Grafana Labs   ☰

Grafana Cloud documentation

Monitoring ingest errors

🏠 › Metrics › Prometheus › Errors monitoring

There are a few places to look for problems and errors when sending metrics to Grafana Cloud.

- The Grafana Cloud billing and usage dashboard available by default and shows ingest errors.
- Both Prometheus and the Grafana Cloud Agent log errors, where there are internal metrics you can look out for.

Problems and solutions are described below.

Usage dashboard

The bottom-right graph shows Discarded Samples aggregated by reason. It queries the metric `grafanacloud_instance_samples_discarded` from the `grafanacloud-usage` data source.
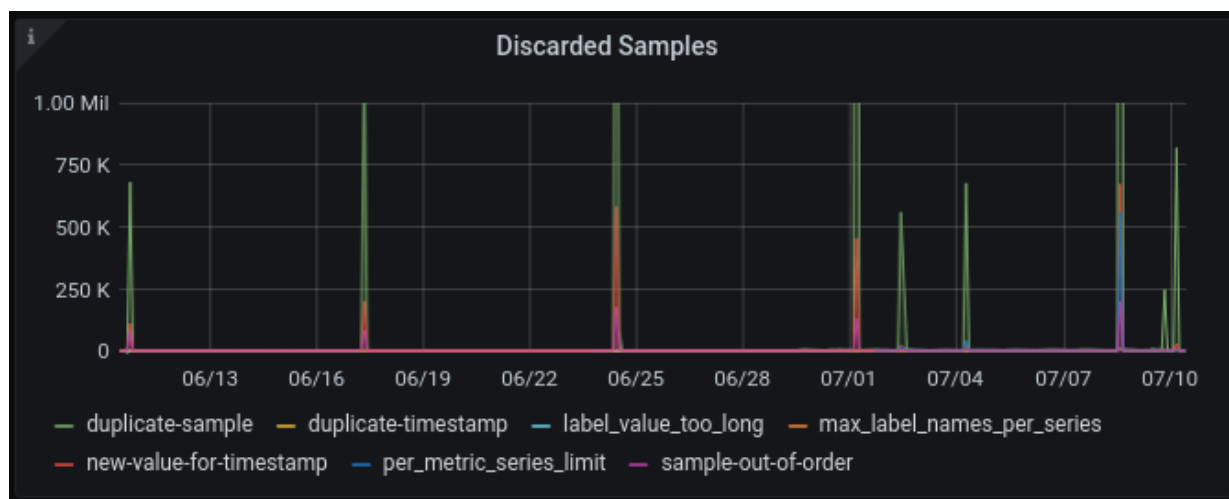
This can help you detect problems, but sometimes you'll need to do a bit of investigation on your side to locate the source of the issue. See the details below in the Ingest problems section.

If you are hitting rate limits or number limits, please contact Grafana Labs support.

See Usage monitoring for more information about the Grafana Cloud billing and usage dashboard.

Prometheus internal metrics

Prometheus exposes internal metrics on its `/metrics` endpoint on port 9090 by default. Make sure to scrape this. https://prometheus.io/docs/prometheus/latest/getting_started/#configuring-prometheus-to-monitor-itself

The metrics to look for (and probably configure an alert for) are:

Metric	Type	Help
`prometheus_remote_storage_failed_samples_total`	counter	Total number of samples which failed on send to remote storage.
`prometheus_remote_storage_dropped_samples_total`	counter	Total number of samples which were dropped due to the queue being full.
`prometheus_remote_storage_queue_length`	gauge	The number of processed samples queued to be sent to the remote storage.
`prometheus_remote_storage_sent_batch_duration_seconds`	histogram	Duration of sample batch send calls to the remote storage.
`prometheus_remote_storage_succeeded_samples_total`	counter	Total number of samples successfully sent to remote storage.

