



IS SOMETHING ON FIRE?

End-to-end monitoring of infrastructure, systems, and applications using Prometheus

presented by:
Gabriel Tocci

SIG Introduction



Industry Commitment

SIG was established in 1987 serving higher education with IT initiatives that enhance services for students, faculty, staff and alumni.



Ellucian® Partner

SIG is one of the largest privately held higher education consulting firms in the U.S., and we have been a long-standing partner with Ellucian for many years.



Consulting Continuity

Powered by more than 100 professionals, with an average tenure of 9.5 years, SIG can provide consulting continuity to keep your goals on task.



Agility to Respond

SIG provides a full lifecycle of services from strategy through managed services, and our ability to move quickly is what we do best. We can respond to uncertainty with flexibility as new opportunities arise.

More than
100
People



More than
35
Years Serving
Higher Education

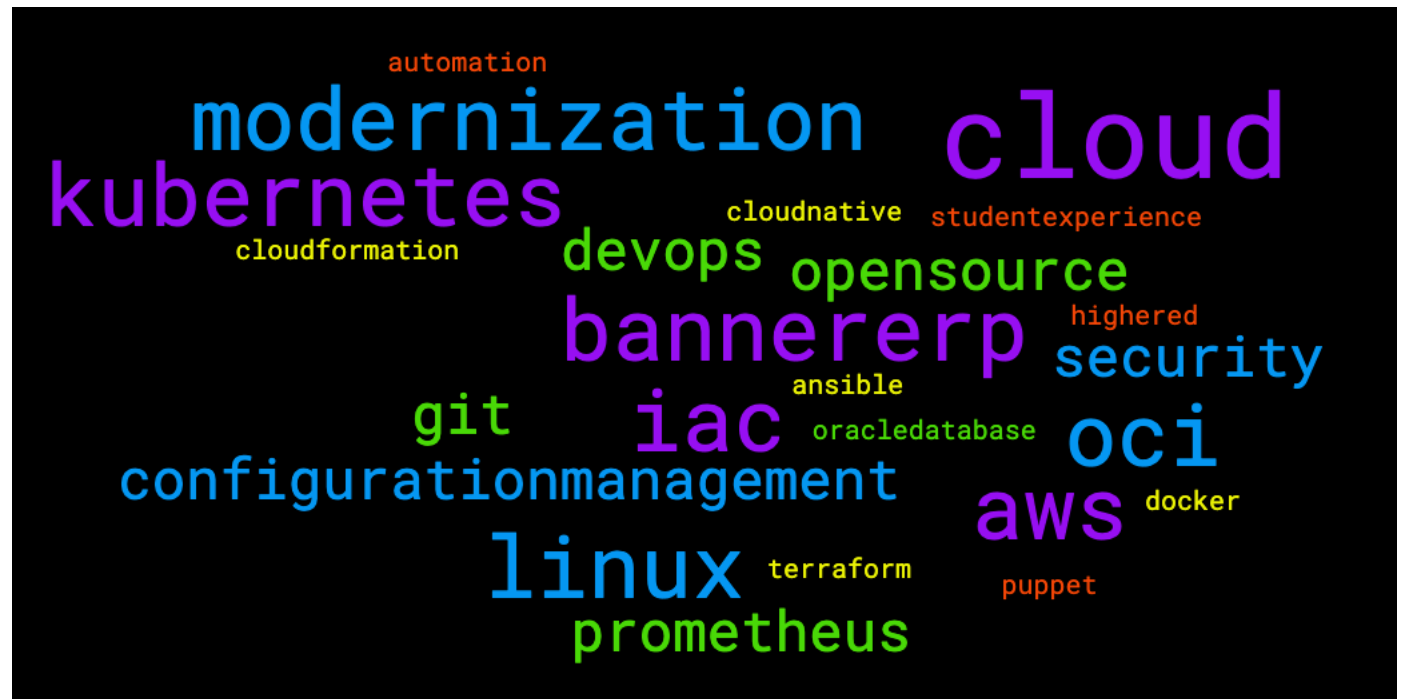
Presenter



Gabriel Tocci

Senior Certified Cloud Architect

tocci@sigcorp.com



Prometheus Stack Overview

Metrics-based
monitoring, alerting,
and **visualization** with:

- Prometheus
- Alert Manager
- Grafana



Prometheus



Alertmanager



Grafana

Components

Prometheus

- SoundCloud tool. Second “Cloud Native Computing Foundation” graduated project, after Kubernetes
- Uses agents (exporters) to expose metrics for Prometheus
- Prometheus collects (scrapes) metrics from exporters and stores data for use
- Ad-hoc “search” / filter and graph metrics collected by metric names and tags for analysis.
 - PromQL: Query data similarly to Elasticsearch/Lucene, Kibana, Splunk with simplified language.

Alert Manager

- Add-on component for Prometheus to allow Prometheus metrics to dispatch alerts.
- Supports various notification targets: email, PagerDuty, Slack, etc.

Grafana

- Analytics and visualization system
- Build dashboards from a myriad of data sources
 - (including Prometheus, Cloudwatch, RDBMS, etc.)



Exporters

- Node (host)
- MS SQL
- MySql
- Oracle
- Postgres
- Redis
- ProcessExporter
- HTTP Endpoints
- RabbitMQ
- Kafka
- BIG-IP / F5
- Apache
- HAProxy
- Nginx
- Varnish
- AWS
- Azure
- Cloudflare
- Digital Ocean
- Rancher
- Github
- Jira

<https://prometheus.io/docs/instrumenting/exporters/>

Software with Exposed Prometheus Metrics

- Ansible Tower
- HAProxy
- Traefik
- Docker Daemon
- Gitlab
- Kubernetes
- CockroachDB
- App Connect Enterprise
- Deph
- FreeBSD Kernel
- Linkerd
- Netdata
- OpenZiti
- Pretix
- Vector

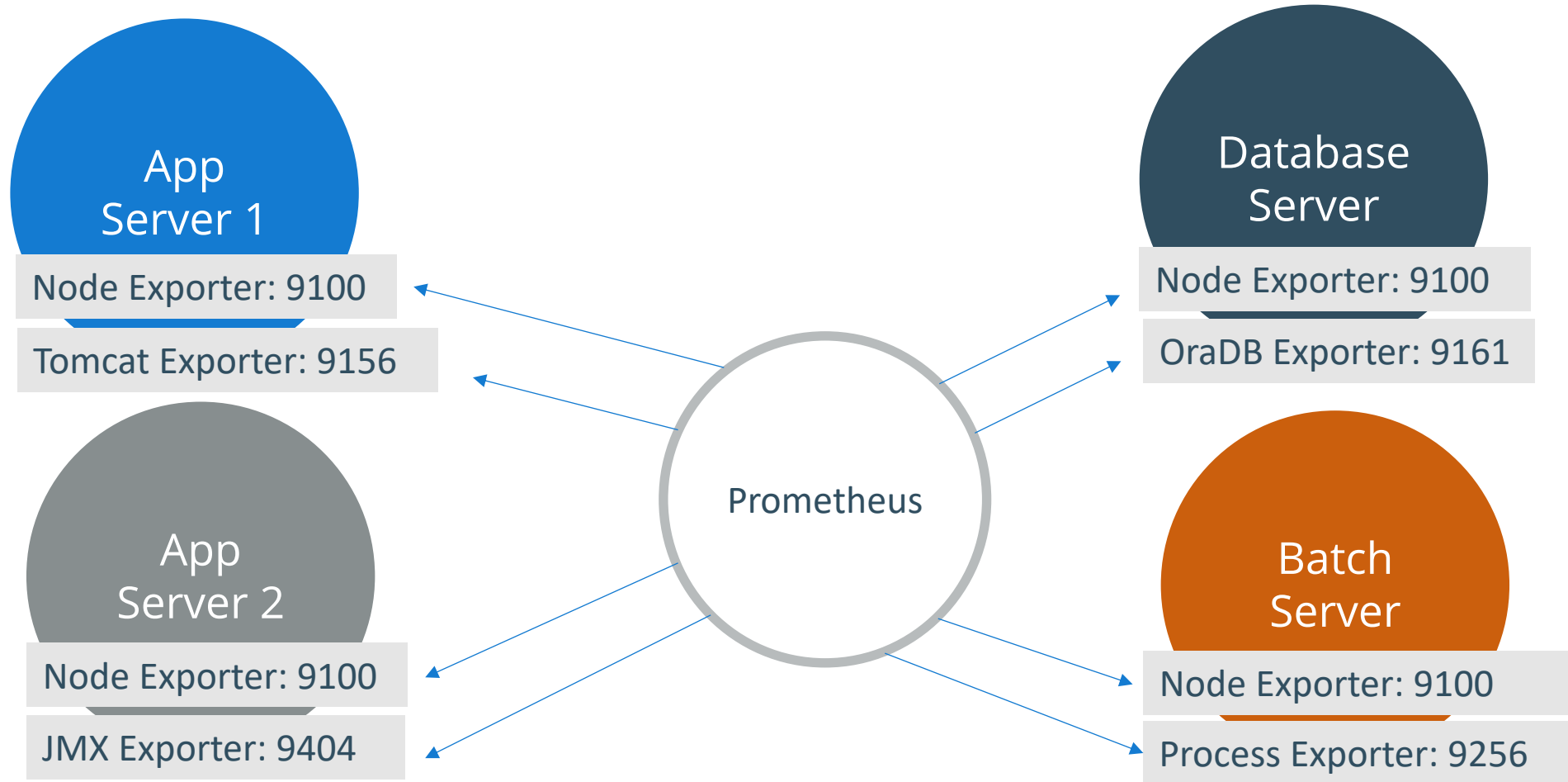
<https://prometheus.io/docs/instrumenting/exporters/>

