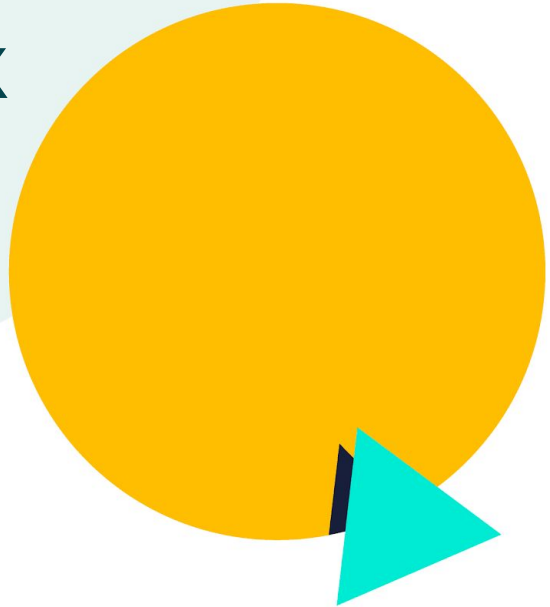
wrike

Introducing Promabbix Bringing APM from Prometheus to Zabbix at Scale



David Pech, 2025

Wrike - About Us

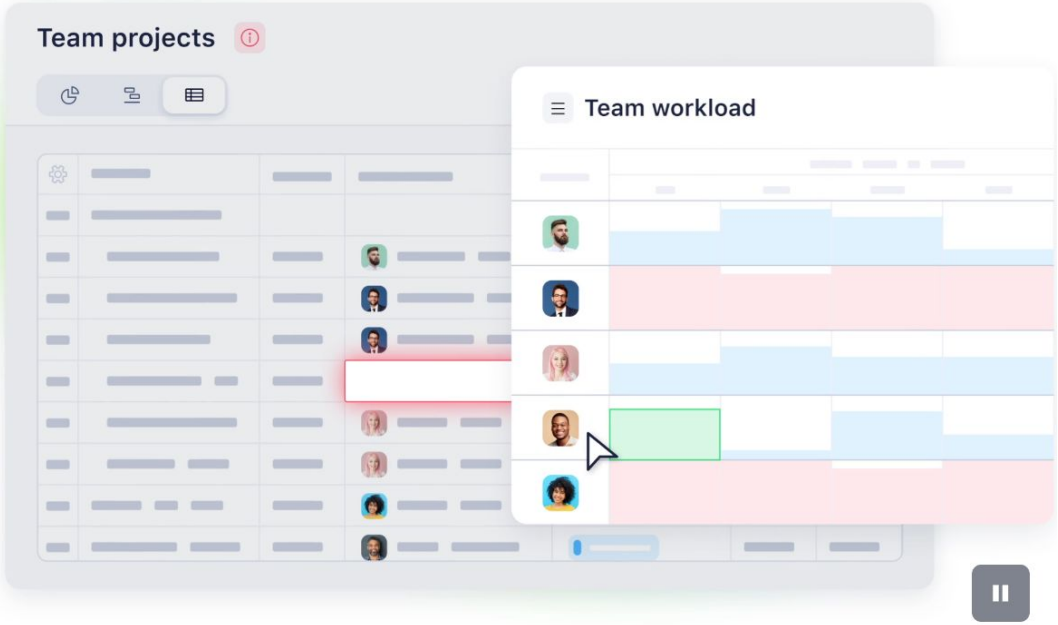
Andrey Menzhinskiy

David Pech

One platform to streamline all workflows

Experience the freedom to build, connect, automate, and scale your work with Wrike.

Log in Contact us



Our Approach to Monitoring

- Own the stack, no vendor-lock in
- Cloud Native, but staying conservative
- Everything in GIT
- Everything automated (git push -> pipeline to apply the config)
- Single pane of glass
- Same stack for both infrastructure and applications

Scale of Wrike

- 2 On-Prem locations (US, EU), mostly GCP, but also AWS, Azure
- 3 Separate Zabbix instances, 15 Zabbix Proxies
- ~ 2300 Servers in Zabbix
- 20 Kubernetes clusters in Zabbix, running ~2.5k Deployments and ~6k Pods

- Largest Zabbix Postgres (GCP CloudSQL) - 10 TB (Bug)
- Technology Ownership Sheet ~100 lines
    - Interviews: “We don’t enumerate technologies in our stack, just ask for any technology and likely we’ll say that we operate it to some extent”.

