containerday
Bologna, 12 ottobre 2023

Viaggio all'interno di Prometheus

Architettura, funzionamento, novità e prospettive future nel monitoraggio cloud native

Martino Fornasa

Independent Consultant - Trainer

formazione-kubernetes.it | fornasa.it

@mfornasa

joind.in/talk/16f8c


Suono 1


Suono 2

General
Home
Documentation

Monitoring
Tactical Overview
Service Detail
Host Detail
Status Overview
Status Summary
Status Grid
Status Map
3-D Status Map

Service Problems
Host Problems
Network Outages

Comments
Downtime

Process Info
Performance Info
Scheduling Queue

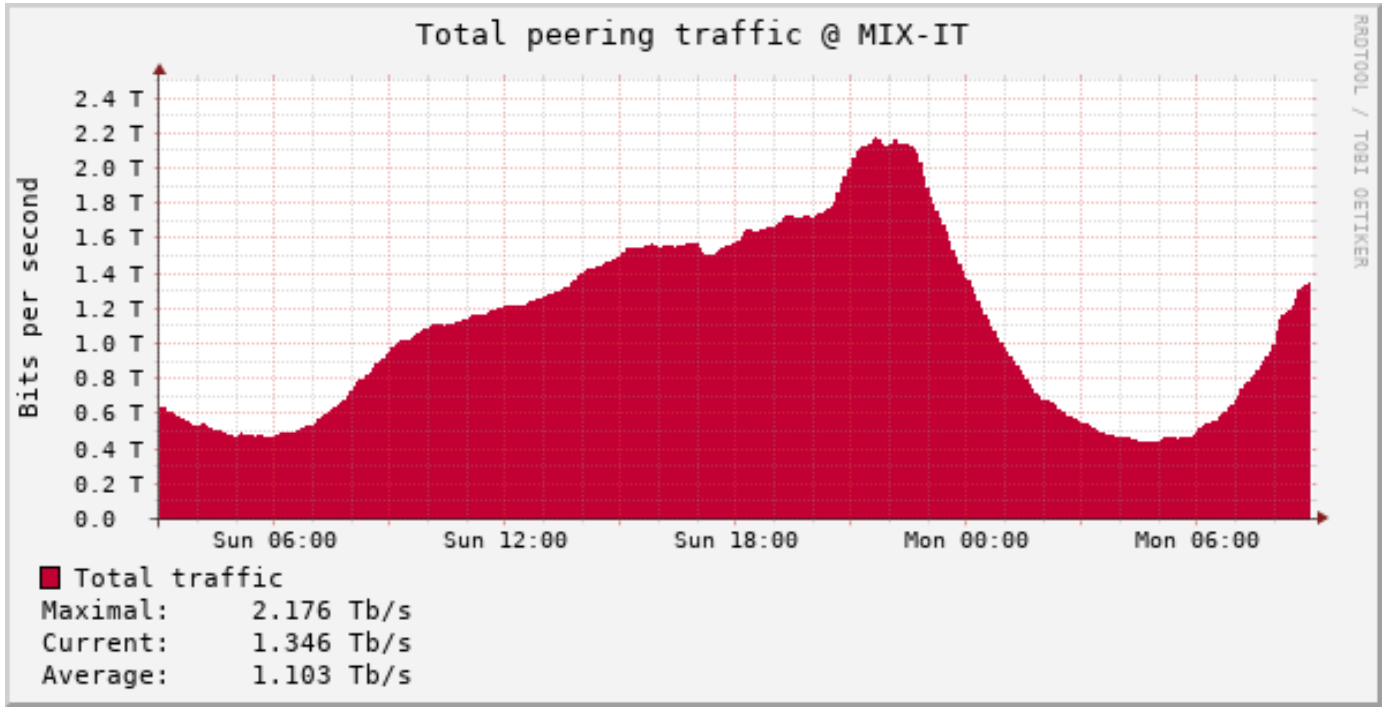
Reporting
Trends
Availability
Alert Histogram
Alert History
Alert Summary
Notifications
Event Log

Configuration
View Config

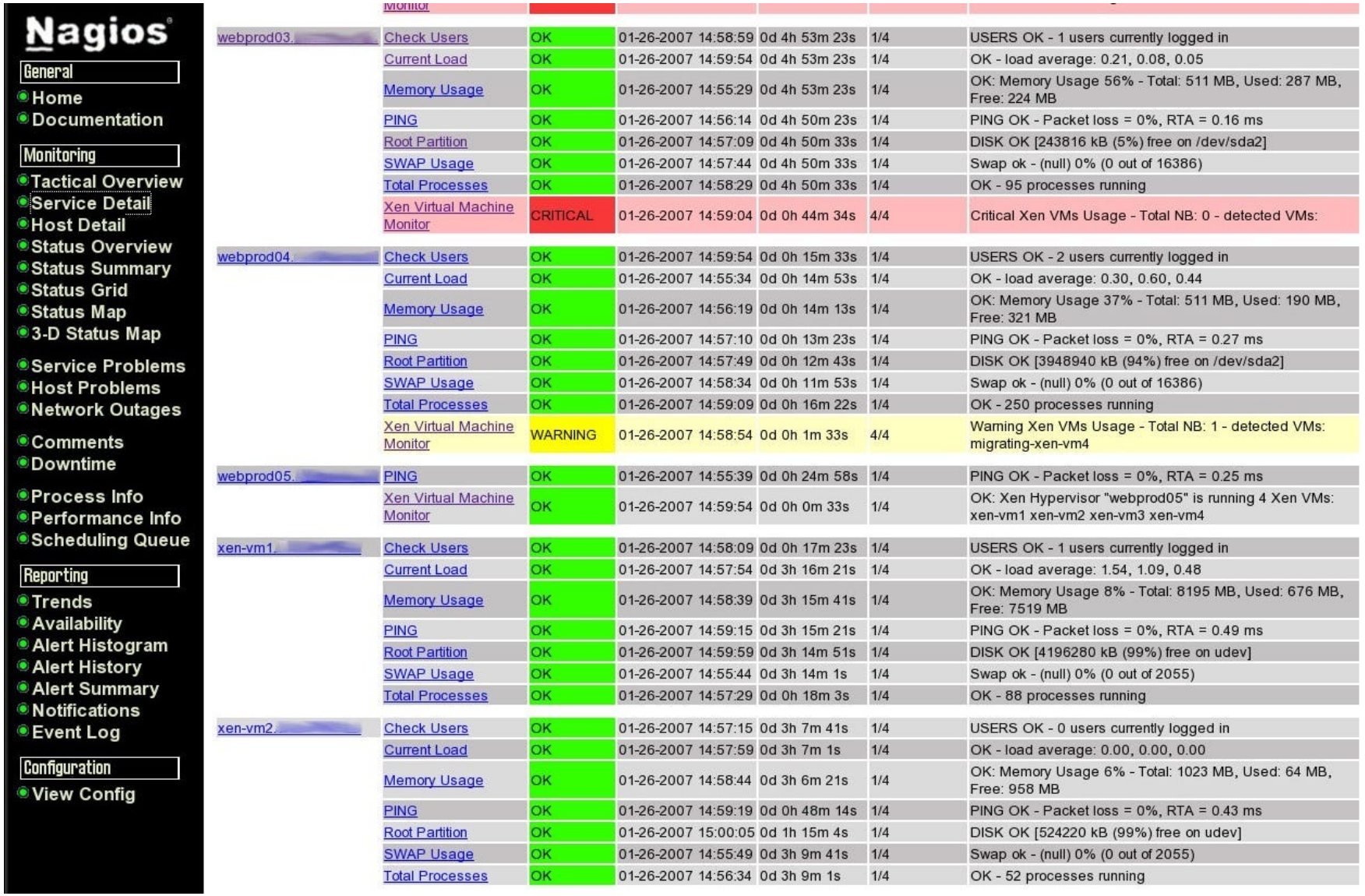
Monitor						
webprod03.	Check Users	OK	01-26-2007 14:58:59	0d 4h 53m 23s	1/4	USERS OK - 1 users currently logged in
	Current Load	OK	01-26-2007 14:59:54	0d 4h 53m 23s	1/4	OK - load average: 0.21, 0.08, 0.05
	Memory Usage	OK	01-26-2007 14:55:29	0d 4h 53m 23s	1/4	OK: Memory Usage 56% - Total: 511 MB, Used: 287 MB, Free: 224 MB
	PING	OK	01-26-2007 14:56:14	0d 4h 50m 23s	1/4	PING OK - Packet loss = 0%, RTA = 0.16 ms
	Root Partition	OK	01-26-2007 14:57:09	0d 4h 50m 33s	1/4	DISK OK [243816 kB (5%) free on /dev/sda2]
	SWAP Usage	OK	01-26-2007 14:57:44	0d 4h 50m 33s	1/4	Swap ok - (null) 0% (0 out of 16386)
	Total Processes	OK	01-26-2007 14:58:29	0d 4h 50m 33s	1/4	OK - 95 processes running
	Xen Virtual Machine Monitor	CRITICAL	01-26-2007 14:59:04	0d 0h 44m 34s	4/4	Critical Xen VMs Usage - Total NB: 0 - detected VMs:
webprod04.	Check Users	OK	01-26-2007 14:59:54	0d 0h 15m 33s	1/4	USERS OK - 2 users currently logged in
	Current Load	OK	01-26-2007 14:55:34	0d 0h 14m 53s	1/4	OK - load average: 0.30, 0.60, 0.44
	Memory Usage	OK	01-26-2007 14:56:19	0d 0h 14m 13s	1/4	OK: Memory Usage 37% - Total: 511 MB, Used: 190 MB, Free: 321 MB
	PING	OK	01-26-2007 14:57:10	0d 0h 13m 23s	1/4	PING OK - Packet loss = 0%, RTA = 0.27 ms
	Root Partition	OK	01-26-2007 14:57:49	0d 0h 12m 43s	1/4	DISK OK [3948940 kB (94%) free on /dev/sda2]
	SWAP Usage	OK	01-26-2007 14:58:34	0d 0h 11m 53s	1/4	Swap ok - (null) 0% (0 out of 16386)
	Total Processes	OK	01-26-2007 14:59:09	0d 0h 16m 22s	1/4	OK - 250 processes running
	Xen Virtual Machine Monitor	WARNING	01-26-2007 14:58:54	0d 0h 1m 33s	4/4	Warning Xen VMs Usage - Total NB: 1 - detected VMs: migrating-xen-vm4
webprod05.	PING	OK	01-26-2007 14:55:39	0d 0h 24m 58s	1/4	PING OK - Packet loss = 0%, RTA = 0.25 ms
	Xen Virtual Machine Monitor	OK	01-26-2007 14:59:54	0d 0h 0m 33s	1/4	OK: Xen Hypervisor "webprod05" is running 4 Xen VMs: xen-vm1 xen-vm2 xen-vm3 xen-vm4
xen-vm1.	Check Users	OK	01-26-2007 14:58:09	0d 0h 17m 23s	1/4	USERS OK - 1 users currently logged in
	Current Load	OK	01-26-2007 14:57:54	0d 3h 16m 21s	1/4	OK - load average: 1.54, 1.09, 0.48
	Memory Usage	OK	01-26-2007 14:58:39	0d 3h 15m 41s	1/4	OK: Memory Usage 8% - Total: 8195 MB, Used: 676 MB, Free: 7519 MB
	PING	OK	01-26-2007 14:59:15	0d 3h 15m 21s	1/4	PING OK - Packet loss = 0%, RTA = 0.49 ms
	Root Partition	OK	01-26-2007 14:59:59	0d 3h 14m 51s	1/4	DISK OK [4196280 kB (99%) free on udev]
	SWAP Usage	OK	01-26-2007 14:55:44	0d 3h 14m 1s	1/4	Swap ok - (null) 0% (0 out of 2055)
	Total Processes	OK	01-26-2007 14:57:29	0d 0h 18m 3s	1/4	OK - 88 processes running
xen-vm2.	Check Users	OK	01-26-2007 14:57:15	0d 3h 7m 41s	1/4	USERS OK - 0 users currently logged in
	Current Load	OK	01-26-2007 14:57:59	0d 3h 7m 1s	1/4	OK - load average: 0.00, 0.00, 0.00
	Memory Usage	OK	01-26-2007 14:58:44	0d 3h 6m 21s	1/4	OK: Memory Usage 6% - Total: 1023 MB, Used: 64 MB, Free: 958 MB
	PING	OK	01-26-2007 14:59:19	0d 0h 48m 14s	1/4	PING OK - Packet loss = 0%, RTA = 0.43 ms
	Root Partition	OK	01-26-2007 15:00:05	0d 1h 15m 4s	1/4	DISK OK [524220 kB (99%) free on udev]
	SWAP Usage	OK	01-26-2007 14:55:49	0d 3h 9m 41s	1/4	Swap ok - (null) 0% (0 out of 2055)