\$ curl localhost:9100/metrics

```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.2853e-05
go_gc_duration_seconds{quantile="0.25"} 4.8939e-05
go_gc_duration_seconds{quantile="0.5"} 6.6122e-05
go_gc_duration_seconds{quantile="0.75"} 7.6978e-05
go_gc_duration_seconds{quantile="1"} 0.002877178
go_gc_duration_seconds_sum 8.273970152
go_gc_duration_seconds_count 76038
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.15.8"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.98992e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.49724037016e+11
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.979467e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1.619611962e+09
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 5.782905439544153e-05
```


Prometheus

Prometheus Architecture



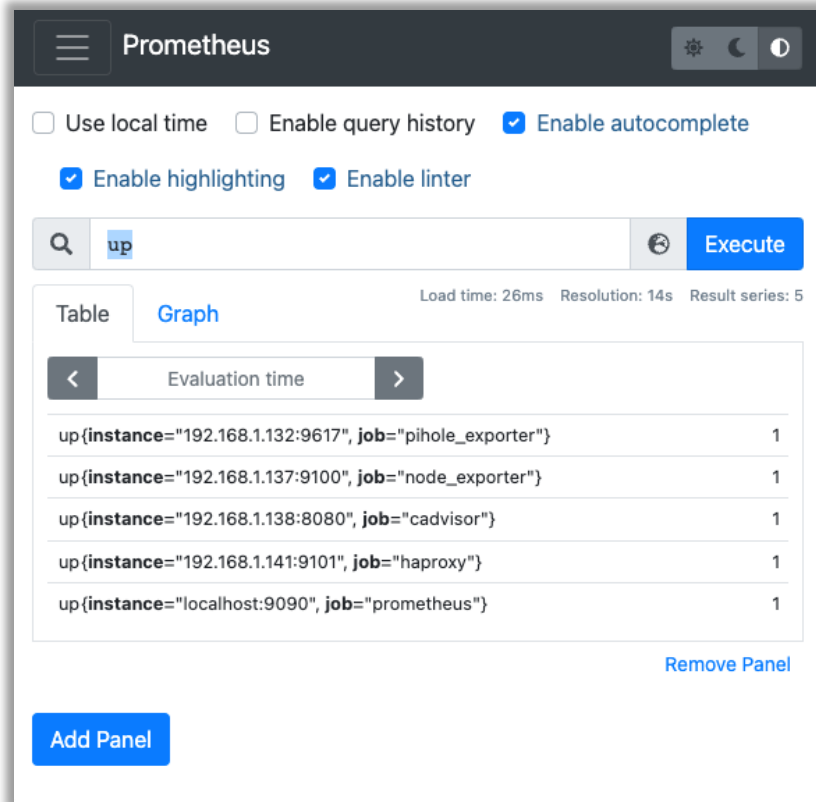
<https://github.com/prometheus/prometheus/wiki/Default-port-allocations>

Prometheus: Scrape Configuration

prometheus.yml 919 Bytes

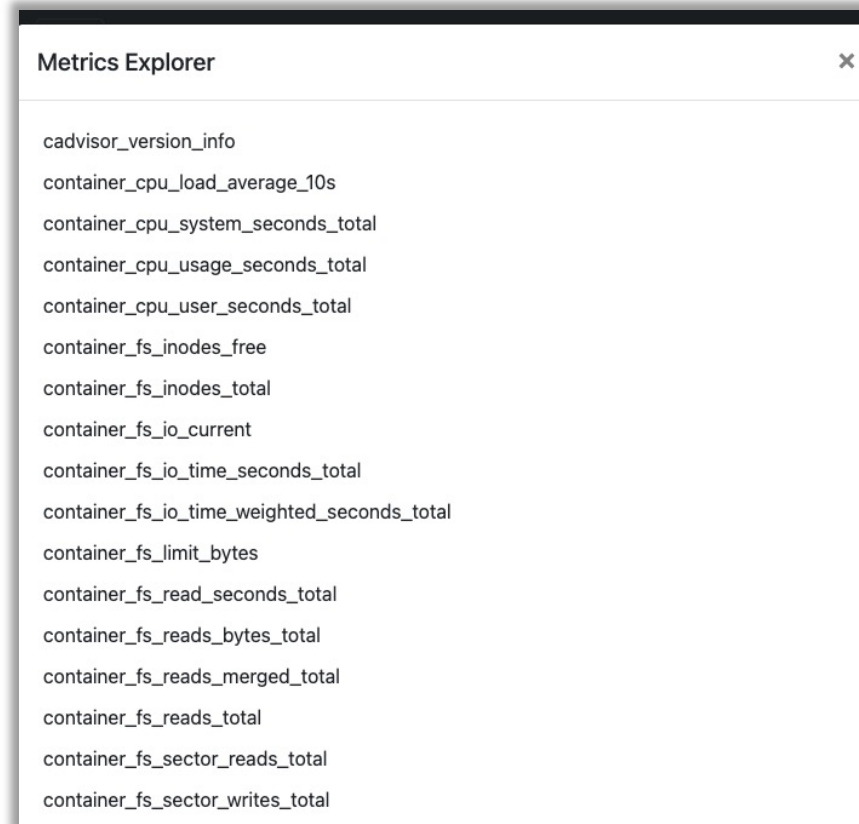
```
1 # my global config
2 global:
3   scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
4   evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
5
6 alerting:
7   alertmanagers:
8     - static_configs:
9       - targets:
10         - alertmanager:9093
11
12 rule_files:
13   - "primary_alert_rules.yml"
14
15 scrape_configs:
16   - job_name: "prometheus"
17     static_configs:
18       - targets: ["localhost:9090"]
19   - job_name: "pihole_exporter"
20     static_configs:
21       - targets: ["192.168.1.132:9617"]
22   - job_name: "node_exporter"
23     static_configs:
24       - targets: ["192.168.1.137:9100", "192.168.1.138:9100", "192.168.1.139:9100"]
25   - job_name: "cadvisor"
26     static_configs:
27       - targets: ["192.168.1.138:8080", "192.168.1.139:8080", "192.168.1.142:8080"]
28   - job_name: "haproxy"
29     static_configs:
30       - targets: ["192.168.1.141:9101", "192.168.1.144:9101"]
31
32
```

Prometheus: PromQL Ad-Hoc Query



The screenshot shows the Prometheus web interface. At the top, there are settings for 'Use local time', 'Enable query history', 'Enable autocomplete', 'Enable highlighting', and 'Enable linter'. A search bar contains the query 'up' and an 'Execute' button. Below the search bar, there are tabs for 'Table' and 'Graph'. The 'Table' view shows a table with 5 rows of results. The table has a header 'Evaluation time' and a 'Remove Panel' button at the bottom right. An 'Add Panel' button is located at the bottom left.

