

# Monitoring Opencast

...

...with Prometheus, Alertmanager and Grafana

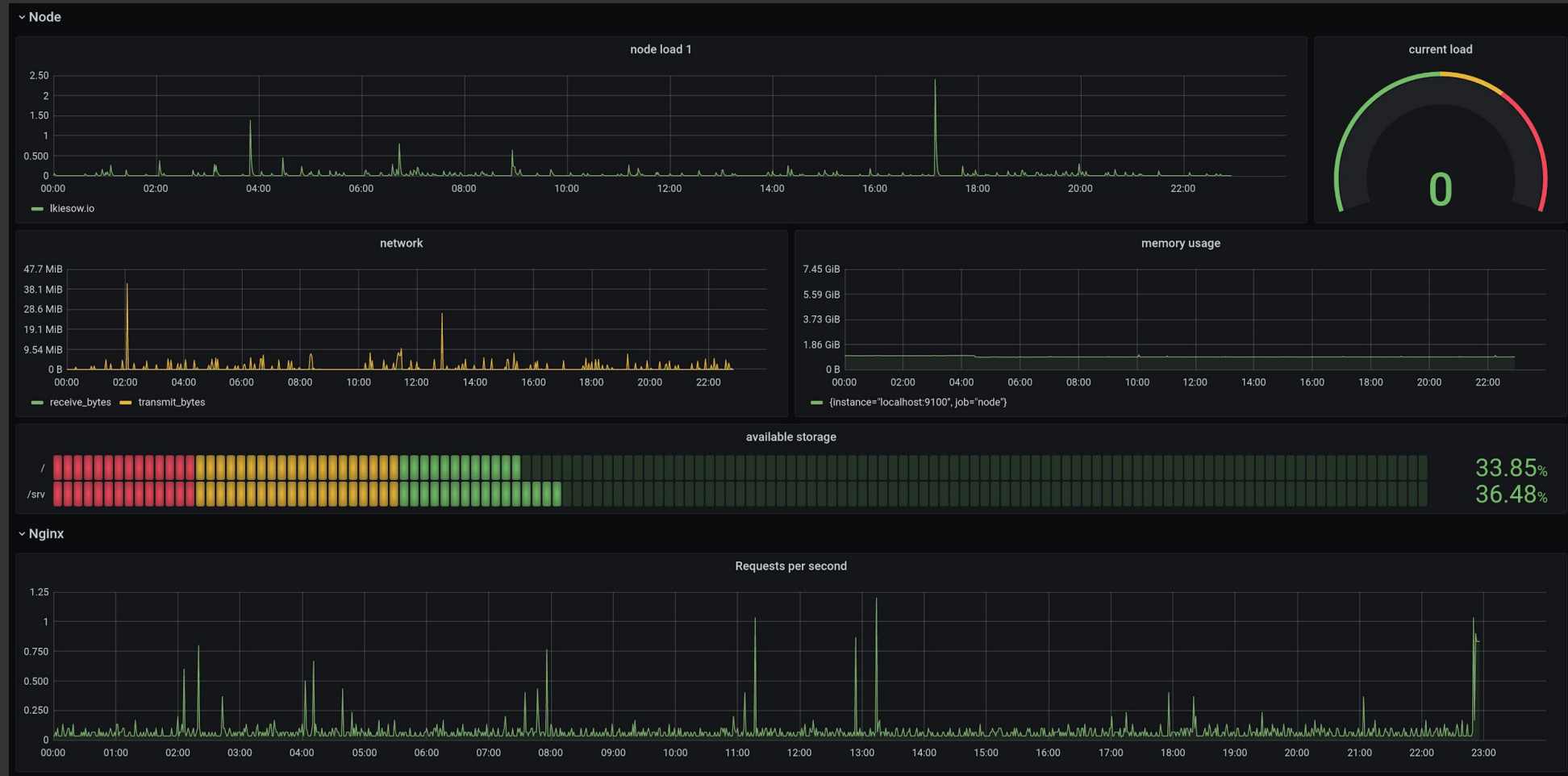
**Monitor services, send alerts, display metrics.**

# Prometheus

From metrics to insight

- Service-based monitoring
- HTTP metrics endpoints
  - Hundreds of stand-alone exporters
  - Built-in exporters in services
- Prometheus scrapes the exporter
  - Internal time-series database

---



node\_exporter and nginx\_exporter metrics

# Exporters

Get metrics from... everything

- `node_exporter`
  - System metrics
- `mysqld_exporter`
  - Database metrics
- `nginx_exporter`
  - HTTP server metrics
  
