Incident management for small service teams

Michael Hofmann

team background

- CKI: Continuous Kernel Integration - CI as a service
- prevent bugs from being merged into kernel trees
- managing the CI infrastructure for Red Hat's kernel development
- in a nutshell:
  - GitLab pipeline per kernel revision, testing in Beaker
  - platforms: OpenShift, OpenStack, Beaker, AWS EC2
  - RabbitMQ AMQP messaging cluster hosted on AWS
- home page and documentation: https://cki-project.org
- code: https://gitlab.com/cki-project
  - one GitLab CI pipeline and ~ 70 microservices/cron jobs
  - ~20 changes/day merged and automatically deployed to production

introduction

- incident: unexpected event that disrupts business operational processes or reduces the quality of a service (The Internet)
- management: how to mitigate and resolve incidents, and prevent them from happening again
- small service teams: Two Pizza Rule -> between six and ten people

In the following:

- incident detection
- incident management

incident detection

finding out before your customers

detection: monitoring and alerting setup

- the early an incident is detected, the more time there is to fix it
- detecting issues in build/test pipelines, u-services, cron jobs, FaaS, …
- components:
  - logging (Loki)
  - metrics (Prometheus)
  - visualization (Grafana)
  - exceptions (Sentry)
  - alerting (Alertmanager)
- simplify onboarding of services to these as much as possible

logging - Loki

- Loki log aggregation system:
  - set of labels for each log stream, no indexing of log contents
  - ingestion via promtail which pushes to Loki
- log sources:
  - pods: log files + tee stdout to a file, ingest via promtail sidecar
  - cronjobs: tee directly into promtail stdin
  - journald: scrape via promtail
  - AWS CloudWatch, …
- allows log-based alerting
- apply everywhere: Kubernetes YAML templating

metrics - Prometheus

- Prometheus monitoring system:
  - named time series with a set of labels
  - pulls from metrics HTTP endpoints
- data types:
  - counters: can only go up, gauges: can also go down
  - histograms/summaries: estimate distributions
- Python support via prometheus-client
  - extra thread with metrics HTTP endpoint
- Kubernetes autodiscover to scrape metrics from all running pods
- aggregate across namespaces/clusters via federation