Evaluation time	
<code>up{instance="192.168.1.132:9617", job="pihole_exporter"}</code>	1
<code>up{instance="192.168.1.137:9100", job="node_exporter"}</code>	1
<code>up{instance="192.168.1.138:8080", job="cadvisor"}</code>	1
<code>up{instance="192.168.1.141:9101", job="haproxy"}</code>	1
<code>up{instance="localhost:9090", job="prometheus"}</code>	1



The screenshot shows the Prometheus Metrics Explorer interface. It displays a list of metrics, including:

- cadvisor_version_info
- container_cpu_load_average_10s
- container_cpu_system_seconds_total
- container_cpu_usage_seconds_total
- container_cpu_user_seconds_total
- container_fs_inodes_free
- container_fs_inodes_total
- container_fs_io_current
- container_fs_io_time_seconds_total
- container_fs_io_time_weighted_seconds_total
- container_fs_limit_bytes
- container_fs_read_seconds_total
- container_fs_reads_bytes_total
- container_fs_reads_merged_total
- container_fs_reads_total
- container_fs_sector_reads_total
- container_fs_sector_writes_total

Prometheus: Filtering / Graphing

The screenshot shows the Prometheus web interface. At the top, there are navigation icons and the word "Prometheus". Below that, there are several checkboxes: "Use local time" (unchecked), "Enable query history" (unchecked), "Enable autocomplete" (checked), "Enable highlighting" (checked), and "Enable linter" (checked). A search bar contains the query `up{instance="192.168.1.138:8080"}` and an "Execute" button. Below the search bar, there are tabs for "Table" and "Graph". The "Table" tab is active, showing a table with one row: `up(instance="192.168.1.138:8080", job="cadvisor")` with a value of 1. There are also "Add Panel" and "Remove Panel" buttons.

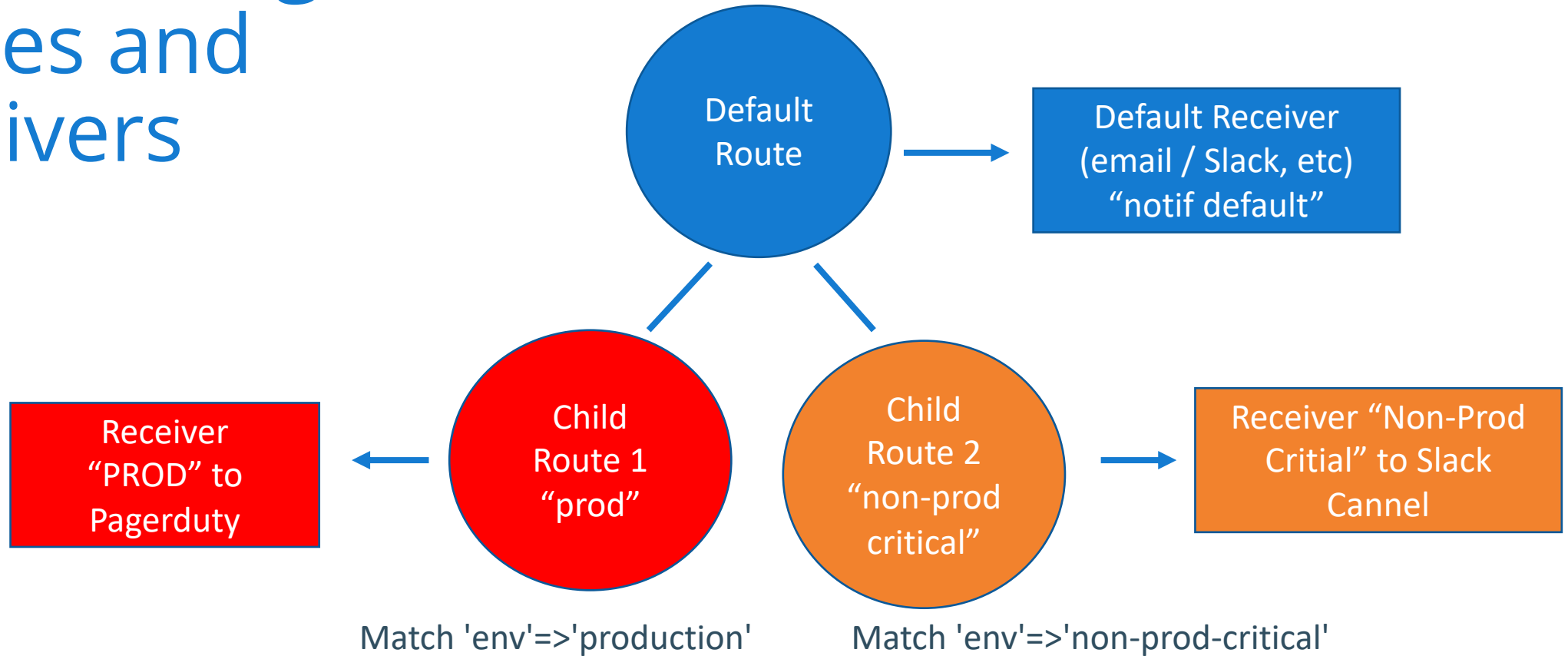
The screenshot shows the Prometheus web interface with the "Graph" tab selected. The search bar contains the query `node_load1{instance="192.168.1.137:9100", }` and an "Execute" button. Below the search bar, there are tabs for "Table" and "Graph". The "Graph" tab is active, showing a line graph of `node_load1` over time. The y-axis ranges from 0.40 to 1.80, and the x-axis shows time from 15:30 to 16:20. The graph shows a fluctuating yellow line representing the load. There are also "Show Exemplars" and "Res. (s)" buttons.

Alert Manager

Alert Rules

- alert: PRODInstanceDown
 expr: 'up == 0'
 for: 1m
 labels:
 severity: **critical**
 annotations:
 description: '{{ \$labels.instance }} of job {{ \$labels.job }} has been down for 5 minutes.'
 summary: 'Instance {{ \$labels.instance }} down'
- alert: HTTPERRORDetected
 annotations:
 message: 'Probe on {{ \$labels.instance }} is throwing an HTTPS 5xx or 4xx Error.'
 summary: 'HTTP ERROR Detected.'
 expr: '**probe_http_status_code > 399**'
 for: 1m
 labels:
 severity: **critical**

Alert Manager: Routes and Receivers



It then traverses the child nodes. If `continue` is set to `false`, it stops after the first matching child. If `continue` is `true` on a matching node, the alert will continue matching against subsequent siblings.

