All Example Alerts - Group Name: "Example"

---

example alerts

Alert Name: `folder_disk_usage`

- *Message*: `"folder {{ $labels.path }} crossed disk usage limit on {{$labels.instance}}"` - see example of this in pagerduty

- *Severity*: **page**

- *Owners and contacts*:

    - owner: Example Team
    - owner: Example Data Team
    - slack team: #dd-proxy
    - slack on-call: #proxy_on_call
    - slack data on-call: #data_platform__on_call
    - escalate: escalation path in pagerduty

  There is some multi team ownership here because of the service users of the folder/storage.

- *Symptoms*:
  Root cause is the amount of disk space (in bytes) exceeds a threshold. We monitor these folders in particular for this alert:

    - /opt/proc/visits
    - /opt/proc/kafka
    - /tmp/kafka_uploader see typical usage

  > The monitoring is done via cron with a script that generates metrics for prometheus.

  See the last 5 incidents

- *Impact Assessment*:
  This can be very serious. Particuraly because Kafka is involved as both a consumer and producer. You can get serious issues from any of these Example types:

    - service1
    - service2

  If you have checked most things and you suspect that the Kafka consumer is broken then escalate to the Example Data Platform team.

- *Remediation Actions*:
  The specific action or steps to remediate this alert depend on various factors. But, most importantly it is related to whether we are alerting on folders or partitions. This alert is triggered on folders.

  See the Disk Alert Actions table below for what to do in most cases.

- *Runbook*:

  - This page you are reading
  - Example Team Cookbooks

- *Useful Links*:

  - Logs
  - Dashboard
  - Mega Example Dashboard
  - Example Team Knowledge Center
  - Data Team Knowledge Center
  - Kafka

- *Resolution*:
  Check that the Alert has gone back to the ok state

Disk Alert Actions

The typical investigations to do depends on whether it was a folder or partition resource type, and what was the absolute path in question.

Here are the typical investigative steps (either by logging onto a box or checking external logs etc) so...

Check if there are any files growing too fast, logs files too large or artifact files that got dumped (like core files). Use all the usual du/df type utilities.

[`du -h --apparent-size`, `df -hk /var/log`, `du -sch /`, `lsof | grep deleted`]

If you identify any suspects figures out who owns/created it (process). Can you safely delete it if the process is still running. Do you still need to data so need to export it.
Check the various log files used by any of the Example family of processes. For example:

- proc/fluentbit.log
- proc.log
- syslog
- ...

Something to keep in mind is some files are sparse files so the real usage is a lot less that reported - see `/var/log/lastlog` for example. Are there too many logs for example in /var/log/incapsula? Check if the cause of the disk increase still an issue. Maybe a Kafka consumer is not running but the producer is? Is there any evidence of too many small files been written. Are we running out of file descriptors. In some of these cases we need to reach out to the Data Platform or NetOps teams.

Some other things to looks out for are code dumps/crashes or flaky network connections causing a backup of failed rebase efforts for revlite. Also the common config resources can take up space. You can check this by looking at the git state of the directories - and the associated logs.

Additional detailed information can be found in the Example Disk Monitor page. Also good stuff in the Example Team Knowledge Center and the Example Data Team Knowledge Center