Some History

- MRTG (1995)
- **RRDTool (1999)**

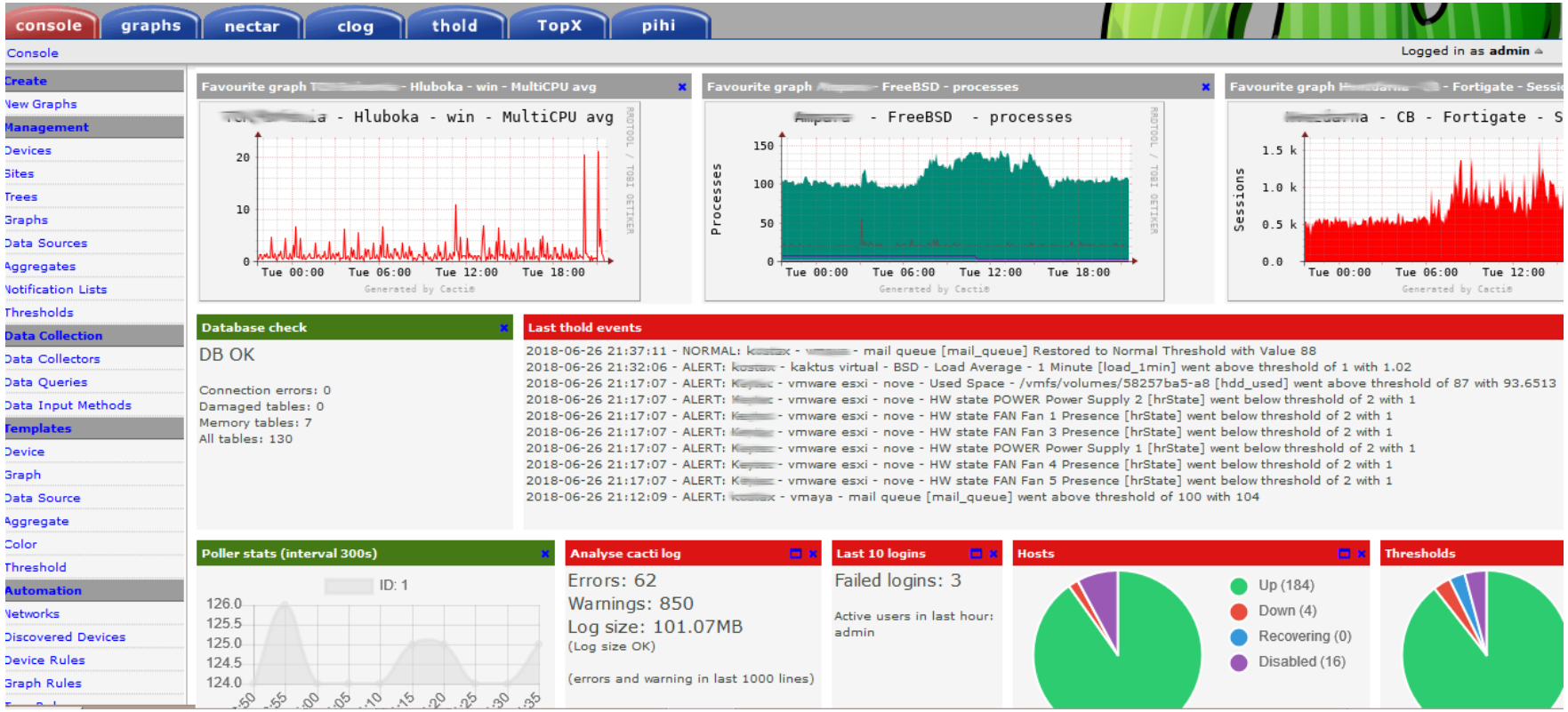


- **Nagios (1999)**
- Cacti (2001)
- Zabbix (2001)