Alert Manager: Receivers

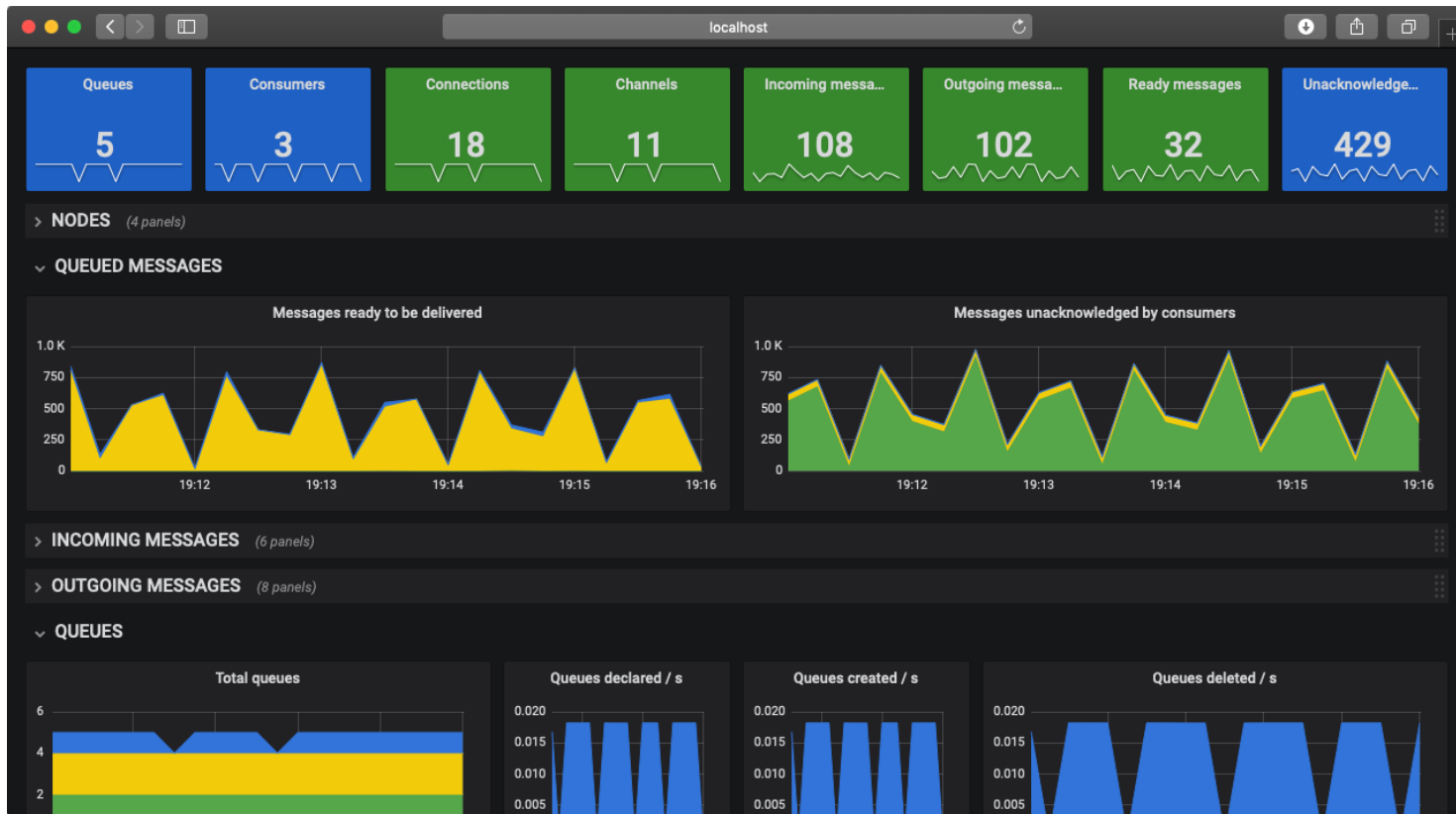
- Email
- PagerDuty
- Slack
- MS Teams
- WeChat
- Telegram
- Pushover
- Opsgenie
- VictorOps / Splunk On-Call
- Custom Webhook
- More...

Alert Manager: Example Routes

```
# The root route with all parameters, which are inherited by the child
# routes if they are not overwritten.
route:
  receiver: 'default-receiver'
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 4h
  group_by: [cluster, alertname]
# All alerts that do not match the following child routes
# will remain at the root node and be dispatched to 'default-receiver'.
routes:
# All alerts with service=mysql or service=cassandra
# are dispatched to the database pager.
- receiver: 'database-pager'
  group_wait: 10s
  matchers:
    - service=~"mysql|cassandra"
# All alerts with the team=frontend label match this sub-route.
# They are grouped by product and environment rather than cluster
# and alertname.
- receiver: 'frontend-pager'
  group_by: [product, environment]
  matchers:
    - team="frontend"
```

Grafana

Grafana Dashboards



RabbitMQ Native Metrics and Dashboard

<https://rabbitmq.com/prometheus.html>

Grafana: Community Dashboards



Postfix

★ 4/5 5 ratings

1.72K downloads

Prometheus



NodeJS Application Dashboard

★ 4.6/5 5 ratings

23.3K downloads

Prometheus



PostgreSQL Database


★ 4.2/5 5 ratings

66.6K downloads

Prometheus

<https://grafana.com/grafana/dashboards/>

Grafana: Community Dashboards



Node Exporter Full by rfrail3


DASHBOARD
Last updated: 7 months ago

Start with Grafana Cloud and the new FREE tier. Includes 10K series Prometheus or Graphite Metrics and 50gb Loki Logs

Downloads: 1
Reviews: 3

[Add your review](#)

Overview | Revisions | Reviews



Nearly all default values exported by Prometheus node exporter graphed.

Only requires the default job_name: node, add as many targets as you need in '/etc/prometheus/prometheus.yml'.

```
- job_name: node
  static_configs:
    - targets: ['localhost:9100']
```

Recommended for prometheus-node-exporter the arguments '--collector.systemd --collector.processes' because the graph uses some of their metrics.

Since revision 16, for prometheus-node-exporter v0.18 or newer. Since revision 12, for prometheus-node-exporter v0.16 or newer.

Available on github: <https://github.com/rfrail3/grafana-dashboards.git>

Get this dashboard:

1860

[Copy ID to Clipboard](#)

[Download JSON](#)
[How do I import this dashboard?](#)

Dependencies:

- GRAFANA 7.3.7
- GAUGE
- GRAPH
- PROMETHEUS 1.0.0
- SINGLESTAT

Import

Import dashboard from file or Grafana.com

[Upload JSON file](#)