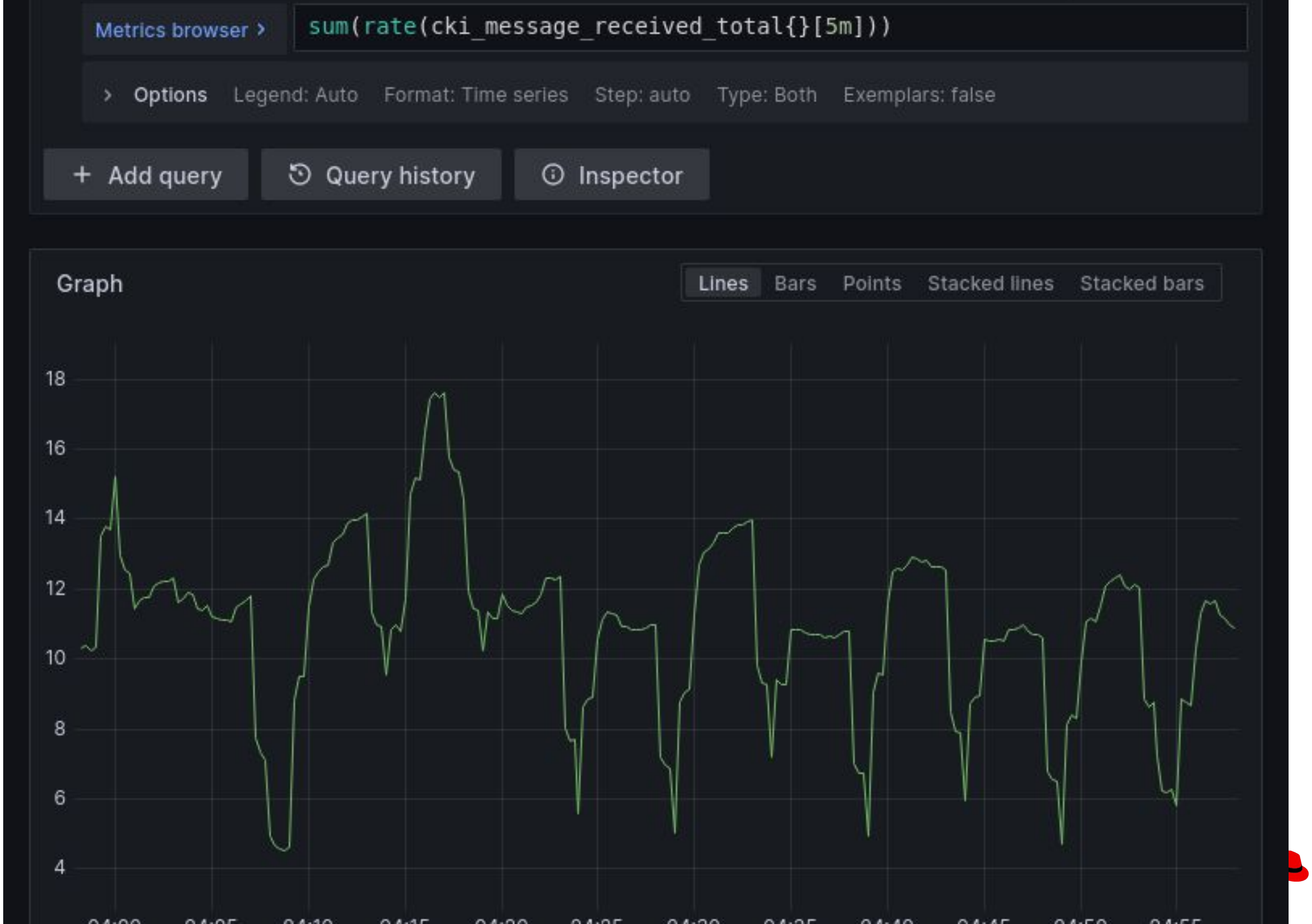
example metric

- Python code:

```
import prometheus_client

METRIC_MESSAGE_RECEIVED = prometheus_client.Counter(
   'cki_message_received', 'Number of queue messages received')

def received callback(function, *args, **kwargs):
   METRIC_MESSAGE_RECEIVED.inc()

prometheus_client.start_http_server(8765)
```

- $ curl service:8765/metrics:

```
# HELP cki_message_received_created Number of queue messages received
# TYPE cki_message_received_created gauge
cki_message_received_created 1.687063249592228e+09
```

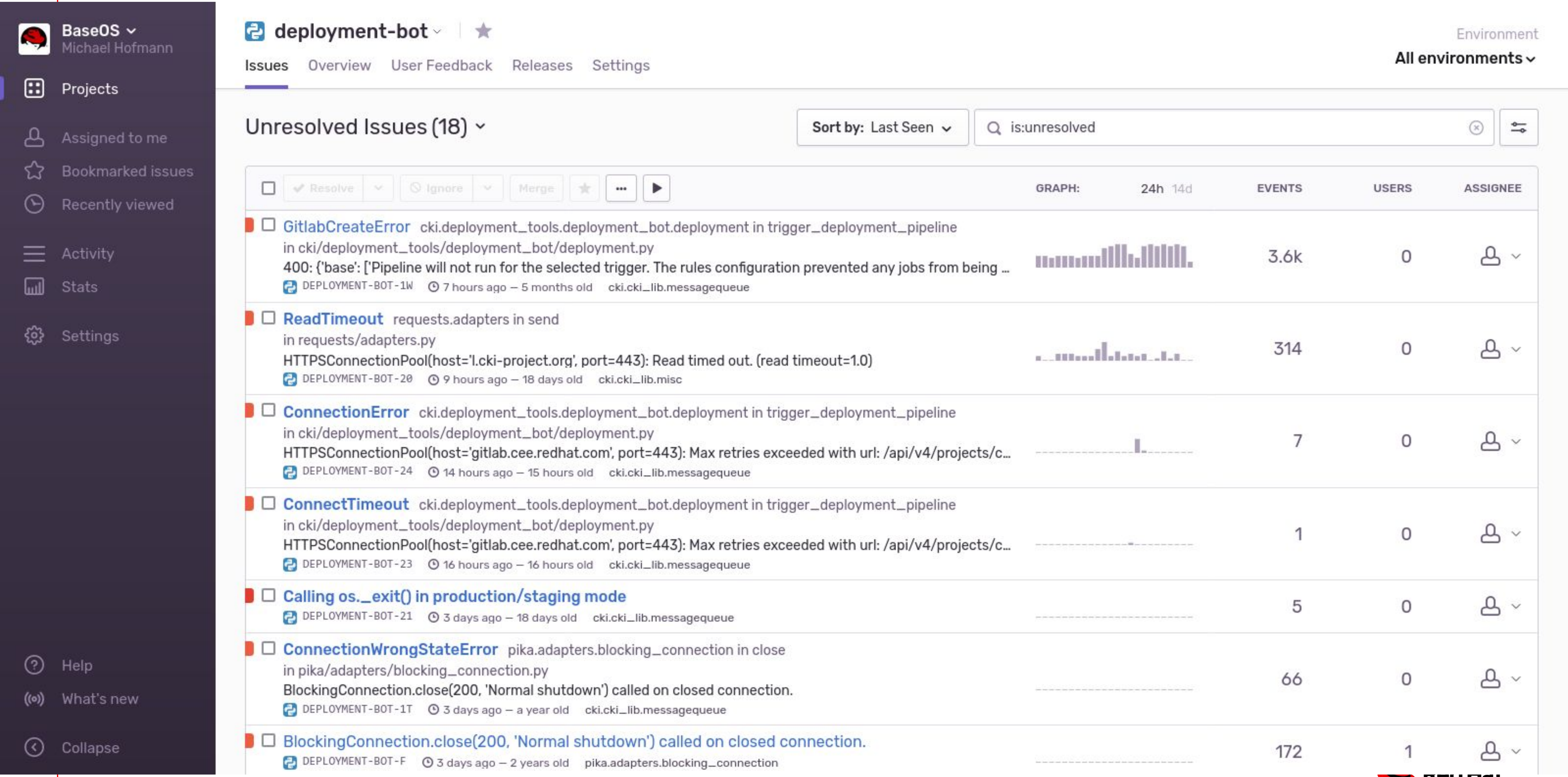


```
sum(rate(cki_message_received_total{}[5m]))
```