Application (APM) vs Infrastructure Monitoring

- Typically very different use-cases and tooling
- For infrastructure typically much more oriented around how the infrastructure is composed
  - Server A, Disk sda, Partition 1 is running out of space
- APM much more around business values / higher level queries
  - Error rate for all API server Pods is < 1% in last 5 mins

Wrike History Lesson

- Zabbix before 2020
- Kubernetes in production since 2020, with Istio since 2020
- Slowly moving from VMs -> Kubernetes for the whole stack
- Currently: Zabbix 6.0

Sticking to Zabbix

- We like Zabbix
  - Every morning the whole team goes over Daily - alert history for yesterday
- Continuity and utilizing previous investments
- Zabbix-agents on VMs

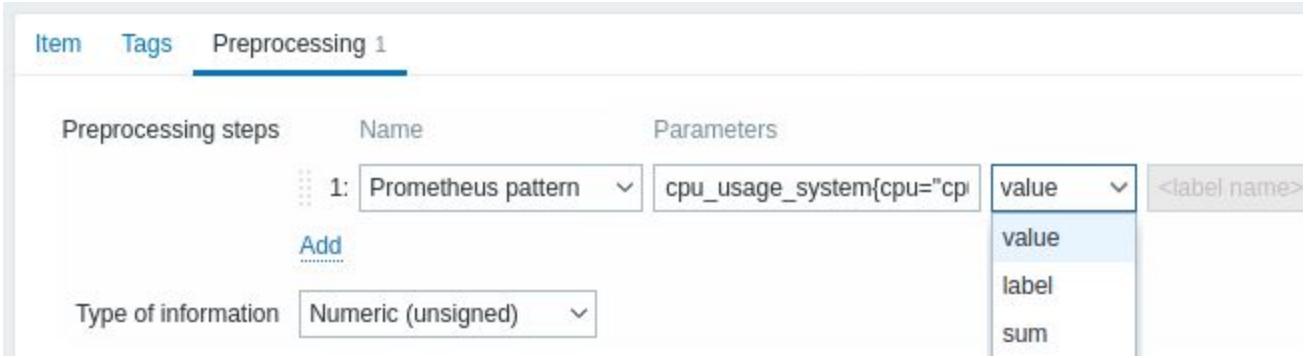
Single Pane of Glass

- Prometheus and AlertManager don’t allow good history view
  - vs. Easy-to-use Zabbix Top100 triggers
- Thresholds definition - single source of truth
- Single place-to-go for all currently active alerts
- Used both by operations and development teams
- … and much more

Can we have it, please?

Prom.>Z.: Scrape Metric Endpoints by Zabbix

- Not for production



Prom.>Z.: Load (a lot of) Data from kube-apiserver

- Templates to read from kube-api server via API
- Well…
  - Scalability issues
  - Pod-level items, their TTL after the Pod is dead
- We want much advanced alerting
  - One of the Pods is healthy (we don’t care which)
  - P99 of latency for service A is < 1s in 5min interval

Prom.>Z.: Alertmanager Alerts to Zabbix Trapper Item

- No logic and structure / evaluation in Zabbix
- Mostly separate stacks with a minimal connection / interstep between the two
- Project abandoned
  - https://github.com/gmauleon/alertmanager-zabbix-webhook

Bottom line

- Incompatible data models
  - Metric with dimensions vs. Host -> Item hierarchy
- ClickOps in Zabbix is standard
- Scaling is problematic
- Scraping intervals
  - minutes in Zabbix vs. sub-minute in Prometheus
  - number of targets, service-discovery
- Alerting approaches
  - Mostly single-host Zabbix vs. over dozen of time-series Prometheus

Zabbix 8.0 LTS | Zabbix 8.2

Zabbix 8.0 LTS

Planned release date: Q2 2026

Observability

- OpenTelemetry data collection In design
  Cloud native scalable collection and processing of OpenTelemetry data
  ⭐ Top voted!
- OpenTelemetry data visualization
  New view designed for efficient search and visualization of OpenTelemetry tracing data
  ⭐ Top voted!
- Optimized storage engine for telemetry
  Support of new storage engines optimized for scalable storage and retrieval of large volumes of tracing, log, time-series as well as any streaming data
- Log based observability
  Enables real-time monitoring and troubleshooting by leveraging log information for a more comprehensive understanding of system performance and health

Zabbix Mobile Application

- Mobile Application In dev New
  Zabbix Mobile Application for iOS and Android platforms will provide push-notifications, access to problems and historical data, and easy to use problem management functions. It will be released along with Zabbix 8.0 LTS.
  ⭐ Top voted!