webprod03.	Check Users	OK	01-26-2007 14:58:59	0d 4h 53m 23s	1/4	USERS OK - 1 users currently logged in
	Current Load	OK	01-26-2007 14:59:54	0d 4h 53m 23s	1/4	OK - load average: 0.21, 0.08, 0.05
	Memory Usage	OK	01-26-2007 14:55:29	0d 4h 53m 23s	1/4	OK: Memory Usage 56% - Total: 511 MB, Used: 287 MB, Free: 224 MB
	PING	OK	01-26-2007 14:56:14	0d 4h 50m 23s	1/4	PING OK - Packet loss = 0%, RTA = 0.16 ms
	Root Partition	OK	01-26-2007 14:57:09	0d 4h 50m 33s	1/4	DISK OK [243816 kB (5%) free on /dev/sda2]
	SWAP Usage	OK	01-26-2007 14:57:44	0d 4h 50m 33s	1/4	Swap ok - (null) 0% (0 out of 16386)
	Total Processes	OK	01-26-2007 14:58:29	0d 4h 50m 33s	1/4	OK - 95 processes running
	Xen Virtual Machine Monitor	CRITICAL	01-26-2007 14:59:04	0d 0h 44m 34s	4/4	Critical Xen VMs Usage - Total NB: 0 - detected VMs:
webprod04.	Check Users	OK	01-26-2007 14:59:54	0d 0h 15m 33s	1/4	USERS OK - 2 users currently logged in
	Current Load	OK	01-26-2007 14:55:34	0d 0h 14m 53s	1/4	OK - load average: 0.30, 0.60, 0.44
	Memory Usage	OK	01-26-2007 14:56:19	0d 0h 14m 13s	1/4	OK: Memory Usage 37% - Total: 511 MB, Used: 190 MB, Free: 321 MB
	PING	OK	01-26-2007 14:57:10	0d 0h 13m 23s	1/4	PING OK - Packet loss = 0%, RTA = 0.27 ms
	Root Partition	OK	01-26-2007 14:57:49	0d 0h 12m 43s	1/4	DISK OK [3948940 kB (94%) free on /dev/sda2]
	SWAP Usage	OK	01-26-2007 14:58:34	0d 0h 11m 53s	1/4	Swap ok - (null) 0% (0 out of 16386)
	Total Processes	OK	01-26-2007 14:59:09	0d 0h 16m 22s	1/4	OK - 250 processes running
	Xen Virtual Machine Monitor	WARNING	01-26-2007 14:58:54	0d 0h 1m 33s	4/4	Warning Xen VMs Usage - Total NB: 1 - detected VMs: migrating-xen-vm4
webprod05.	PING	OK	01-26-2007 14:55:39	0d 0h 24m 58s	1/4	PING OK - Packet loss = 0%, RTA = 0.25 ms
	Xen Virtual Machine Monitor	OK	01-26-2007 14:59:54	0d 0h 0m 33s	1/4	OK: Xen Hypervisor "webprod05" is running 4 Xen VMs: xen-vm1 xen-vm2 xen-vm3 xen-vm4
xen-vm1.	Check Users	OK	01-26-2007 14:58:09	0d 0h 17m 23s	1/4	USERS OK - 1 users currently logged in
	Current Load	OK	01-26-2007 14:57:54	0d 3h 16m 21s	1/4	OK - load average: 1.54, 1.09, 0.48
	Memory Usage	OK	01-26-2007 14:58:39	0d 3h 15m 41s	1/4	OK: Memory Usage 8% - Total: 8195 MB, Used: 676 MB, Free: 7519 MB
	PING	OK	01-26-2007 14:59:15	0d 3h 15m 21s	1/4	PING OK - Packet loss = 0%, RTA = 0.49 ms
	Root Partition	OK	01-26-2007 14:59:59	0d 3h 14m 51s	1/4	DISK OK [4196280 kB (99%) free on udev]
	SWAP Usage	OK	01-26-2007 14:55:44	0d 3h 14m 1s	1/4	Swap ok - (null) 0% (0 out of 2055)
	Total Processes	OK	01-26-2007 14:57:29	0d 0h 18m 3s	1/4	OK - 88 processes running
xen-vm2.	Check Users	OK	01-26-2007 14:57:15	0d 3h 7m 41s	1/4	USERS OK - 0 users currently logged in
	Current Load	OK	01-26-2007 14:57:59	0d 3h 7m 1s	1/4	OK - load average: 0.00, 0.00, 0.00
	Memory Usage	OK	01-26-2007 14:58:44	0d 3h 6m 21s	1/4	OK: Memory Usage 6% - Total: 1023 MB, Used: 64 MB, Free: 958 MB
	PING	OK	01-26-2007 14:59:19	0d 0h 48m 14s	1/4	PING OK - Packet loss = 0%, RTA = 0.43 ms
	Root Partition	OK	01-26-2007 15:00:05	0d 1h 15m 4s	1/4	DISK OK [524220 kB (99%) free on udev]
	SWAP Usage	OK	01-26-2007 14:55:49	0d 3h 9m 41s	1/4	Swap ok - (null) 0% (0 out of 2055)
	Total Processes	OK	01-26-2007 14:56:34	0d 3h 9m 1s	1/4	OK - 52 processes running

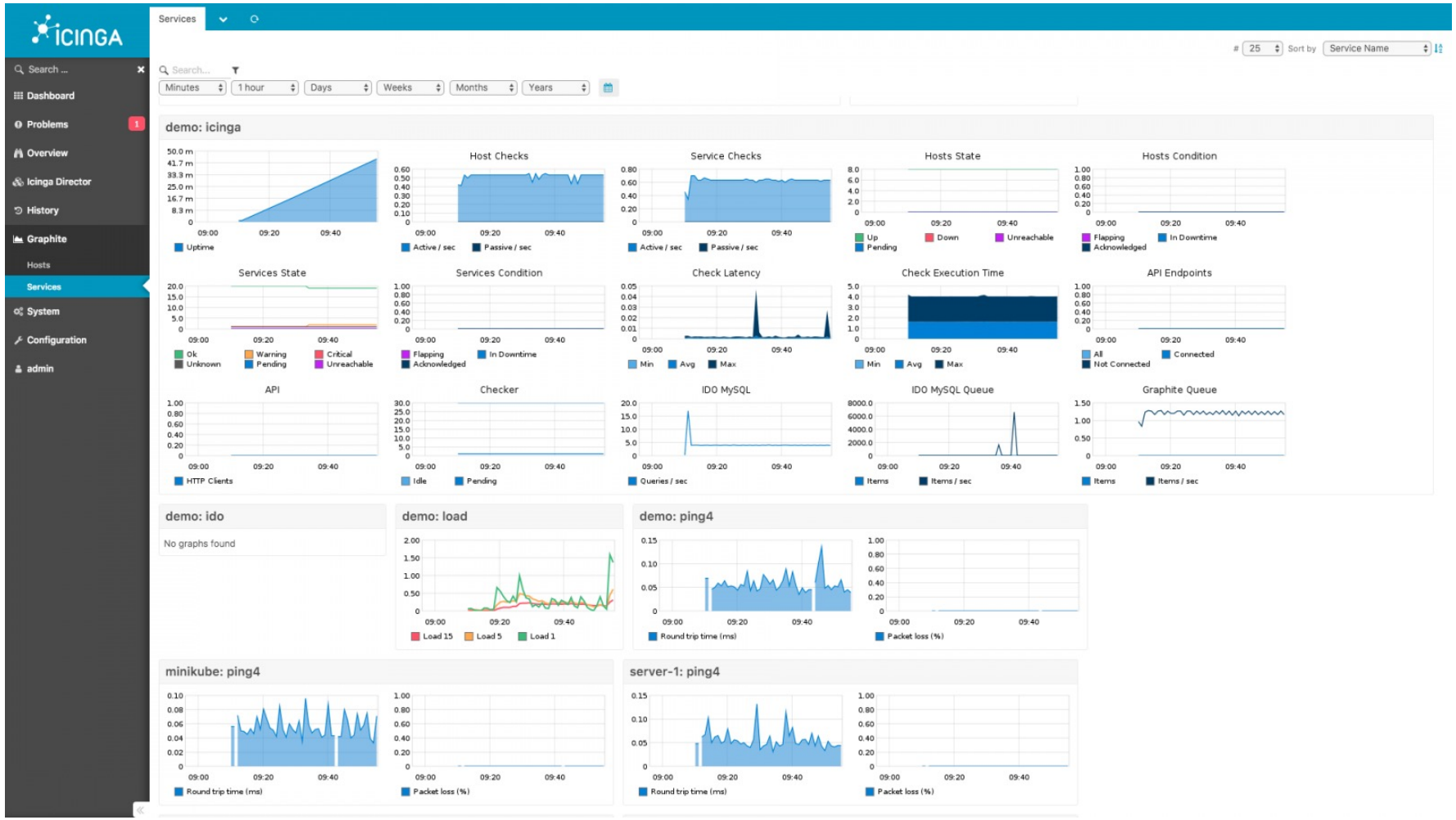
- Nagios (1999)
- **Cacti (2001)**
- Zabbix (2001)