Options Legend: Auto Format: Time series Step: auto Type: Both Exemplars: false

exceptions - Sentry

- collect exceptions via Sentry:
  - know about weird issues before your users
  - track errors in real time
  - allows to fix the long tail of unlikely errors
- Python support via sentry-sdk
  - hooks into exception handler, error logging
  - SENTRY_DSN env variable with server + secret
- shows source context, variables, exceptions, SQL, HTTP info, …
  - custom contexts to include more information



Unresolved Issues (18)

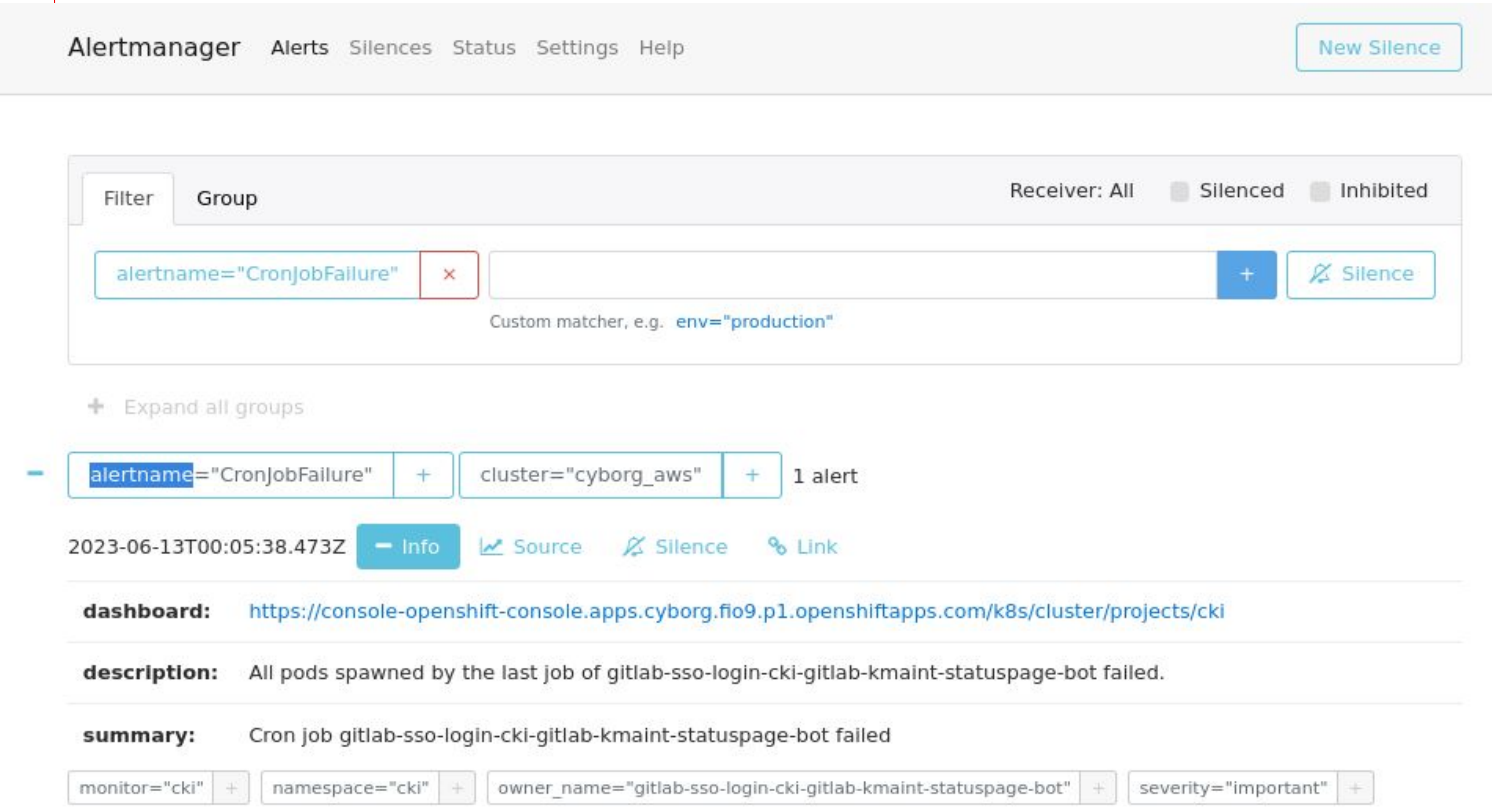
Sort by: Last Seen

is:unresolved

Issue	Events	Users
GitlabCreateError cki.deployment_tools.deployment_bot.deployment in trigger_deployment_pipeline in cki/deployment_tools/deployment_bot/deployment.py 400: {'base': ['Pipeline will not run for the selected trigger. The rules configuration prevented any jobs from being ... DEPLOYMENT-BOT-1W 7 hours ago — 5 months old cki.cki_lib.messagequeue	3.6k	0
ReadTimeout requests.adapters in send in requests/adapters.py HTTPSConnectionPool(host='l.cki-project.org', port=443): Read timed out. (read timeout=1.0) DEPLOYMENT-BOT-20 9 hours ago — 18 days old cki.cki_lib.misc	314	0
ConnectionError cki.deployment_tools.deployment_bot.deployment in trigger_deployment_pipeline in cki/deployment_tools/deployment_bot/deployment.py HTTPSConnectionPool(host='gitlab.cee.redhat.com', port=443): Max retries exceeded with url: /api/v4/projects/c... DEPLOYMENT-BOT-24 14 hours ago — 15 hours old cki.cki_lib.messagequeue	7	0
ConnectTimeout cki.deployment_tools.deployment_bot.deployment in trigger_deployment_pipeline in cki/deployment_tools/deployment_bot/deployment.py HTTPSConnectionPool(host='gitlab.cee.redhat.com', port=443): Max retries exceeded with url: /api/v4/projects/c... DEPLOYMENT-BOT-23 16 hours ago — 16 hours old cki.cki_lib.messagequeue	1	0
Calling os._exit() in production/staging mode DEPLOYMENT-BOT-21 3 days ago — 18 days old cki.cki_lib.messagequeue	5	0
ConnectionWrongStateError pika.adapters.blocking_connection in close in pika/adapters/blocking_connection.py BlockingConnection.close(200, 'Normal shutdown') called on closed connection. DEPLOYMENT-BOT-1T 3 days ago — a year old cki.cki_lib.messagequeue	66	0
