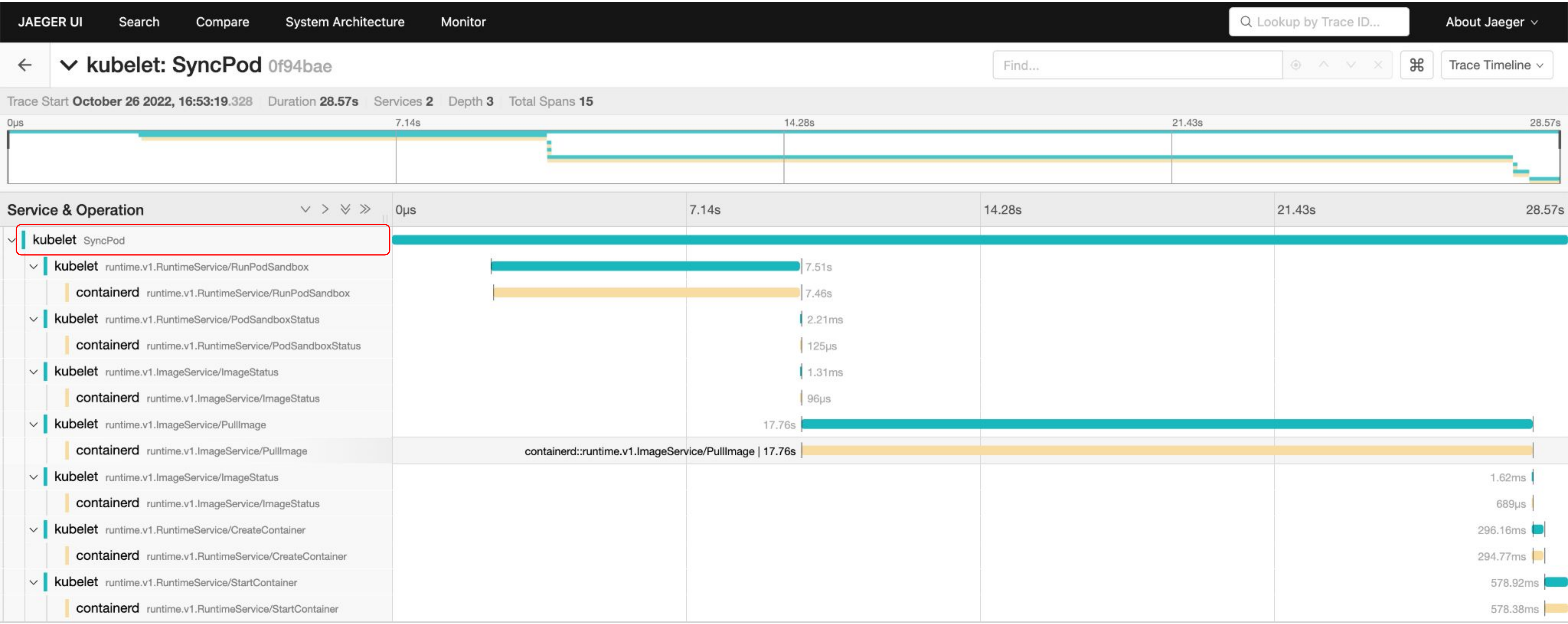
> Process:

> Logs (4)

SpanID: d898f66005ed84b8

Traces: Kubelet + Container Runtime

Proof of Concept: Complete Pod traces



kubelet: SyncPod 0f94bae

Trace Start October 26 2022, 16:53:19.328 Duration 28.57s Services 2 Depth 3 Total Spans 15

Service & Operation	Duration
kubelet SyncPod	
kubelet runtime.v1.RuntimeService/RunPodSandbox	7.51s
containerd runtime.v1.RuntimeService/RunPodSandbox	7.46s
kubelet runtime.v1.RuntimeService/PodSandboxStatus	2.21ms
containerd runtime.v1.RuntimeService/PodSandboxStatus	125µs
kubelet runtime.v1.ImageService/ImageStatus	1.31ms
containerd runtime.v1.ImageService/ImageStatus	96µs
kubelet runtime.v1.ImageService/PullImage	17.76s
containerd runtime.v1.ImageService/PullImage	containerd::runtime.v1.ImageService/PullImage | 17.76s
kubelet runtime.v1.ImageService/ImageStatus	1.62ms
containerd runtime.v1.ImageService/ImageStatus	689µs
kubelet runtime.v1.RuntimeService/CreateContainer	296.16ms
containerd runtime.v1.RuntimeService/CreateContainer	294.77ms
kubelet runtime.v1.RuntimeService/StartContainer	578.92ms
containerd runtime.v1.RuntimeService/StartContainer	578.38ms

Traces

Future Plans:

- Add kubelet spans to track “create pod” instead of just “create container”
- Link from Metrics to Traces with **Prometheus Exemplars**
- Link from Logs to Traces with **Trace + Span IDs in Logs**

KubeCon | CloudNativeCon

Europe 2023

Get involved!

How to contribute

- Attend our SIG meetings!
- Participate in reviews, issues, and docs!
- `kube-state-metrics`, `prometheus-adapter`, and `metrics-server` are seeking new contributors
  - Contact **Damien Grisonnet** (@dgrisonnet)
- `contextual logging` is seeking new contributors
  - Contact **Patrick Ohly** (@pohly)
- `usage-metrics-collector` is seeking new contributors
  - Contact **Elana Hashman** (@ehashman)

Where to find us

- **SIG Meetings:**
  - Regular meeting, alternating **biweekly** on **Thursdays at 9:30am Pacific Time**
  - Triage meeting, alternating **biweekly** on **Thursdays at 9:30am Pacific Time**
- **Slack channel:** #sig-instrumentation
- **Mailing list:** kubernetes-sig-instrumentation
- **Chairs:** @ehashman and @logicalhan
- **Tech leads:** @dashpole and @dgrisonnet

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

KubeCon | CloudNativeCon

Europe 2023