There are also some metrics from the `go-conntrack` library for monitoring HTTP connections. https://gitlab.cncf.ci/prometheus/prometheus/commit/6a633eece19a782cfeddac435b60c1a7abc2488a

Metric	Type

Metric	Type
net_conntrack_dialer_conn_attempted_total{dialer_name="remote_storage"}	counter
net_conntrack_dialer_conn_closed_total{dialer_name="remote_storage"}	counter
net_conntrack_dialer_conn_established_total{dialer_name="remote_storage"}	counter

Metric	Type

Metric	Type
`net_conntrack_dialer_conn_failed_total{dialer_name="remote_storage"}`	counter

Blog   Successes   Community   **Documentation**   ObservabilityCON   Tutorials

Prometheus logs

There are small logs recording `remote_write` activity in Prometheus logs. Failed authentication is logged, but other problems are exposed through internal metrics (see above).

Ingest problems

You may face some issues when sending metrics to Grafana Cloud. For example, hitting some limits or getting failed authentication errors. This section explains the most common errors you may encounter and how to fix them.

Per-metric series limit

The default per-metric series limit is 300K. This is also known as the max cardinality per metric, or the max number of different label values combinations a single metric can have. Grafana Labs support will usually increase this limit after communicating with you, unless this is really a misconfiguration.

From Prometheus best practices on Labels:

> 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.
> https://prometheus.io/docs/practices/naming/#labels

Per-user series limit 🔗

The default active series limit per user is 3 million. This is mostly to protect the cloud platform against misconfigured clients. Grafana Labs support will usually increase this limit after communicating with you.

Max label names per series

The limit is 30.

It's configurable but so far we never increased it for any customer because it comes with a performance penalty. Grafana Labs generally prefers to educate users that having more than 30 labels is just a bad practice.

Greater than max sample age

Samples older than 12 hours in the past are discarded.

Either the machines on which Prometheus is running have their clock set incorrectly, or the remote write queue is 12 hours behind.

Rate limited

There is a soft cap at 100K datapoints per seconds. This is mostly to protect the cloud platform against misconfigured clients. Grafana Labs support will usually increase this limit after communicating with you.

Failed authentication

This error will not show up on the Discarded Samples graph, but you can detect it by monitoring Prometheus logs or the internal metric `prometheus_remote_storage_failed_samples_total` .

Sample Prometheus client error log message for Invalid API Token:

```
ts=2020-06-30T17:25:45.255Z caller=dedupe.go:112 component=remote level=error
remote_name=c78ed9 url=https://prometheus-us-central1.grafana.net/api/prom/push
msg="non-recoverable error" count=100 err="server returned HTTP status 401
Unauthorized: {\"status\": \"error\", \"error\": \"Invalid API Key\"}"
```

Sample Prometheus client error log message for wrong API token role (must be Metrics Publisher):

```
ts=2020-06-30T17:35:00.455Z caller=dedupe.go:112 component=remote level=error
remote_name=872b13 url=https://prometheus-us-central1.grafana.net/api/prom/push
msg="non-recoverable error" count=100 err="server returned HTTP status 401
Unauthorized: {\"status\": \"error\", \"error\": \"invalid authentication credentials,
invalid api key role for attempted action\"}"
```

? Get Support

💬 Community

On this page:

Usage dashboard

Prometheus internal
metrics

Prometheus logs

Ingest problems

　Per-metric series limit

　Per-user series limit

　Max label names per
　series

　Greater than max
　sample age

　Rate limited

　Failed authentication

**Get Product Updates
and News**

Email*

　Email

*required

Subscribe

Grafana Labs

Sitemap

Legal and Security

Terms of Service

Trademark Policy

GRAFANA

Features

Contribute

Dashboards

Plugins

PRODUCTS

Grafana Cloud

Grafana Enterprise

Grafana Metrics Enterprise

OPEN SOURCE

Grafana

Cortex

Graphite

Loki

Metrictank

Prometheus

Tanka

GitHub

LEARN

Blog

Documentation

Downloads

Community

ObservabilityCON

Successes

Training

Webinars and videos

COMPANY

Our Mission

The Team

Press

We're Hiring!

Events

Grafana Cloud Status

Contact