BlockingConnection.close(200, 'Normal shutdown') called on closed connection. DEPLOYMENT-BOT-F 3 days ago — 2 years old pika.adapters.blocking_connection	172	1

alerting - Alertmanager

- surface alerts via Alertmanager:
  - group and route alerts to an alerting destination
  - alerts can also be inhibited and silenced
- Loki/Prometheus define alerts based on logs/metrics
- destinations: email, Slack, text messages, HTTP endpoints, …
- configurable templates, timing for grouped messages



alertname="CronJobFailure" ×

Custom matcher, e.g. env="production"

alertname="CronJobFailure" + cluster="cyborg_aws" + 1 alert

2023-06-13T00:05:38.473Z

dashboard: https://console-openshift-console.apps.cyborg.fio9.p1.openshiftapps.com/k8s/cluster/projects/cki

description: All pods spawned by the last job of gitlab-sso-login-cki-gitlab-kmaint-statuspage-bot failed.

summary: Cron job gitlab-sso-login-cki-gitlab-kmaint-statuspage-bot failed

monitor="cki" + namespace="cki" + owner_name="gitlab-sso-login-cki-gitlab-kmaint-statuspage-bot" + severity="important" +

incident management

just fix it you said?

incident handling process

- incident handling has both technical and social challenges
- technical:
  - mitigate, fix immediate issues
  - fix root cause
- social:
  - who does what when, if at all
- small service teams:
  - no dedicated site reliability engineers (SREs)
  - incident handling is a team responsibility

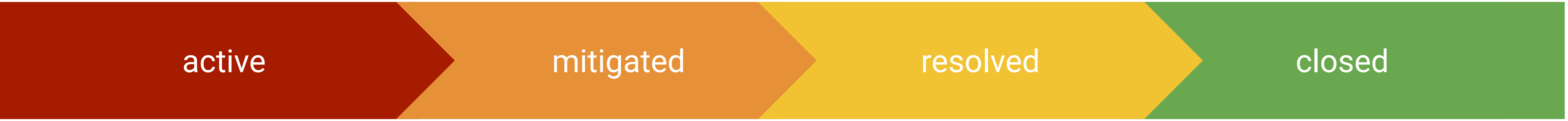
historical approach – just fix it

- hand it to the most knowledgeable person in the (chat) room
- advantages:
  - very short time to recovery (unless that person is on PTO)
- disadvantages:
  - bus factor of ~one
  - no knowledge transfer
  - burnout risk
  - move fast and (hopefully not) break things

currently in use – structured approach

- structured incident handling approach
  - https://cki-project.org/docs/contributing/incidents/
- first thing: create a public incident ticket (I know 😱)
  - collect information, screen shots, links
  - use confidential comments for internal information
- proceed in phases