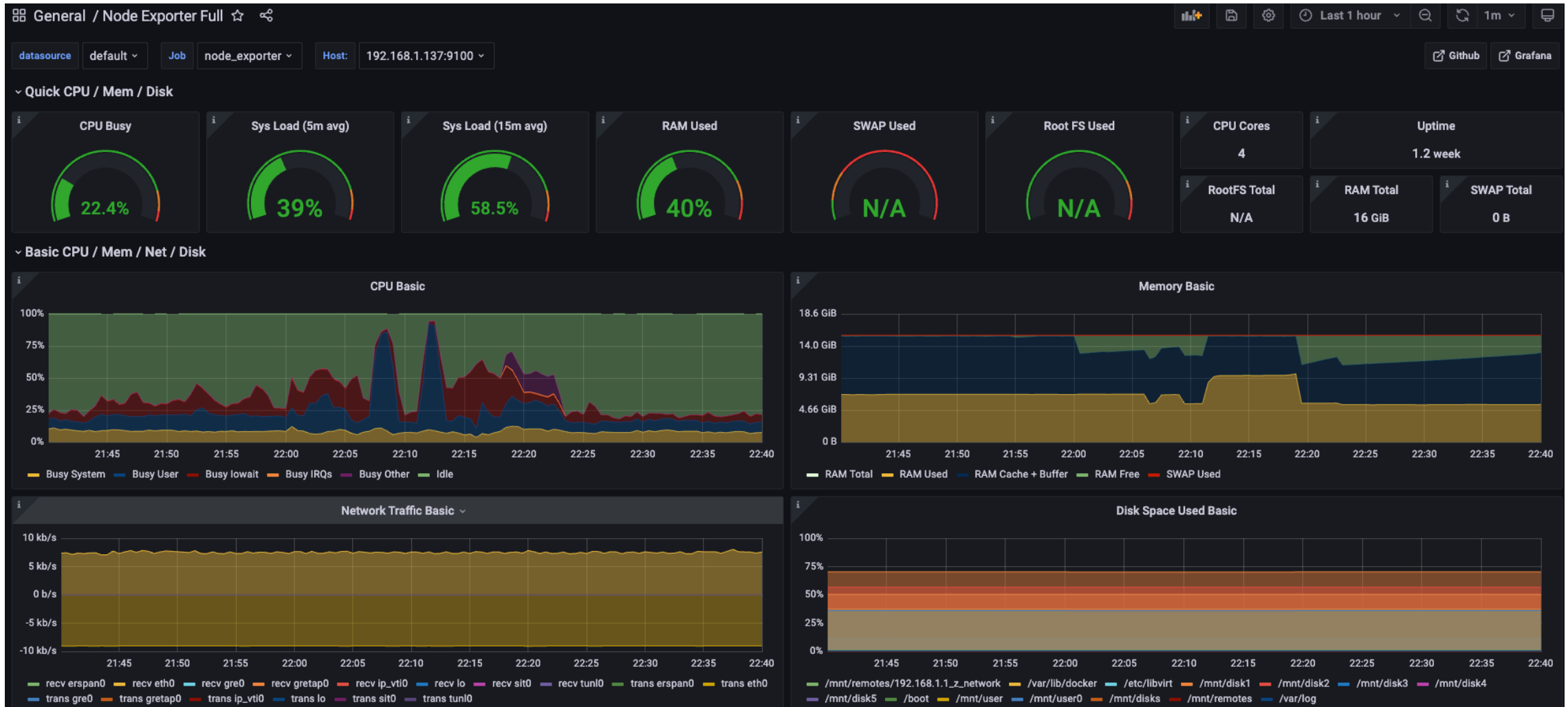
Import via grafana.com

1860 [Load](#)

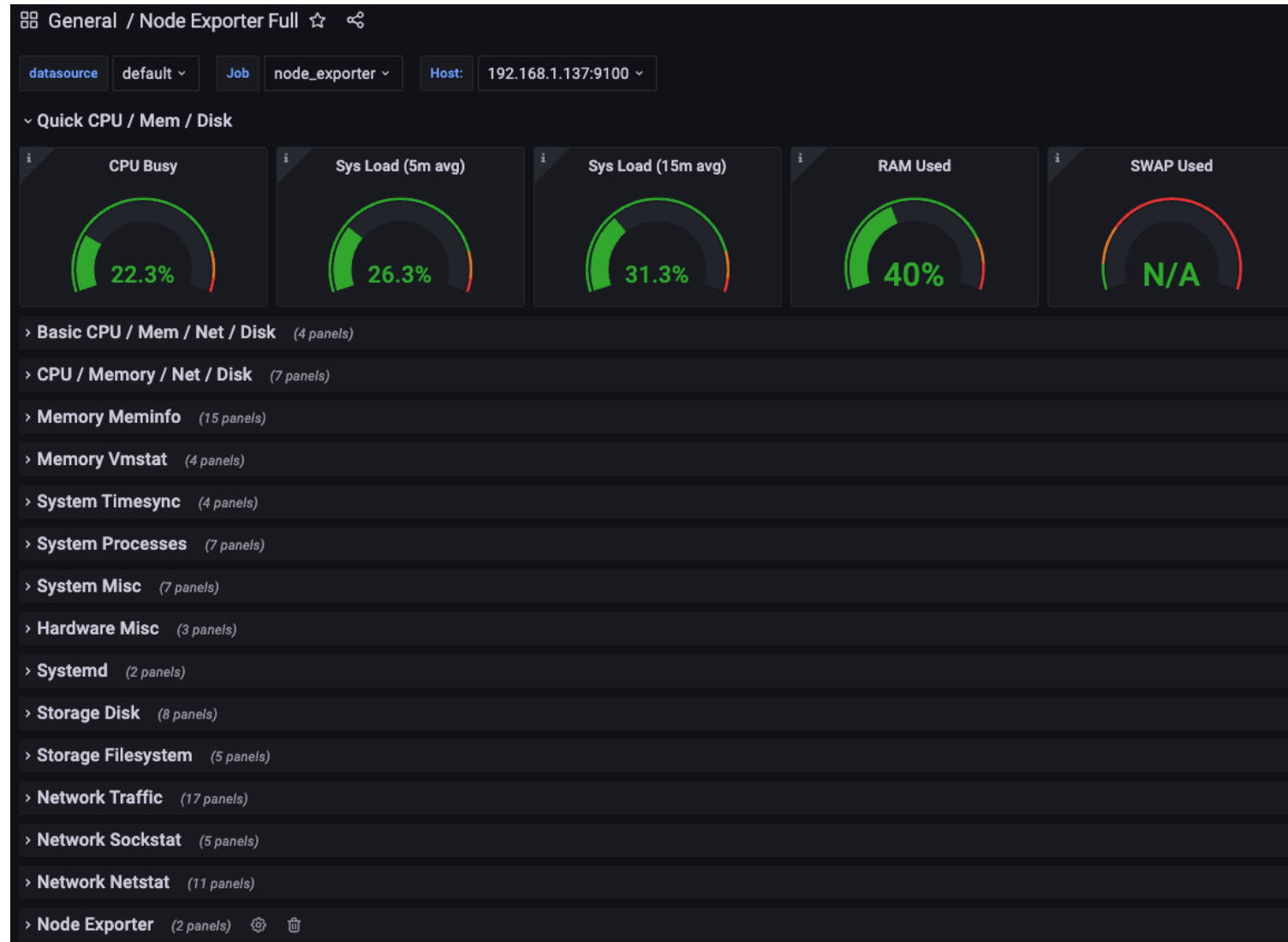
Import via panel json

[Load](#)