- Opencast  $\geq$  9.2

[prometheus.io/docs/instrumenting/exporters/](https://prometheus.io/docs/instrumenting/exporters/)

---

```

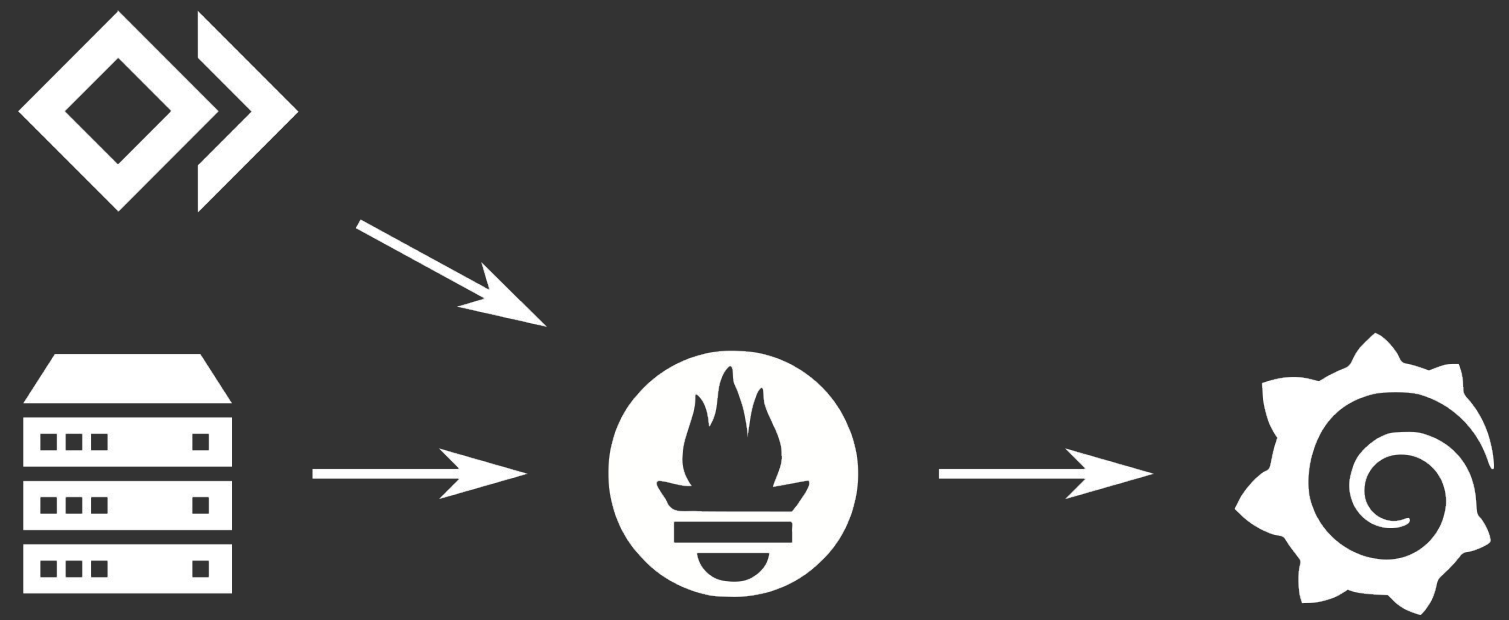
# HELP opencast_job_active Active jobs
# TYPE opencast_job_active gauge
opencast_job_active{host="https://develop.opencast.org",organization="mh_default_org",} 0.0
# HELP opencast_services_total Number of services in a cluster
# TYPE opencast_services_total gauge
opencast_services_total{state="ERROR",} 0.0
opencast_services_total{state="WARNING",} 0.0
opencast_services_total{state="NORMAL",} 88.0
# HELP opencast_version Version of Opencast (based on metrics module)
# TYPE opencast_version gauge
opencast_version{part="major",} 10.0
opencast_version{part="minor",} 0.0
# HELP opencast_job_load_current Maximum job load
# TYPE opencast_job_load_current gauge
opencast_job_load_current{host="https://develop.opencast.org",} 0.0
# ...