everything is better in ~~layers~~phases



active

mitigated

resolved

closed

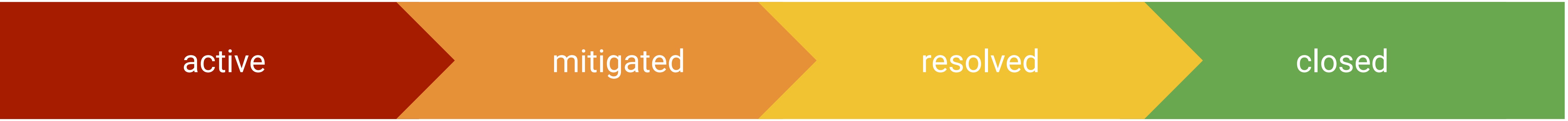
reduce the impact on the production environment

resolve the direct cause of the incident

improve on the root cause of the incident

example: spot instances fail to launch

- spot instance price limits for docker-machine were set too low
- spot price increase resulted in UnfulfillableCapacity error
- docker-machine default is set to USD 0.50



active

mitigated

resolved

closed

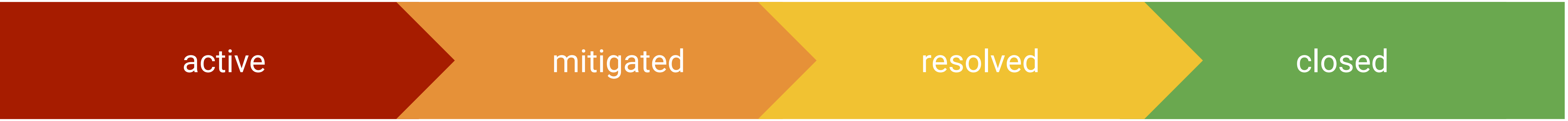
increase spot price limit directly on gitlab-runner VMs

increase spot price limit properly in GitOps

remove spot price limit default so it is not necessary to set spot price limit at all

example: updated SSL certificates are broken

- renewal of SSL certificates for somesite.host.org went wrong
  - before, public CA was used; for renewal, internal CA was used
  - broke all customers without the internal CA cert in their bundle
  - customers complained on the mailing list



active

mitigated

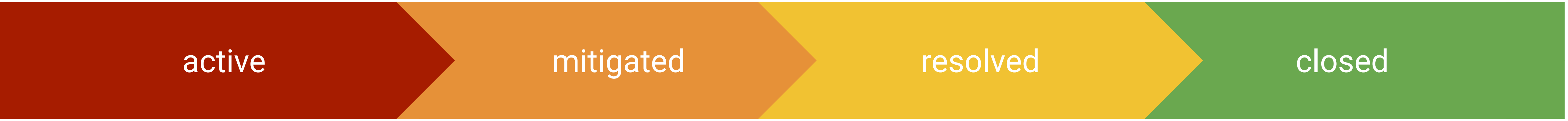
resolved

closed

ask your audience!

example: updated SSL certificates are broken

- renewal of SSL certificates for somesite.host.org went wrong
  - before, public CA was used; for renewal, internal CA was used
  - broke all customers without the internal CA cert in their bundle
  - customers complained on the mailing list