Node Exporter: Server Stats



Node Exporter: More Server Stats



Blackbox-Exporter

Status

UP

HTTP Status Code

200

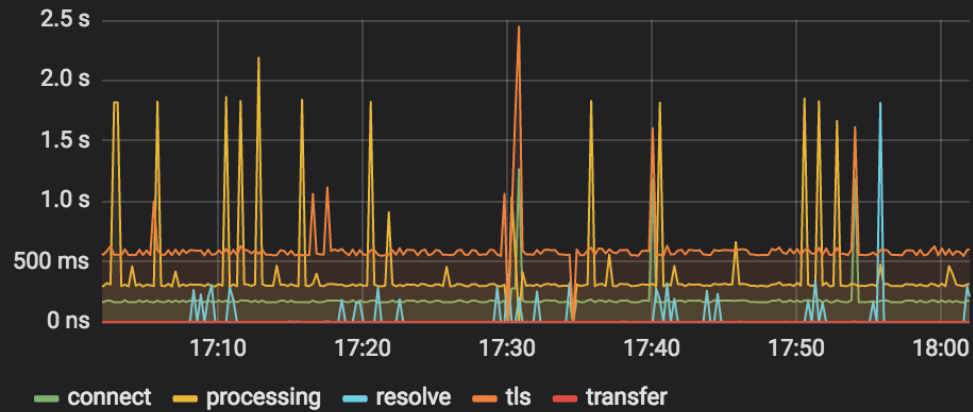
HTTP Version

1.1

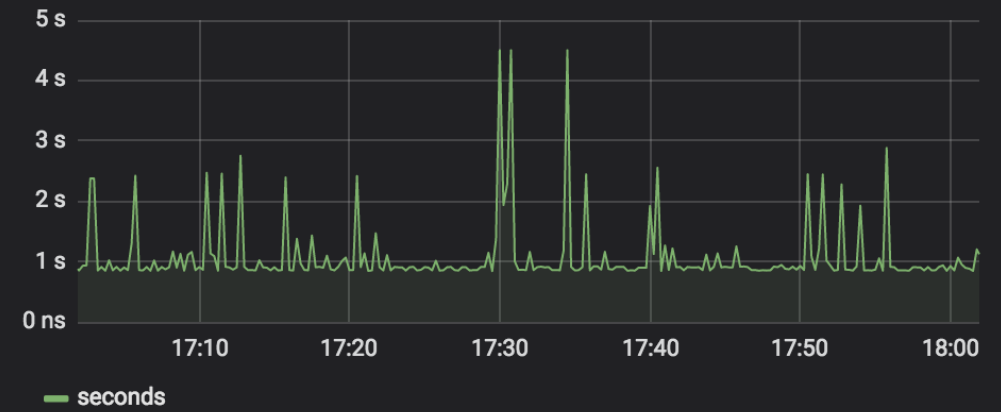
SSL

YES

HTTP Duration



Probe Duration



SSL Expiry

1 year, 9 months, 2 weeks

Average Probe Duration

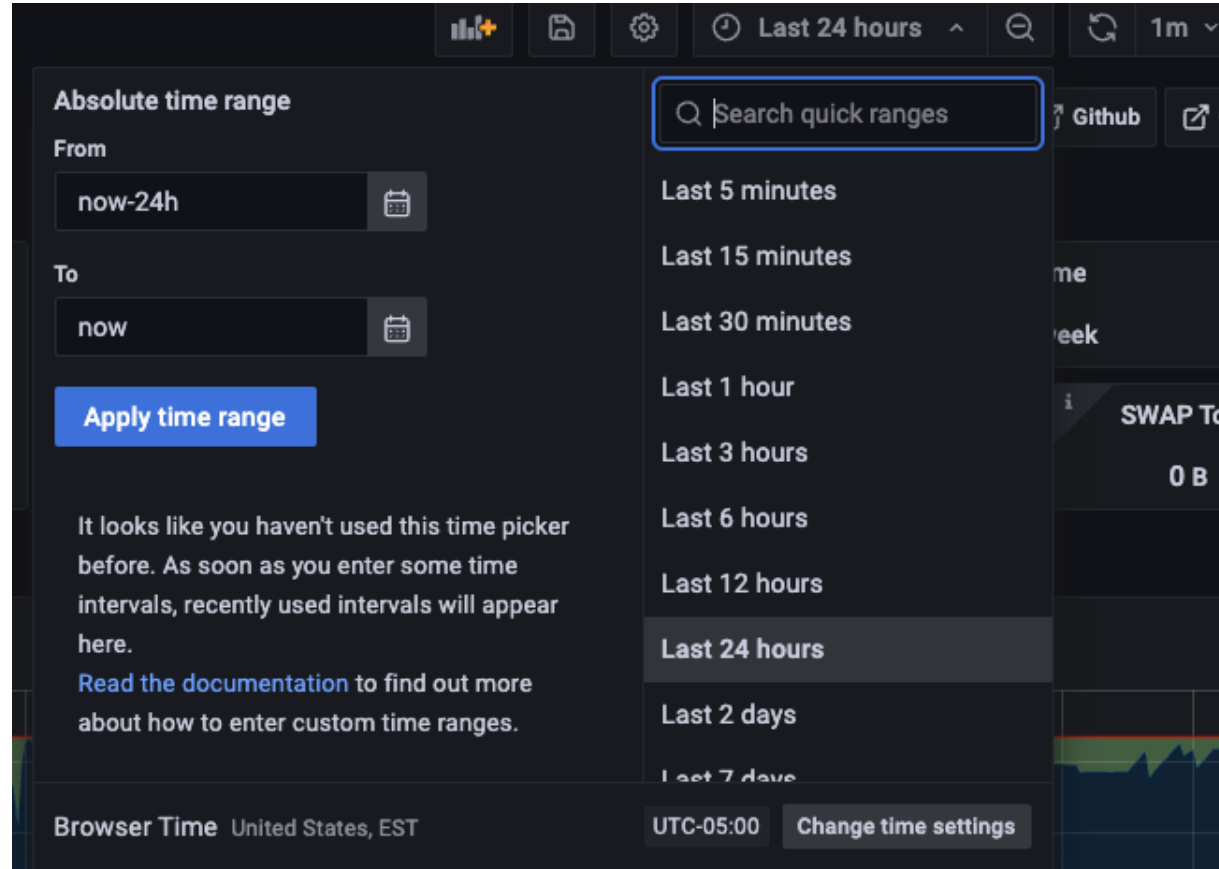
1.1 s

Average DNS Lookup

168 ms

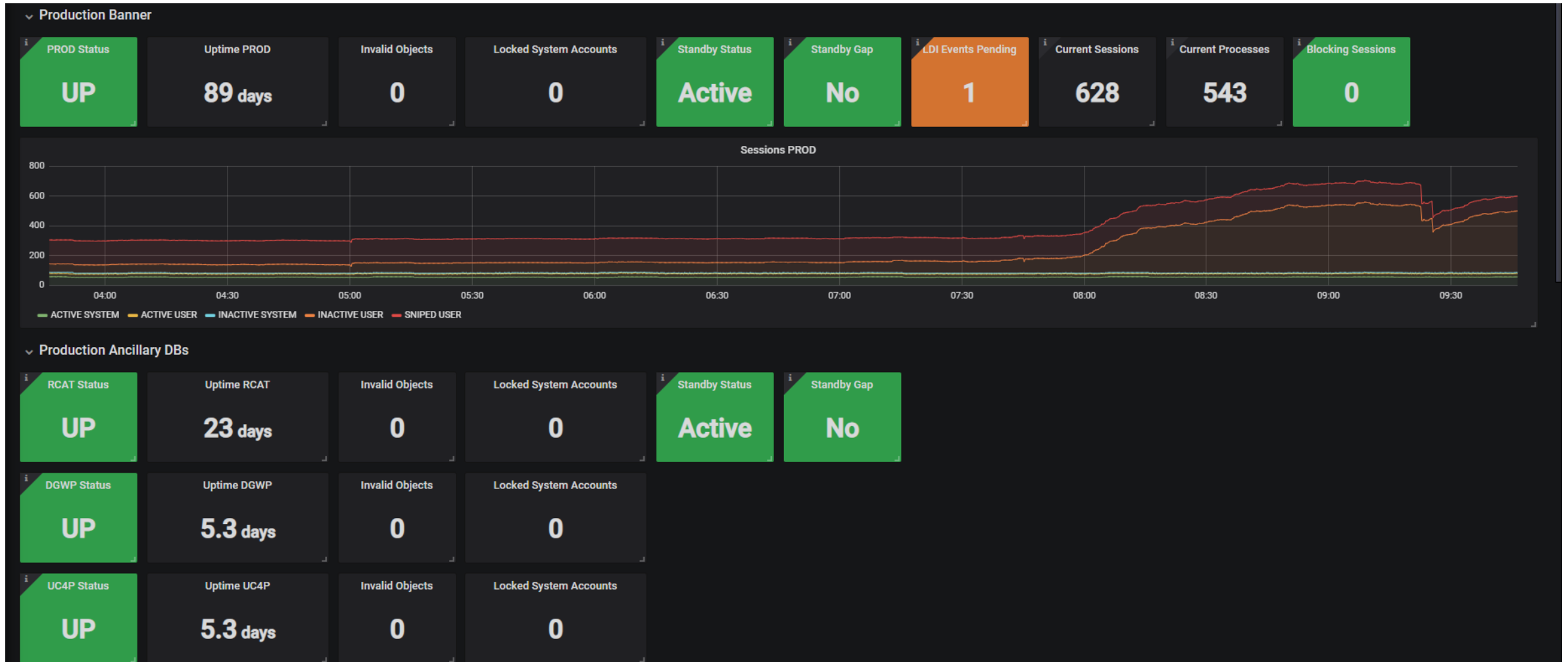
<https://erp.school.edu/SelfServiceApplication/actuator/health>