- Nagios (1999)
- Cacti (2001)
- **Zabbix (2001)**

- **Graphite (2008)**
- Icinga (2009)

- Graphite (2008)
- **Icinga (2009)**



Prometheus History

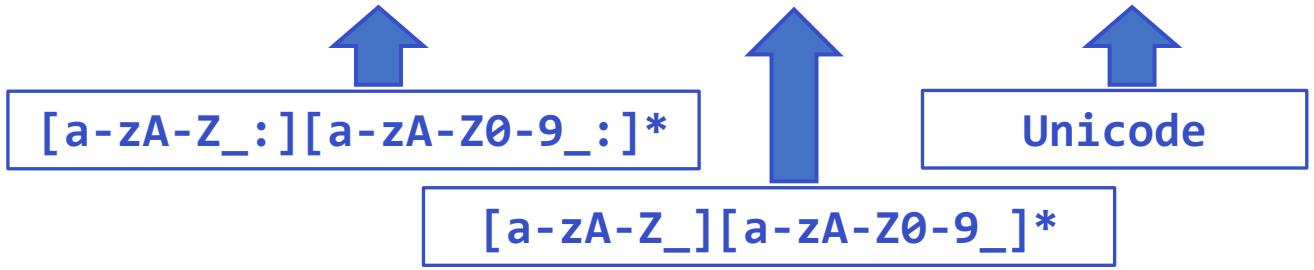
- Developed in 2012 at SoundCloud
- Inspired by *borgmon* at Google
- Joined the CNCF in 2016 (second project after Kubernetes)

- Main goal: To handle a **complex** and **dynamic** environment (cloud, cloud native, microservices, serverless, …)

Data Model

Docs: Data Model

Metric




```
api_http_requests_total{method="POST", handler="/messages"}
```

Sample

(timestamp *milliseconds*, value *float64*)

Metric Types

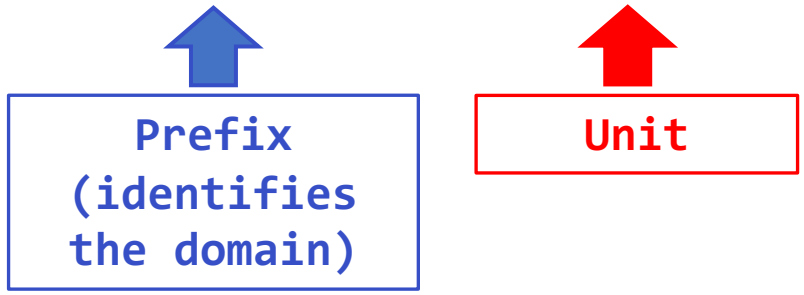
Docs: Metric Types

- Counter -> monotonically increasing
- Gauge -> single numerical value that can arbitrarily go up and down
- Histogram -> sampled observations in buckets
- Native Histogram (new) -> dynamic buckets, higher resolution

Metric and Label Naming Conventions [1]

Docs: Best Practices

- http_request_duration_seconds



- process_cpu_seconds_total — **Accumulating count with unit**
- http_requests_total — **Unit-less Accumulating count**

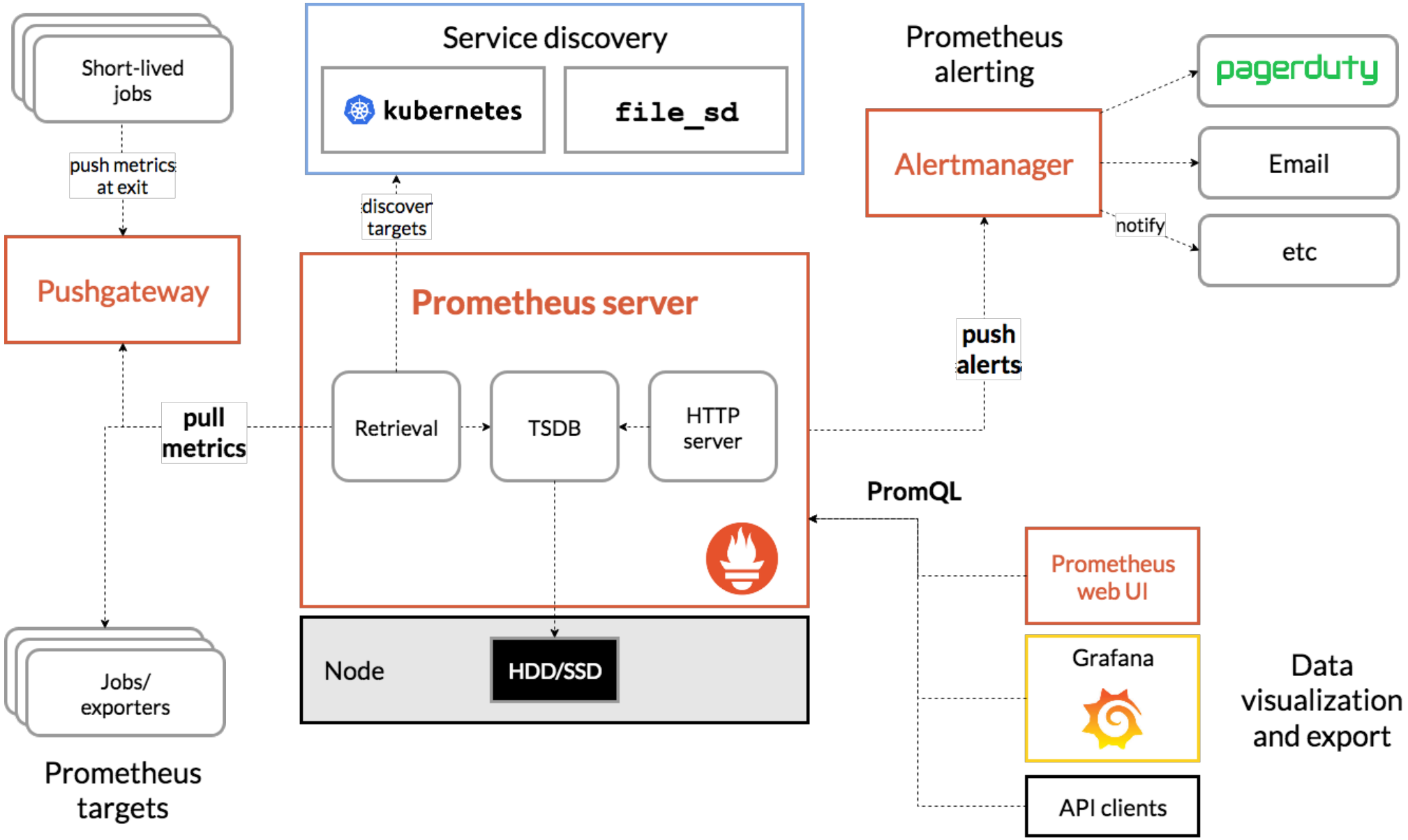
SEE ALSO: Semantic Conventions in OpenTelemetry

Metric and Label Naming Conventions [2]

Docs: Best Practices

```
http_request_total{
  container="grafana-proxy”,
  endpoint="nginx-http",
  handler="/api/annotations",
  instance="10.42.66.101:8080",
  job="rancher-monitoring-grafana",
  method="get",
  namespace="cattle-monitoring-system",
  pod="rancher-monitoring-grafana-67d56665c-fwskr",
  service="rancher-monitoring-grafana",
  statuscode="200"}
```

CAUTION: Remember that every unique combination of key-value label pairs represents a new time series, which can dramatically increase the amount of data stored. Do not use labels to store dimensions with high cardinality (many different label values), such as user IDs, email addresses, or other unbounded sets of values.