active

mitigated

resolved

closed

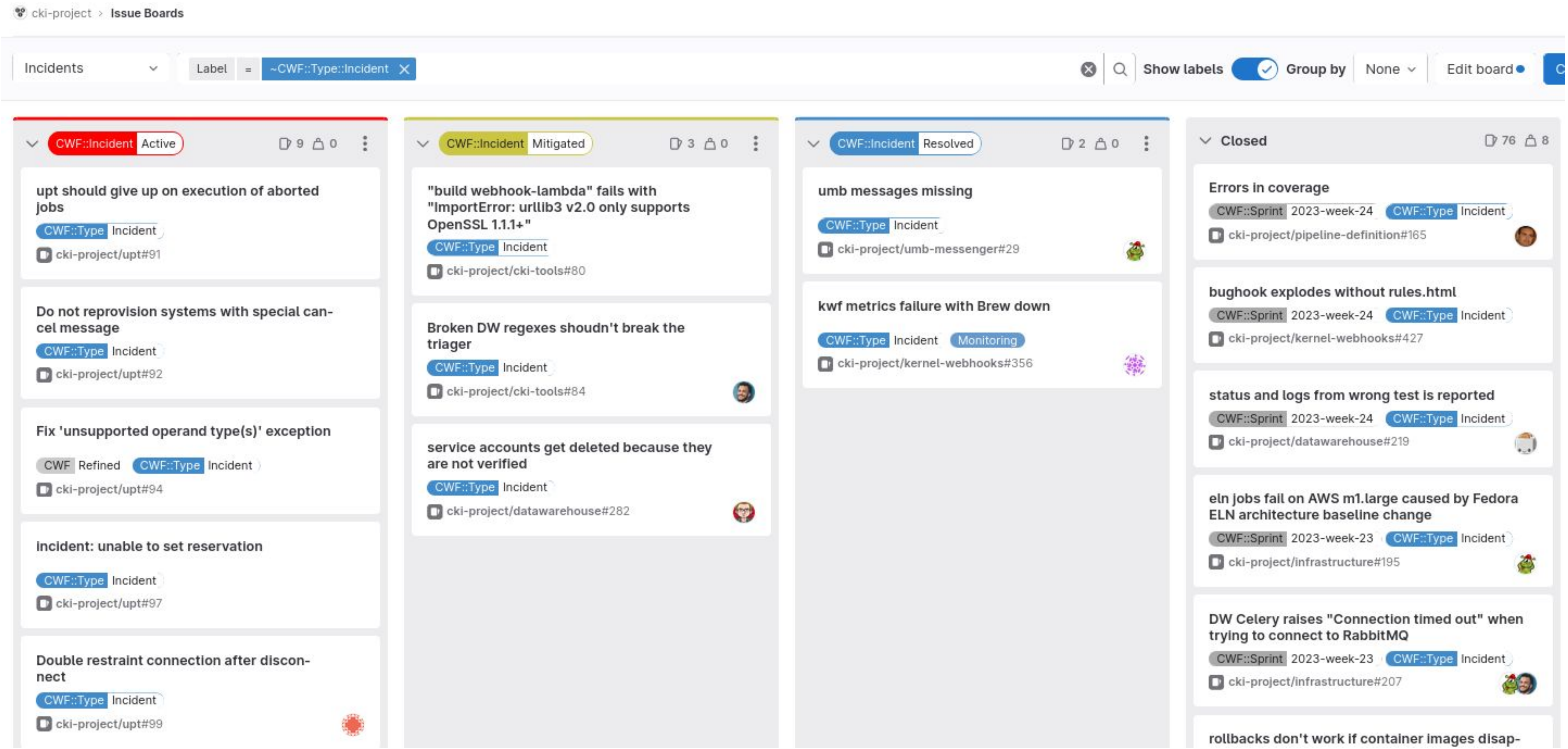
restore previous SSL certificate

re-request SSL certificate from public CA

add monitoring for correct CA
automate certificate renewal

social dynamics and summary

- dealing with social dynamics:
  - no phase is allowed to be skipped
  - later phases are not less important
  - different phases can be handled by different people
  - use a Kanban board to track progress
  - weekly review meeting
- surprisingly this actually more-or-less works



cki-project > Issue Boards

Incidents | Label = ~CWF::Type::Incident | Show labels | Group by None | Edit board

CWF::Incident Active 9 0

upt should give up on execution of aborted jobs
CWF::Type Incident
cki-project/upt#91

Do not reprovision systems with special cancel message
CWF::Type Incident
cki-project/upt#92

Fix 'unsupported operand type(s)' exception
CWF Refined CWF::Type Incident
cki-project/upt#94

incident: unable to set reservation
CWF::Type Incident
cki-project/upt#97

Double restraint connection after disconnect
CWF::Type Incident
cki-project/upt#99

CWF::Incident Mitigated 3 0

"build webhook-lambda" fails with "ImportError: urllib3 v2.0 only supports OpenSSL 1.1.1+"
CWF::Type Incident
cki-project/cki-tools#80

Broken DW regexes shoudn't break the triager
CWF::Type Incident
cki-project/cki-tools#84

service accounts get deleted because they are not verified
CWF::Type Incident
cki-project/datawarehouse#282

CWF::Incident Resolved 2 0

umb messages missing
CWF::Type Incident
cki-project/umb-messenger#29

kwf metrics failure with Brew down
CWF::Type Incident Monitoring
cki-project/kernel-webhooks#356

Closed 76 8

Errors in coverage
CWF::Sprint 2023-week-24 CWF::Type Incident
cki-project/pipeline-definition#165