Grafana: Filtering / Time Series



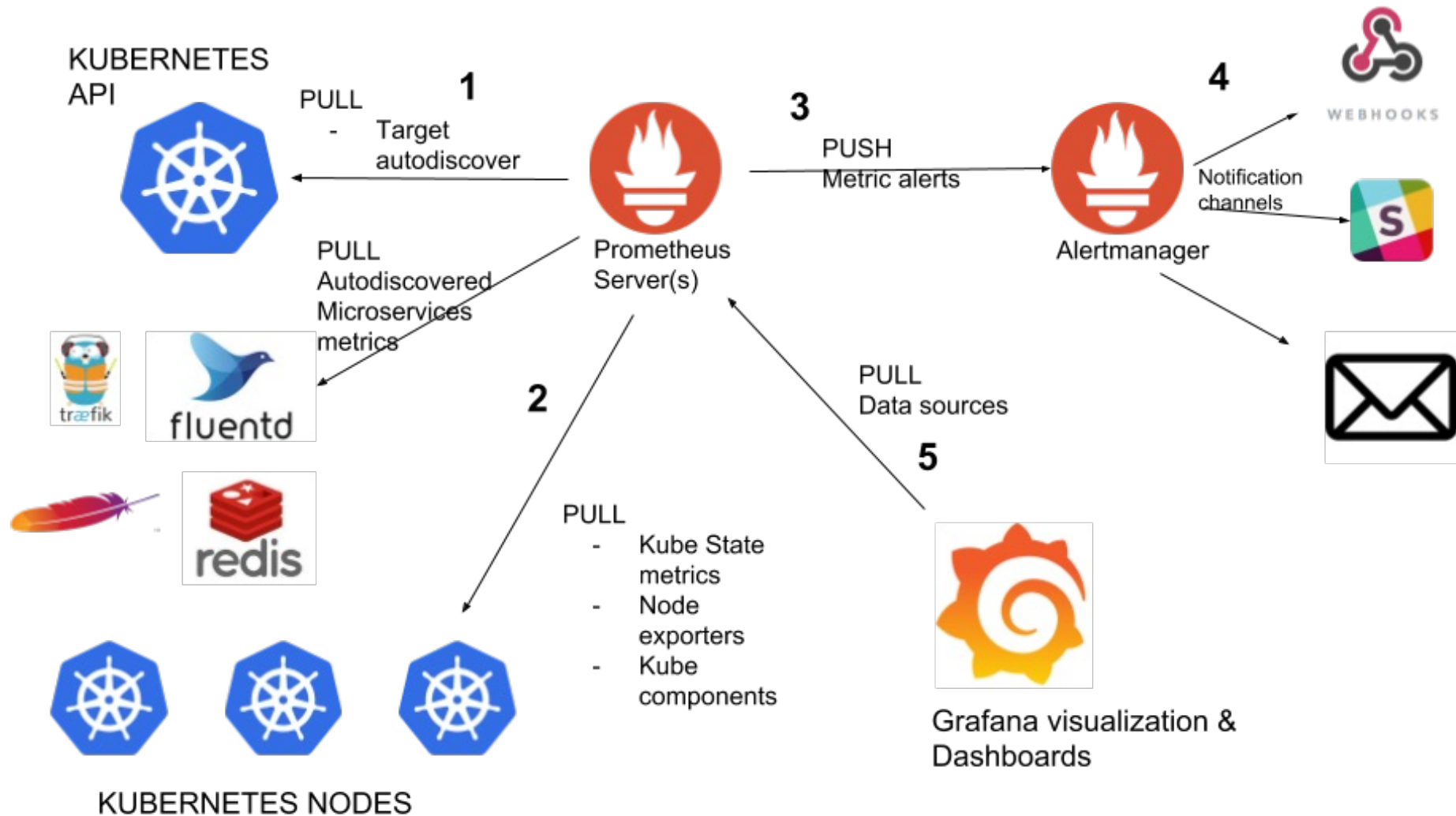
The screenshot shows the Grafana time range picker interface. At the top, there are icons for a dashboard, a panel, settings, and a time range dropdown set to "Last 24 hours". Below this, the "Absolute time range" section is visible. It has two input fields: "From" with the value "now-24h" and "To" with the value "now". A blue "Apply time range" button is positioned below these fields. To the right of the "From" field, a search box labeled "Search quick ranges" is open, displaying a list of quick range options: "Last 5 minutes", "Last 15 minutes", "Last 30 minutes", "Last 1 hour", "Last 3 hours", "Last 6 hours", "Last 12 hours", "Last 24 hours" (which is highlighted), "Last 2 days", and "Last 7 days". Below the "Apply time range" button, there is a message: "It looks like you haven't used this time picker before. As soon as you enter some time intervals, recently used intervals will appear here." followed by a link to "Read the documentation to find out more about how to enter custom time ranges." At the bottom of the interface, the "Browser Time" is shown as "United States, EST" with "UTC-05:00" and a "Change time settings" button.

Grafana: Custom Dashboards

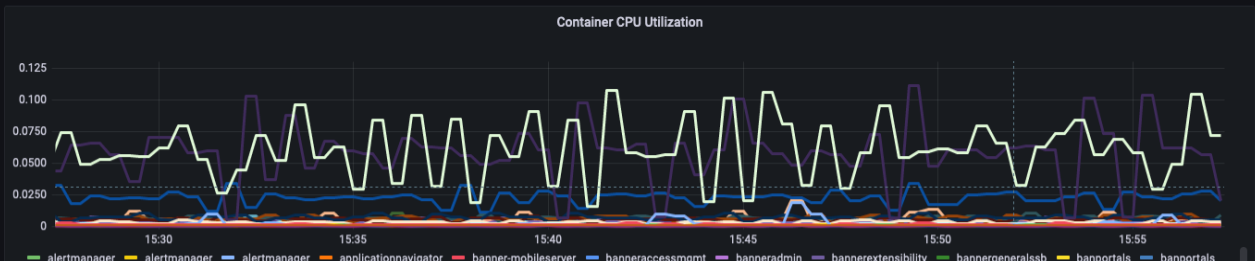
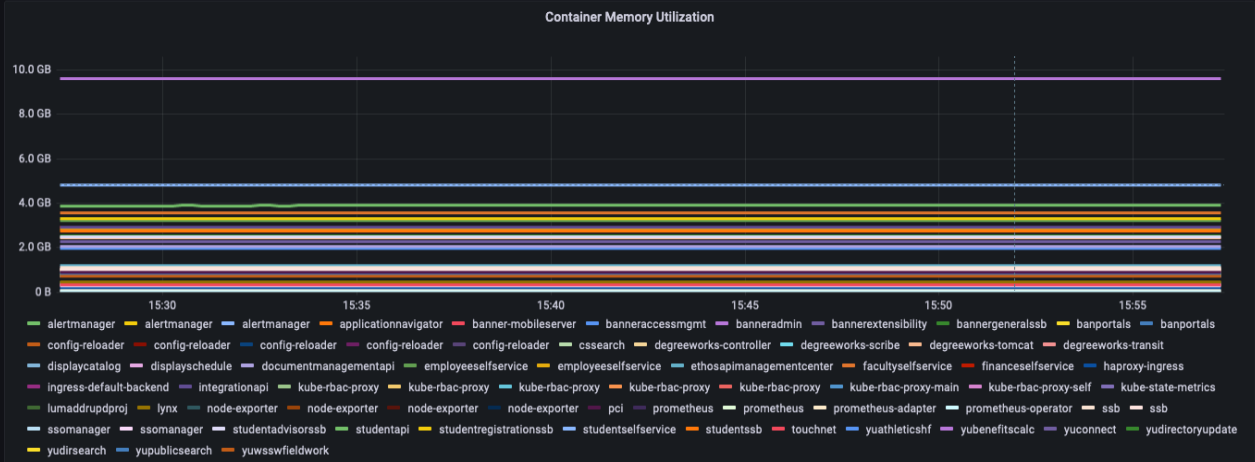
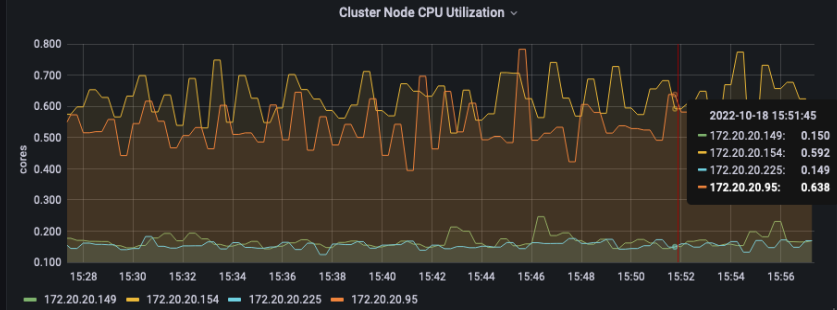
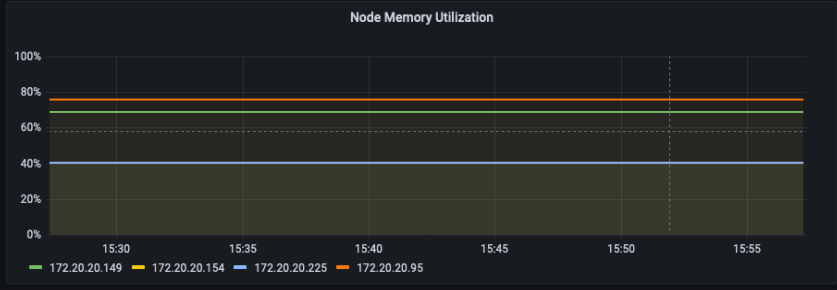
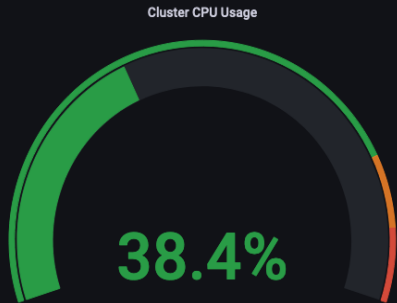
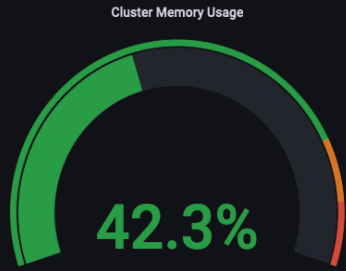