Short-lived jobs
push metrics at exit
Pushgateway
Service discovery
kubernetes
file_sd
discover targets
Prometheus server
Retrieval
TSDB
HTTP server
pull metrics
Node
HDD/SSD
Jobs/ exporters
Prometheus targets
Prometheus alerting
Alertmanager
pagerduty
Email
etc
notify
push alerts
PromQL
Prometheus web UI
Grafana
API clients
Data visualization and export

Service Discovery

Docs: Configuration

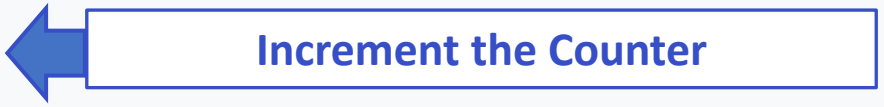
serviceMonitor/default/rancher-monitoring-apiserver/0 (3/3 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration
https://192.168.122.45:6443/metrics	UP	endpoint="https" instance="192.168.122.45:6443" job="apiserver" namespace="default" service="kubernetes"	38.8s ago	157.017ms
https://192.168.122.43:6443/metrics	UP	endpoint="https" instance="192.168.122.43:6443" job="apiserver" namespace="default" service="kubernetes"	12.493s ago	173.505ms
https://192.168.122.44:6443/metrics	UP	endpoint="https" instance="192.168.122.44:6443" job="apiserver" namespace="default" service="kubernetes"	42.726s ago	170.559ms

Instrumenting Software: Example

Docs: Client Libraries

```java
import io.prometheus.metrics.core.metrics.Counter;
import io.prometheus.metrics.exporter.httpserver.HTTPServer;
import io.prometheus.metrics.instrumentation.jvm.JvmMetrics;

import java.io.IOException;

public class App {

    public static void main(String[] args) throws InterruptedException, IOException {

        JvmMetrics.builder().register(); // initialize the out-of-the-box JVM metrics

        Counter counter = Counter.builder()
                .name("my_count_total")
                .help("example counter")
                .labelNames("status")
                .register();

        counter.labelValues("ok").inc();
        counter.labelValues("ok").inc();
        counter.labelValues("error").inc();

        HTTPServer server = HTTPServer.builder()
                .port(9400)
                .buildAndStart();

        System.out.println("HTTPServer listening on port http://localhost:" + server.getPort() + "/metrics");

        Thread.currentThread().join(); // sleep forever
    }
}
```



Exporters and Integrations

Docs: Exporters

- Exposing existing metrics from third-party systems
  - Databases
  - Hardware
  - Issue trackers and Continuous Integration
  - Messaging Systems
  - Storage
  - HTTP
  - APIs
  - Logging
  - Other monitoring systems
  - …

Querying: PromQL

Docs: Querying

- PromQL: Custom query language
- Recording rules (precompute frequently needed or computationally expensive expressions)
- What are other TSDBs doing?
- Standardization efforts: CNCF Observability Query Language Standard (QLS) workgroup

Prometheus Alerts Graph Status Help Classic UI

Use local time | Enable query history | Enable autocomplete | Use experimental editor | Enable highlighting | Enable linter

container_cpu_usage_seconds_total

Execute

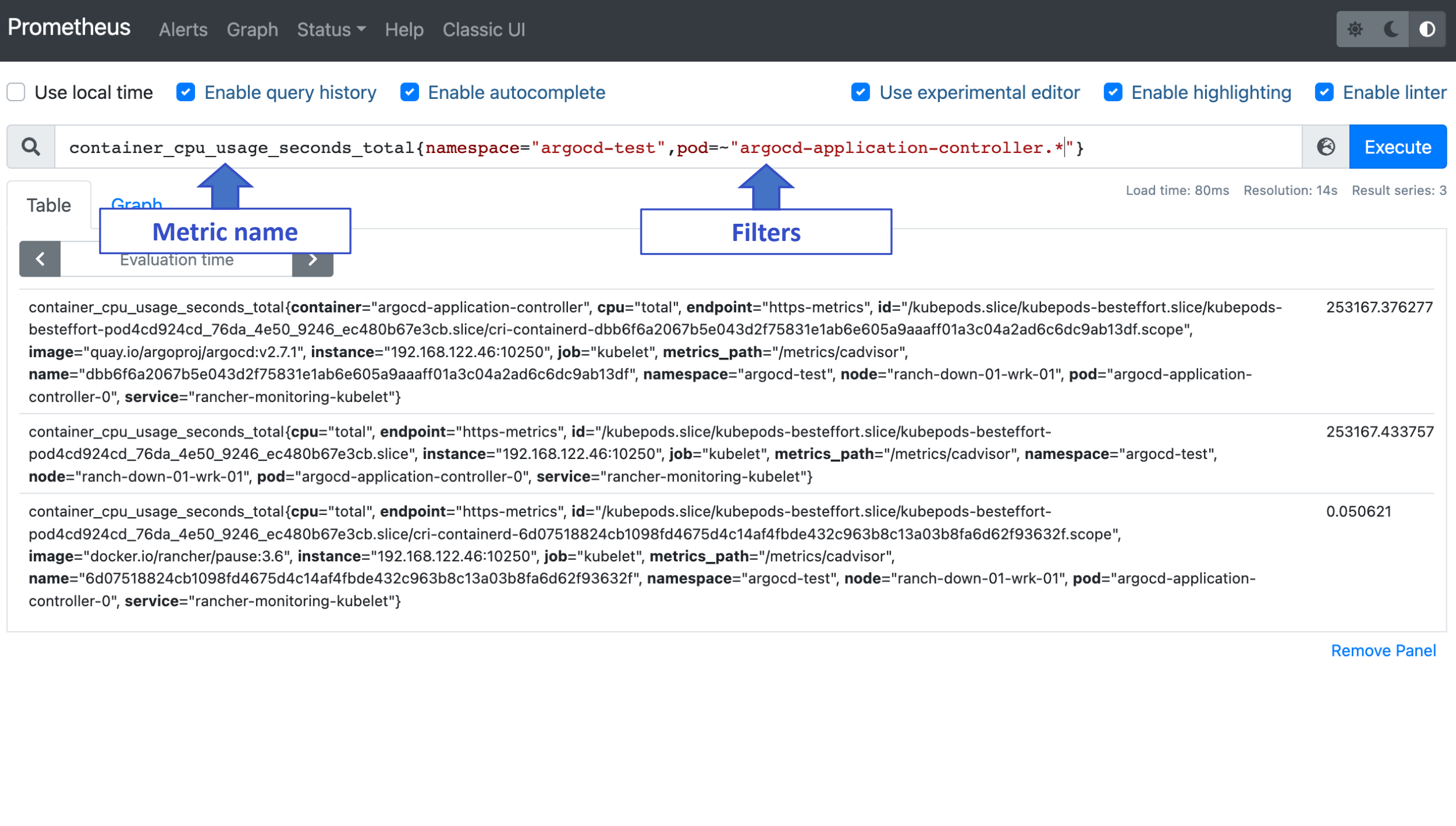
Load time: 279ms Resolution: 14s Result series: 305