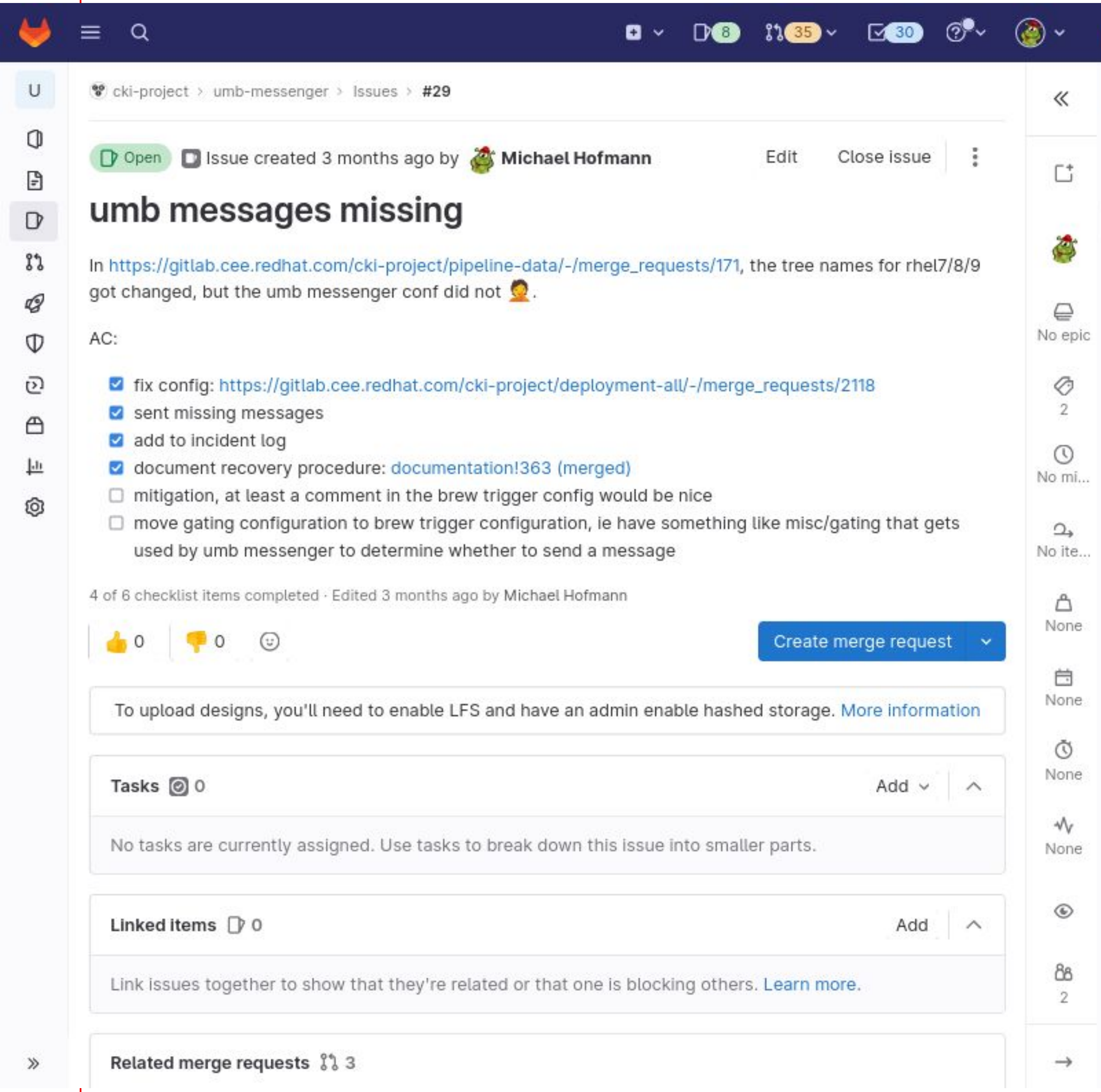
bughook explodes without rules.html
CWF::Sprint 2023-week-24 CWF::Type Incident
cki-project/kernel-webhooks#427

status and logs from wrong test is reported
CWF::Sprint 2023-week-24 CWF::Type Incident
cki-project/datawarehouse#219

eln jobs fail on AWS m1.large caused by Fedora ELN architecture baseline change
CWF::Sprint 2023-week-23 CWF::Type Incident
cki-project/infrastructure#195

DW Celery raises "Connection timed out" when trying to connect to RabbitMQ
CWF::Sprint 2023-week-23 CWF::Type Incident
cki-project/infrastructure#207

rollbacks don't work if container images disap-



Issue created 3 months ago by Michael Hofmann

umb messages missing

In https://gitlab.cee.redhat.com/cki-project/pipeline-data/-/merge_requests/171, the tree names for rhel7/8/9 got changed, but the umb messenger conf did not 🤦.

AC:

- ☑ fix config: https://gitlab.cee.redhat.com/cki-project/deployment-all/-/merge_requests/2118
- ☑ sent missing messages
- ☑ add to incident log
- ☑ document recovery procedure: documentation!363 (merged)
- □ mitigation, at least a comment in the brew trigger config would be nice
- □ move gating configuration to brew trigger configuration, ie have something like misc/gating that gets used by umb messenger to determine whether to send a message

4 of 6 checklist items completed · Edited 3 months ago by Michael Hofmann

👍 0 👎 0

To upload designs, you'll need to enable LFS and have an admin enable hashed storage. More information