Cloud Native Use Cases

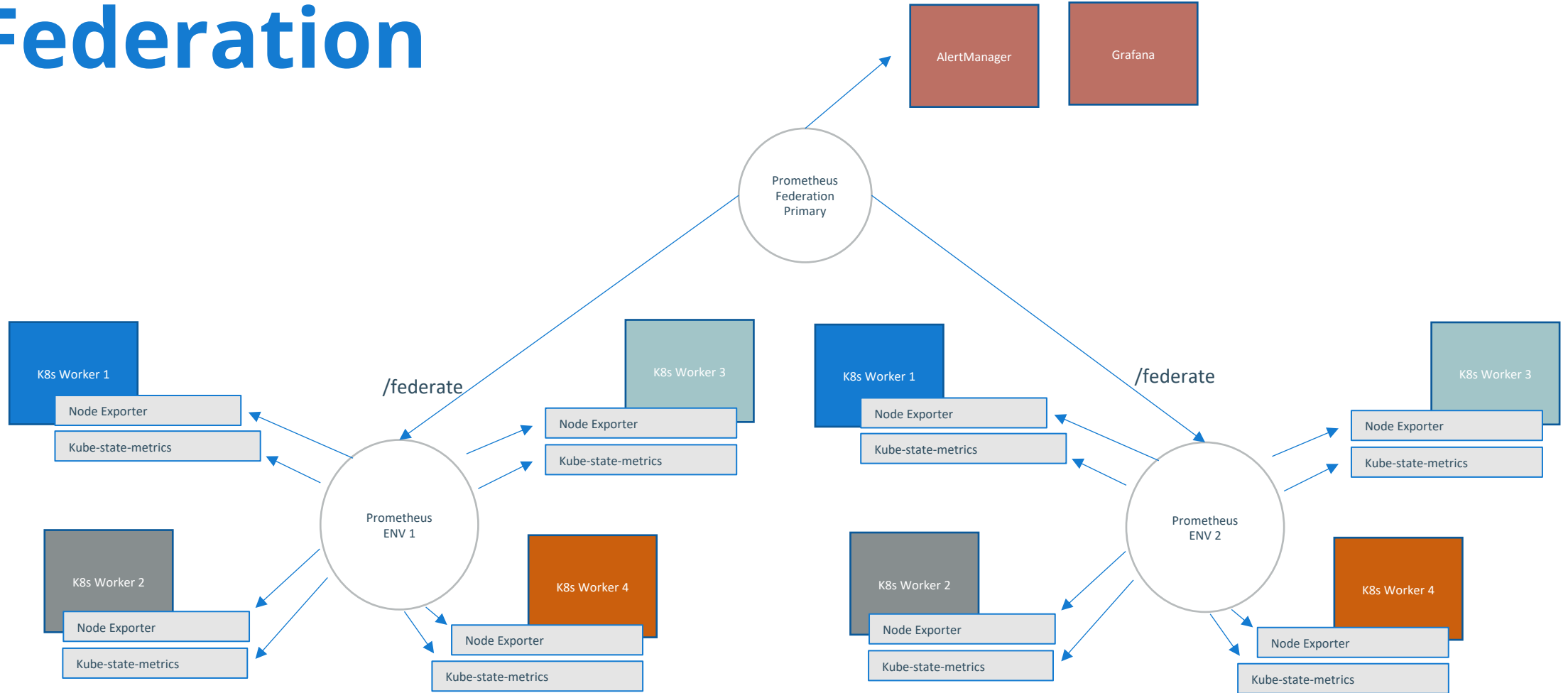
Kube-prometheus



Cluster Health



Federation



GitOps Deployment



cloudalchemy / **ansible-prometheus** Public

<> Code Issues 20 Pull requests 27 Actions Security

master 5 branches 132 tags

SuperQ Merge pull request #380 from sdarwin/feature/to_yaml ...

folder	.chglog	Use git-chglog
folder	.circleci	Update build
folder	.github	switch to molecule v3 and circleci
folder	defaults	🚀 automated upstream release update
folder	handlers	move to molecule 2.x (#108)
folder	meta	Update CI tests (#337)
folder	molecule	Fix circleci
folder	tasks	Retain order in yaml dictionaries
folder	templates	Retain order in yaml dictionaries
folder	vars	Add binary install directory (#259)

prometheus-community / **helm-charts** Public

<> Code Issues 103 Pull requests 13 Actions

main 3 branches 932 tags

dependabot[bot] Bump helm/kind-action from 1.3.0 to 1.4.0 (#2...

folder	.github	Bump helm/kind-action
folder	charts	[prometheus-stackdrive
folder	scripts	fix codeowners workflow
file	.gitignore	Allow rules be enabled i

Saving the Best for Last

Free and Open Source



Prometheus is 100% open source and community-driven. All components are available under the [Apache 2 License](#) on [GitHub](#).

<https://prometheus.io/>

Low Maintenance Overhead

- Minimal configuration: prometheus.yml, alert-rules.yml, alertmanager.yml, Grafana dashboards
- Great community support and community-built rules/dashboards
- No separate database engine - data stored in disk-based time series DB (TSDB)
- Lightweight: Prometheus and exporters are self-contained binaries, easy updates
 - Often bundled into other projects for specific use: Gitlab, OpenShift
- Low barrier to entry. Build, expand and scale as needed




QUESTIONS

Contact

Gabriel Tocci

Senior DBA/Certified Cloud Architect

tocci@sigcorp.com

 /strata-information-group

 @SIGCorpLIVE

 Sigcorp.com