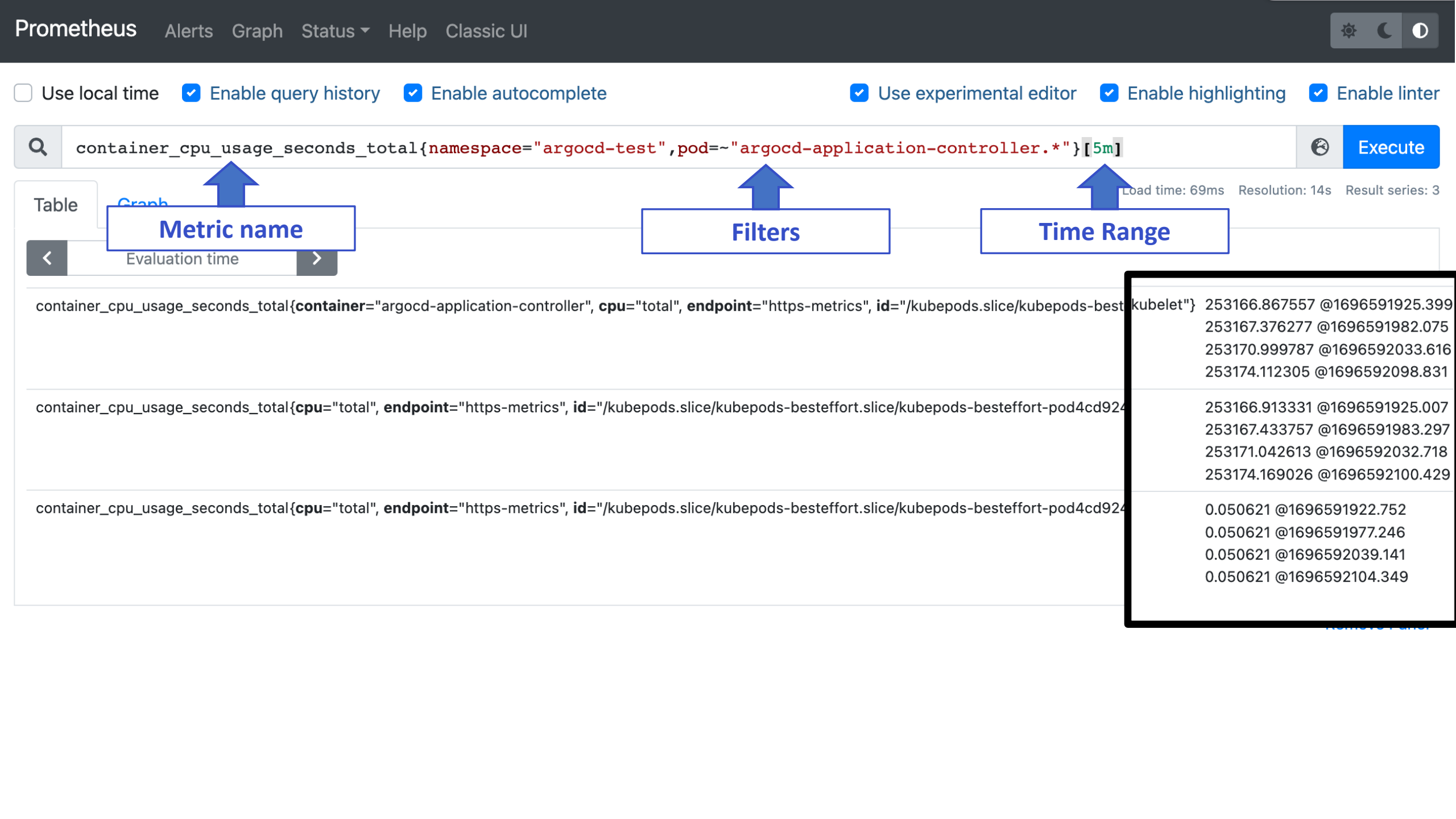
Table Graph

Metric name

Evaluation time

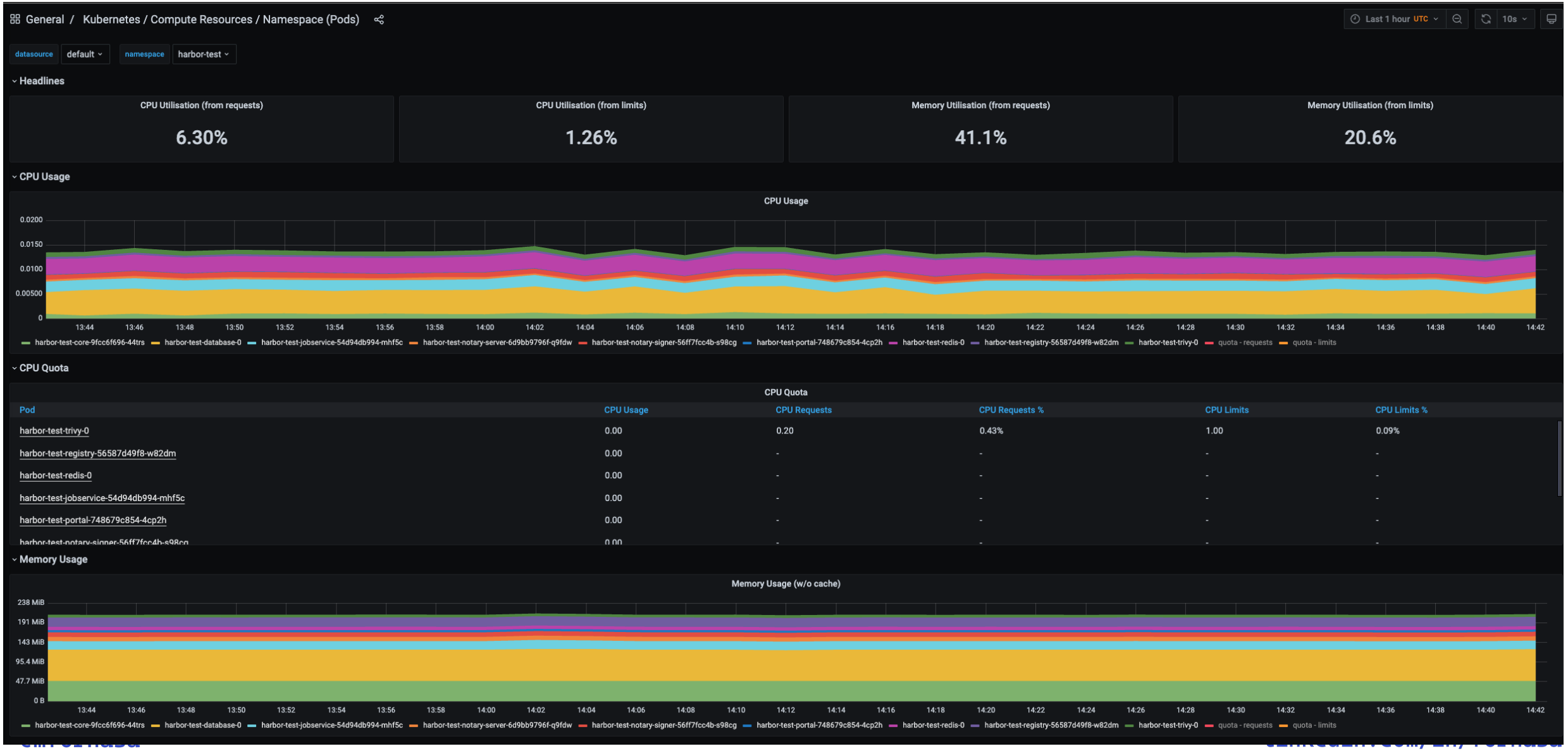
container_cpu_usage_seconds_total{**container**="argocd-application-controller", **cpu**="total", **endpoint**="https-metrics", **id**="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod4cd924cd_76da_4e50_9246_ec480b67e3cb.slice/cri-containerd-dbb6f6a2067b5e043d2f75831e1ab6e605a9aaaff01a3c04a2ad6c6dc9ab13df.scope", **image**="quay.io/argoproj/argocd:v2.7.1", **instance**="192.168.122.46:10250", **job**="kubelet", **metrics_path**="/metrics/cadvisor", **name**="dbb6f6a2067b5e043d2f75831e1ab6e605a9aaaff01a3c04a2ad6c6dc9ab13df", **namespace**="argocd-test", **node**="ranch-down-01-wrk-01", **pod**="argocd-application-controller-0", **service**="rancher-monitoring-kubelet"}	253162.607909
container_cpu_usage_seconds_total{**container**="argocd-applicationset-controller", **cpu**="total", **endpoint**="https-metrics", **id**="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod7523f3d3_f8fb_4b01_9e38_6caef986222a.slice/cri-containerd-fb041aad69a9f58cb734939caf2b552ec5929c11434739dca7875361f6e06041.scope", **image**="quay.io/argoproj/argocd:v2.7.1", **instance**="192.168.122.46:10250", **job**="kubelet", **metrics_path**="/metrics/cadvisor", **name**="fb041aad69a9f58cb734939caf2b552ec5929c11434739dca7875361f6e06041", **namespace**="argocd-test", **node**="ranch-down-01-wrk-01", **pod**="argocd-applicationset-controller-6c88cfbf55-v9sbk", **service**="rancher-monitoring-kubelet"}	4449.427233
container_cpu_usage_seconds_total{**container**="argocd-extensions", **cpu**="total", **endpoint**="https-metrics", **id**="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod2c9fe0a8_32d6_4b14_9370_70a462adcf6d.slice/cri-containerd-7b6c68510a9de221efd85a5249ec1ebecd05e77dc2ae62551e1ba228f000c2ac.scope", **image**="ghcr.io/argoproj-labs/argocd-extensions:latest", **instance**="192.168.122.46:10250", **job**="kubelet", **metrics_path**="/metrics/cadvisor", **name**="7b6c68510a9de221efd85a5249ec1ebecd05e77dc2ae62551e1ba228f000c2ac", **namespace**="argocd-test", **node**="ranch-down-01-wrk-01", **pod**="argocd-server-76b6f578f5-b49f4", **service**="rancher-monitoring-kubelet"}	3346.860682
container_cpu_usage_seconds_total{**container**="argocd-notifications-controller", **cpu**="total", **endpoint**="https-metrics", **id**="/kubepods.slice/kubepods-besteffort.slice/kubepods-	2539.750284