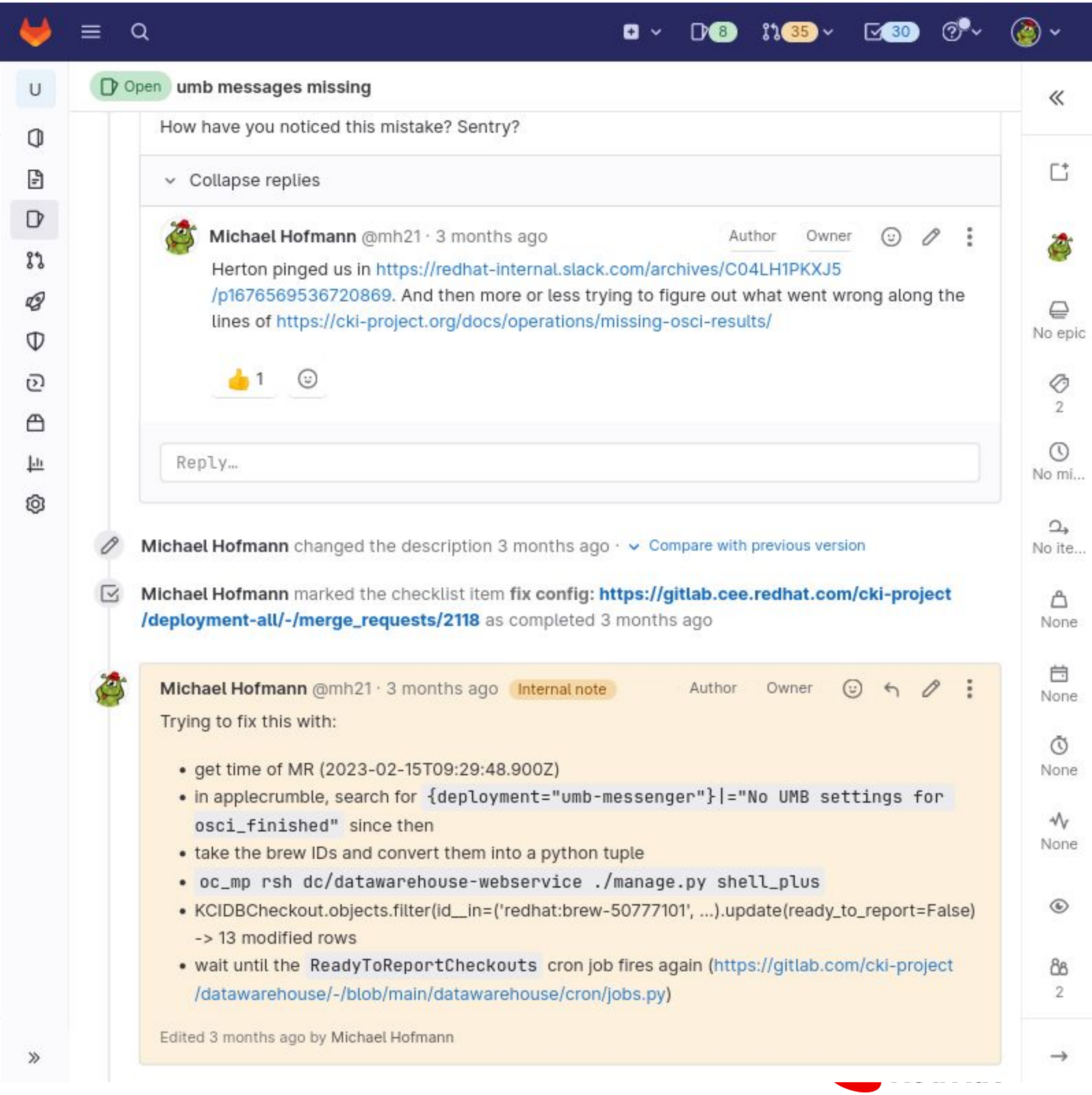
Tasks 0 Add

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0 Add

Link issues together to show that they're related or that one is blocking others. Learn more.

Related merge requests 3



umb messages missing

How have you noticed this mistake? Sentry?

Collapse replies

Michael Hofmann @mh21 · 3 months ago Author Owner

Herton pinged us in https://redhat-internal.slack.com/archives/C04LH1PKXJ5/p1676569536720869. And then more or less trying to figure out what went wrong along the lines of https://cki-project.org/docs/operations/missing-osci-results/

👍 1

Reply…

Michael Hofmann changed the description 3 months ago · Compare with previous version

Michael Hofmann marked the checklist item **fix config: https://gitlab.cee.redhat.com/cki-project/deployment-all/-/merge_requests/2118** as completed 3 months ago

Michael Hofmann @mh21 · 3 months ago Internal note Author Owner

Trying to fix this with:

- get time of MR (2023-02-15T09:29:48.900Z)
- in applecrumble, search for `{deployment="umb-messenger"}|="No UMB settings for osci_finished"` since then
- take the brew IDs and convert them into a python tuple
- `oc_mp rsh dc/datawarehouse-webservice ./manage.py shell_plus`
- KCIDBCheckout.objects.filter(id__in=('redhat:brew-50777101', ...).update(ready_to_report=False) -> 13 modified rows
- wait until the `ReadyToReportCheckouts` cron job fires again (https://gitlab.com/cki-project/datawarehouse/-/blob/main/datawarehouse/cron/jobs.py)

Edited 3 months ago by Michael Hofmann

🤗 Question time 🤗