```

Metrics endpoint: [develop.opencast.org/metrics/](https://develop.opencast.org/metrics/)



Data flow



Data flow



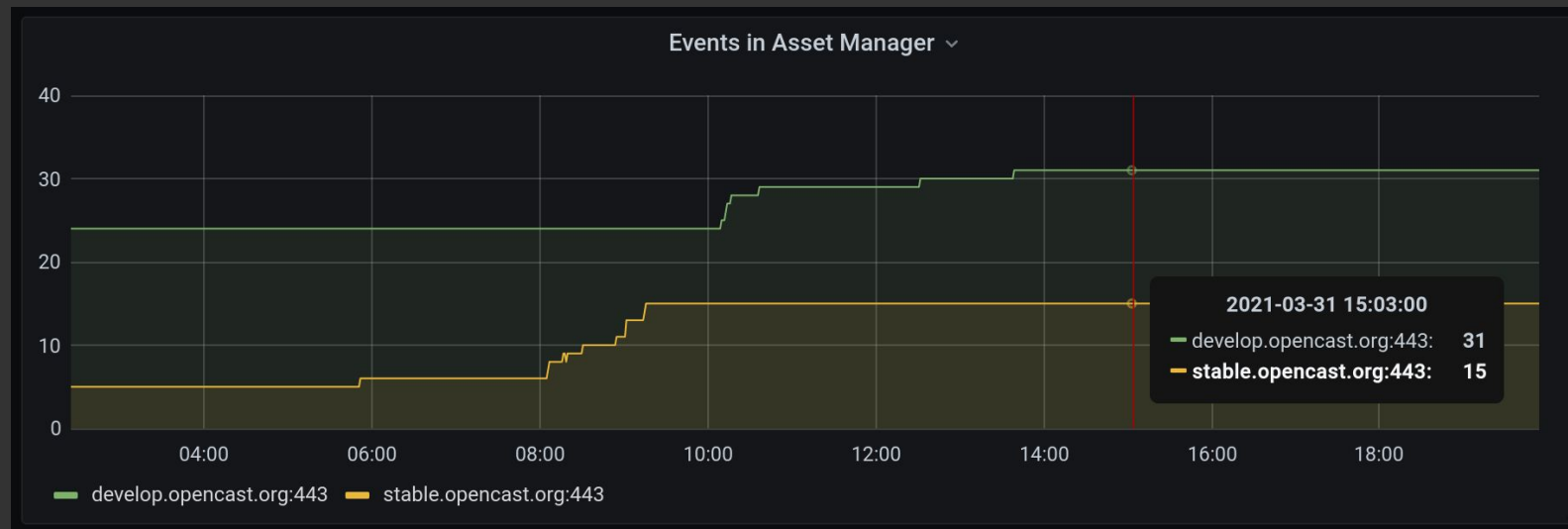
# Opencast

...has an exporter built-in.

- Job load
  - Workflows
  - Services
  - Asset manager
  - Version
  - Up
-



# Opencast specific metrics in 9.2



Additional metrics in 9.3

# Alertmanager

Oh \*\*\*\* that's bad!

- React to certain conditions
- Send notifications
- Based on any metrics you have

**Tip:** Use sparingly

---

1 alert for alertname=InstanceDown

[View In AlertManager](#)

**[1] Firing**

**Labels**

alertname = InstanceDown  
instance = stable.opencast.org:443  
job = opencast  
severity = critical

**Annotations**

description = stable.opencast.org:443 of job opencast has been down for more than 2 minute.  
title = Instance stable.opencast.org:443 down

[Source](#)

Email alert

# Grafana

De-facto standard for visualization

- You have seen it
- It looks good
- It's easy to use
- It's easy to deploy

**Tip:** Start with Grafana Cloud

---

# How to start?

Start simple, then expand.

- All static Go binaries
  - Start with...
    - Opencast
    - Prometheus
    - Grafana
  - Add...
    - node\_exporter
    - Alertmanager
-

```

- job_name: opencast
  scheme: https
  basic_auth:
    username: <oc-user>
    password: <oc-password>
  static_configs:
    - targets:
      - example.opencast.org

```

## Configure Opencast metrics endpoint



```
- alert: InstanceDown
  expr: up == 0
  for: 2m
  annotations:
    title: 'Instance {{ $labels.instance }} down'
    description: '{{ $labels.instance }} of job {{ $labels.job }} has been down for more than 2 minute.'
  labels:
    severity: 'critical'
```

## Alerting

# Statistics

Pretty pictures for your boss and  
for research

- Prometheus is node made statistics
  - It just cares about the moment
  - It will not recover data lost due to network outages, ...
  - Data is not kept indefinitely
- You can use the data nonetheless
  - Answering broader questions works
  - Long-term storage (cortex, Thanos)

done.