Prometheus
Alerts
Graph
Status
Help
Classic UI
Use local time
Enable query history
Enable autocomplete
Use experimental editor
Enable highlighting
Enable linter
container_cpu_usage_seconds_total{namespace="argocd-test",pod=~"argocd-application-controller.*"}
Execute
Table
Graph
Load time: 80ms
Resolution: 14s
Result series: 3
Metric name
Filters
Evaluation time
container_cpu_usage_seconds_total{container="argocd-application-controller", cpu="total", endpoint="https-metrics", id="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod4cd924cd_76da_4e50_9246_ec480b67e3cb.slice/cri-containerd-dbb6f6a2067b5e043d2f75831e1ab6e605a9aaaff01a3c04a2ad6c6dc9ab13df.scope", image="quay.io/argoproj/argocd:v2.7.1", instance="192.168.122.46:10250", job="kubelet", metrics_path="/metrics/cadvisor", name="dbb6f6a2067b5e043d2f75831e1ab6e605a9aaaff01a3c04a2ad6c6dc9ab13df", namespace="argocd-test", node="ranch-down-01-wrk-01", pod="argocd-application-controller-0", service="rancher-monitoring-kubelet"}
253167.376277
container_cpu_usage_seconds_total{cpu="total", endpoint="https-metrics", id="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod4cd924cd_76da_4e50_9246_ec480b67e3cb.slice", instance="192.168.122.46:10250", job="kubelet", metrics_path="/metrics/cadvisor", namespace="argocd-test", node="ranch-down-01-wrk-01", pod="argocd-application-controller-0", service="rancher-monitoring-kubelet"}
253167.433757
container_cpu_usage_seconds_total{cpu="total", endpoint="https-metrics", id="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod4cd924cd_76da_4e50_9246_ec480b67e3cb.slice/cri-containerd-6d07518824cb1098fd4675d4c14af4fbde432c963b8c13a03b8fa6d62f93632f.scope", image="docker.io/rancher/pause:3.6", instance="192.168.122.46:10250", job="kubelet", metrics_path="/metrics/cadvisor", name="6d07518824cb1098fd4675d4c14af4fbde432c963b8c13a03b8fa6d62f93632f", namespace="argocd-test", node="ranch-down-01-wrk-01", pod="argocd-application-controller-0", service="rancher-monitoring-kubelet"}
0.050621
Remove Panel


Prometheus
Alerts
Graph
Status
Help
Classic UI
Use local time
Enable query history
Enable autocomplete
Use experimental editor
Enable highlighting
Enable linter
container_cpu_usage_seconds_total{namespace="argocd-test",pod=~"argocd-application-controller.*"}[5m]
Execute
Load time: 69ms
Resolution: 14s
Result series: 3
Table
Graph
Metric name
Filters
Time Range
Evaluation time
container_cpu_usage_seconds_total{container="argocd-application-controller", cpu="total", endpoint="https-metrics", id="/kubepods.slice/kubepods-best kubelet"}
253166.867557 @1696591925.399
253167.376277 @1696591982.075
253170.999787 @1696592033.616
253174.112305 @1696592098.831
container_cpu_usage_seconds_total{cpu="total", endpoint="https-metrics", id="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod4cd924
253166.913331 @1696591925.007
253167.433757 @1696591983.297
253171.042613 @1696592032.718
253174.169026 @1696592100.429
container_cpu_usage_seconds_total{cpu="total", endpoint="https-metrics", id="/kubepods.slice/kubepods-besteffort.slice/kubepods-besteffort-pod4cd924
0.050621 @1696591922.752
0.050621 @1696591977.246
0.050621 @1696592039.141
0.050621 @1696592104.349

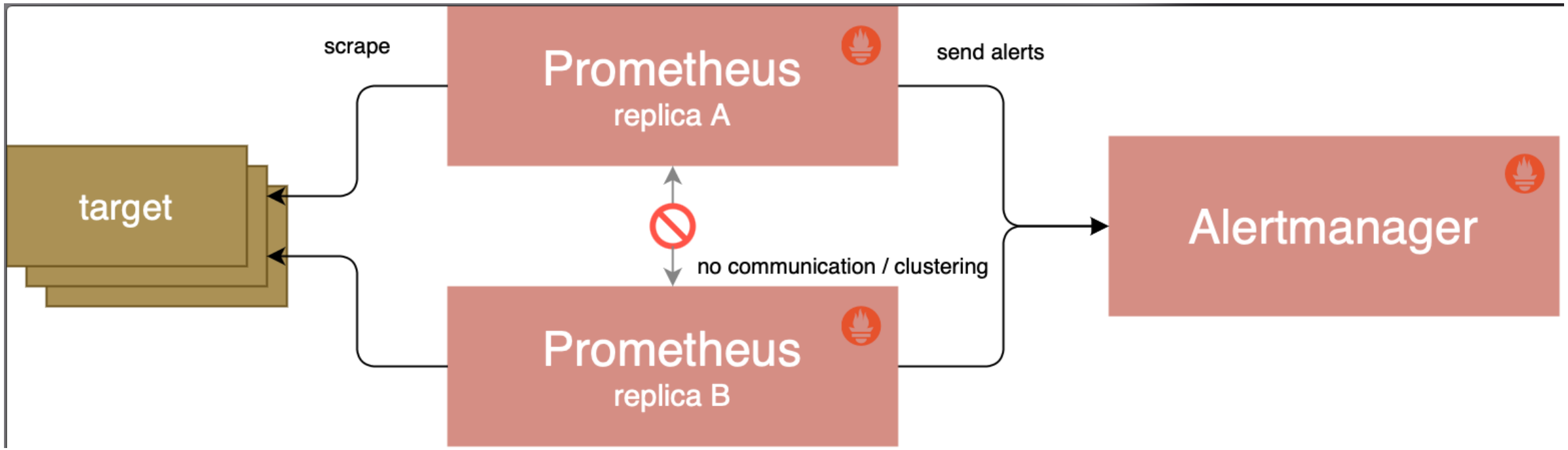
Visualization: Grafana

[Docs: Grafana Support for Prometheus]
[Grafana: Dashboard Collection]



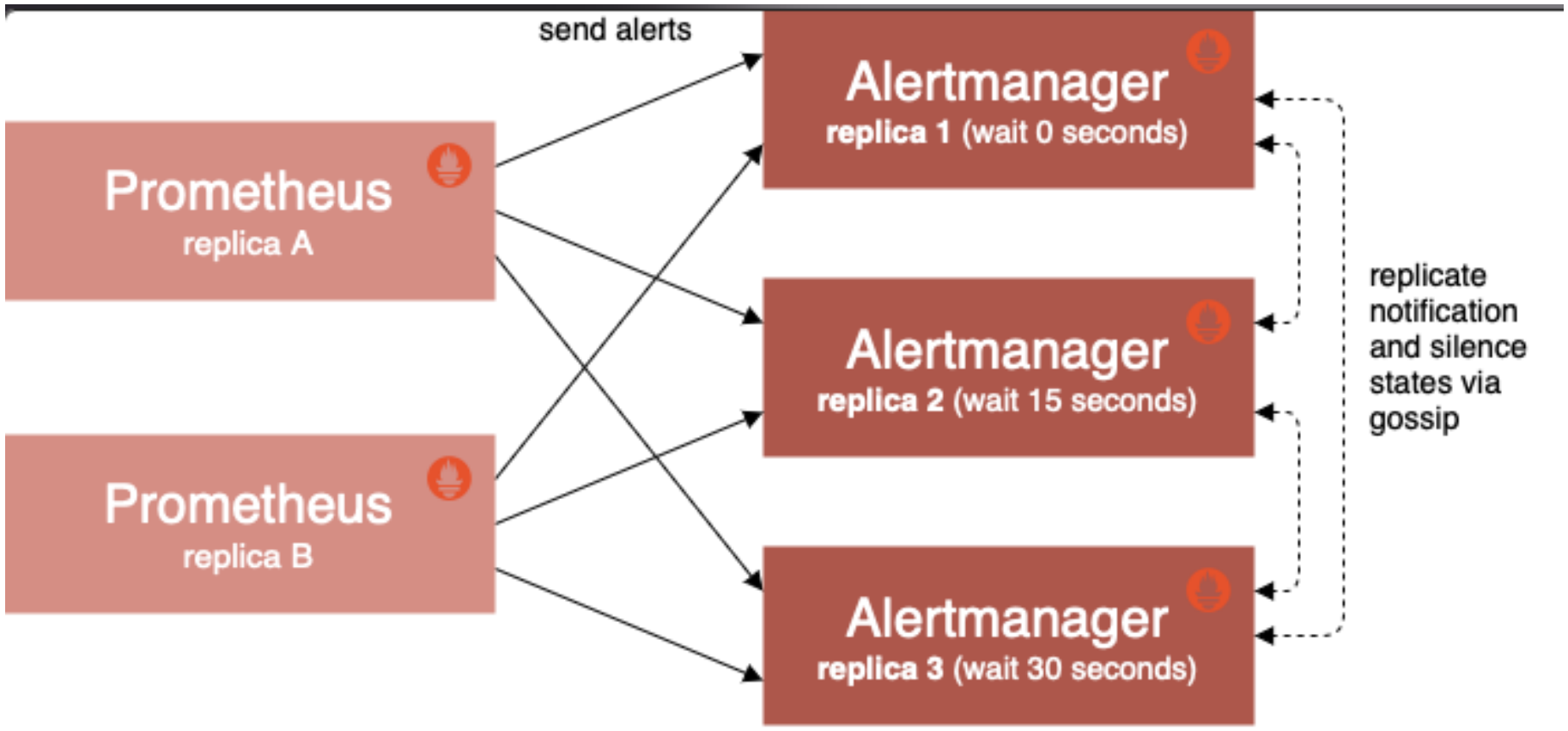
High Availability and Scalability

[Blog: High Availability]



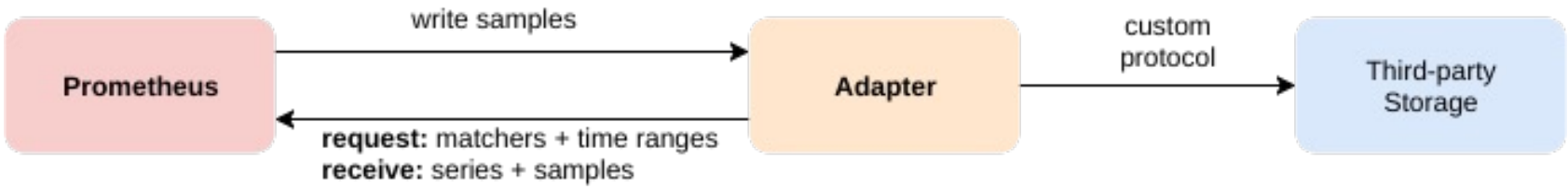
High Availability and Scalability

Blog: High Availability



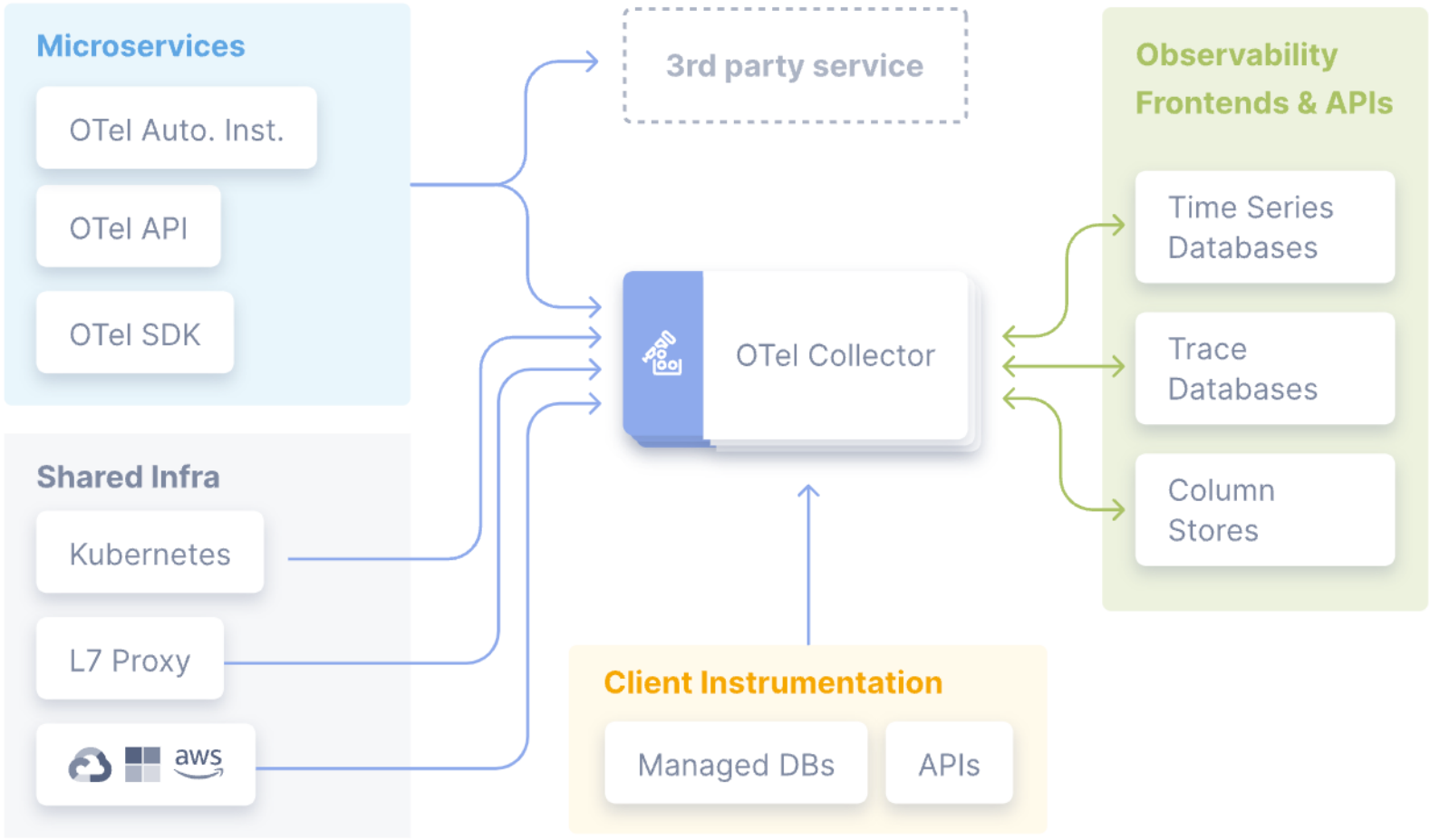
Long term storage e altri TSDB

- Non-goal of Prometheus:
  - Sophisticated scaling and clustering
  - Querying of multiple instances
  - Long-term storage
  - Downsampling and compaction
  - Support of high cardinality
- Projects
  - Thanos
  - Cortex
  - Mimir
  - InfluxDB / VictoriaMetrics / TimescaleDB



OpenTelemetry

[Website: OpenTelemetry]



OpenTelemetry

Website: OpenTelemetry
Website: Vendors

- Specification
- Standard protocol
- Semantic Conventions
- Library Ecosystem
- Automatic Instrumentation
- Language SDKs
- Collector (proxy)
- Tooling

- Querying
- Storage
- Visualization

Prometheus as an OpenTelemetry backend

Grafana Blog Post
Talk at PromCon EU 2023

- Prometheus can be a OTel backend using the *OTel Prometheus Exporter*. Some open issues around that:
  - Metric conversion
  - Mapping of resource attributes
  - Out of order writes
- Preliminary activities for a native support, including the push protocol Open points:
  - Build the *up* metric for push model
  - Convert between data models (resource attributed, charset)
  - Support out of order (currently disabled by default)
  - Documentation
  - …

Questions?

joind.in/talk/16f8c

Watch - Prometheus: The Documentary

Alerting - Rules

Docs: Alerting

Alerting Rules in **Prometheus**

alert: KubeAPIDown **expr:** absent(up{job="apiserver"} == 1) **for:** 15m **labels:** severity: critical **annotations:** description: KubeAPI has disappeared from Prometheus target discovery. runbook_url: https://github.com/kubernetes-monitoring/kubernetes-mixin/tree/master/runbook.md#alert-name-kubeapidown summary: Target disappeared from Prometheus target discovery.	OK		43.666s ago	0.175ms

Alerting - Managing

Docs: Alertmanager

Alertmanager (separate component)

- Notifications: Email, Chat. Waiting times, repeat intervals.
- Grouping: grouping alerts of similar nature
- Inhibitions: mute a set of alerts given that another alert is firing
- Silencing: from the web interface

Observability

A possible definition

Observability lets us understand a system from the outside, by letting us ask questions about that system without knowing its inner workings. Furthermore, it allows us to easily troubleshoot and handle novel problems (i.e. “unknown unknowns”), and helps us answer the question, “Why is this happening?”

In order to be able to ask those questions of a system, the application must be properly instrumented. That is, the application code must emit signals such as traces, metrics, and logs. An application is properly instrumented when developers don’t need to add more instrumentation to troubleshoot an issue, because they have all of the information they need.

A different one

Can you understand what is happening inside the system — can you understand ANY internal state the system may get itself into, simply by asking questions from the outside?

This is such a reliable bait and switch that any time you hear someone talking about “metrics, logs and traces”, you can be pretty damn sure there’s no actual observability going on.