Event correlation and enterprise alarm console

- Complex event processing engine In design

Introducing Promabbix


ZABBIX
PROMABBIX

The Gist of Promabbix

- Prometheus / OpenTelemetry vs. Zabbix can’t be merged in a single tool

BUT

- We can query Prometheus from Zabbix
  - To dynamically enumerate / discover existing time-series (in Prometheus)
  - Calculate alerting metrics per each time-series (in Prometheus)
  - AND Evaluate triggers based on that (in Zabbix)

Promabbix Architecture

Promabbix Configs

AlertManager Recording Rules (PromQL)

sum(jaeger_collector_queue_length) by (k8s_cluster)

Zabbix Trigger Expressions

jaeger_collector_queue_len > {$JAEGER.QUEUE.LEN}

Pseudo-hosts and Macro (Thresholds)

- Jaeger QA, {$JAEGER.QUEUE.LEN}: 10

Run Promabbix CLI

Promabbix CLI generates Zabbix Template and Hosts via Zabbix API

Pseudo-hosts with discovery Items

Typically one per environment, defines LLD Macros

- Jaeger Production (with k8s_cluster =~ PROD)
- Jaeger QA (with k8s_cluster !~ PROD)

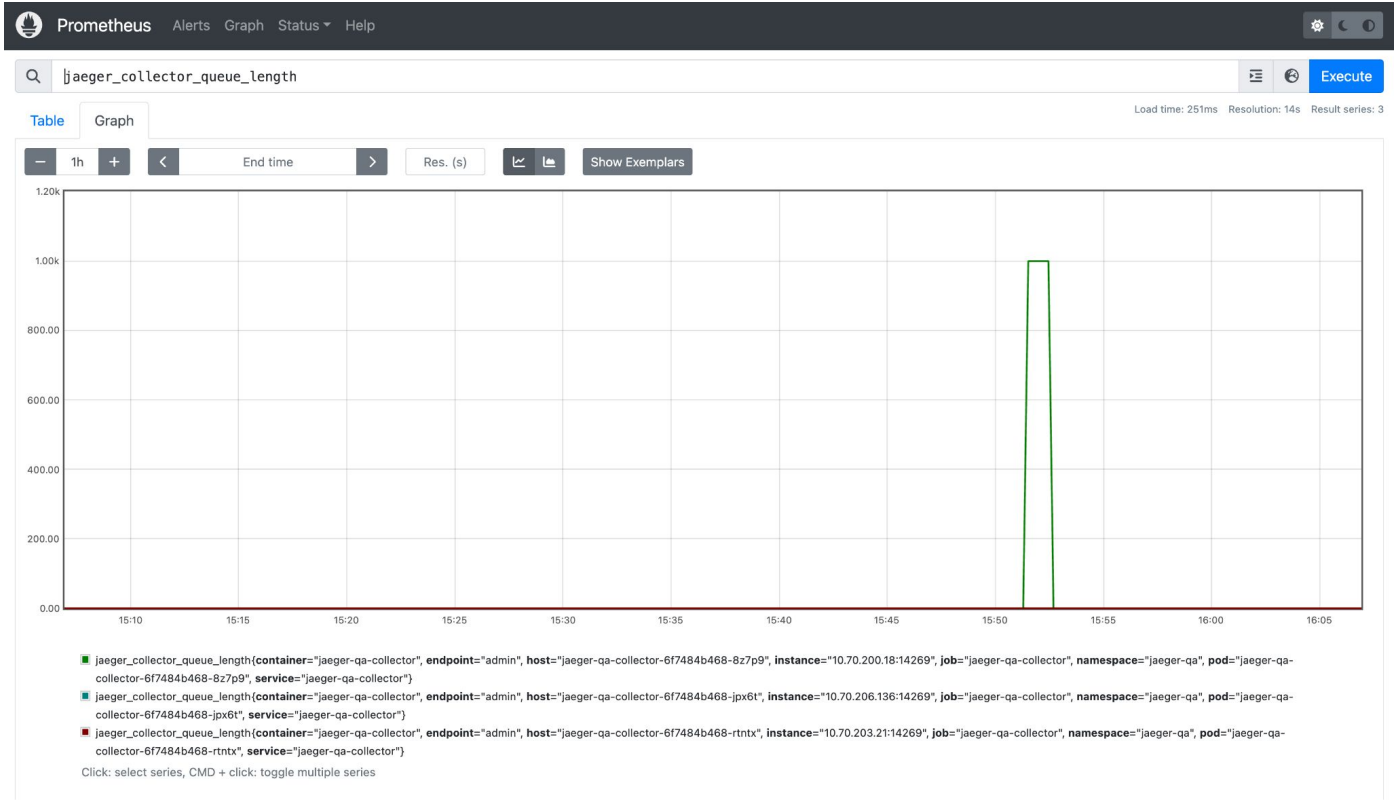
Zabbix makes Query to Prometheus

and creates Items with LLD discovery for existing time-series combinations

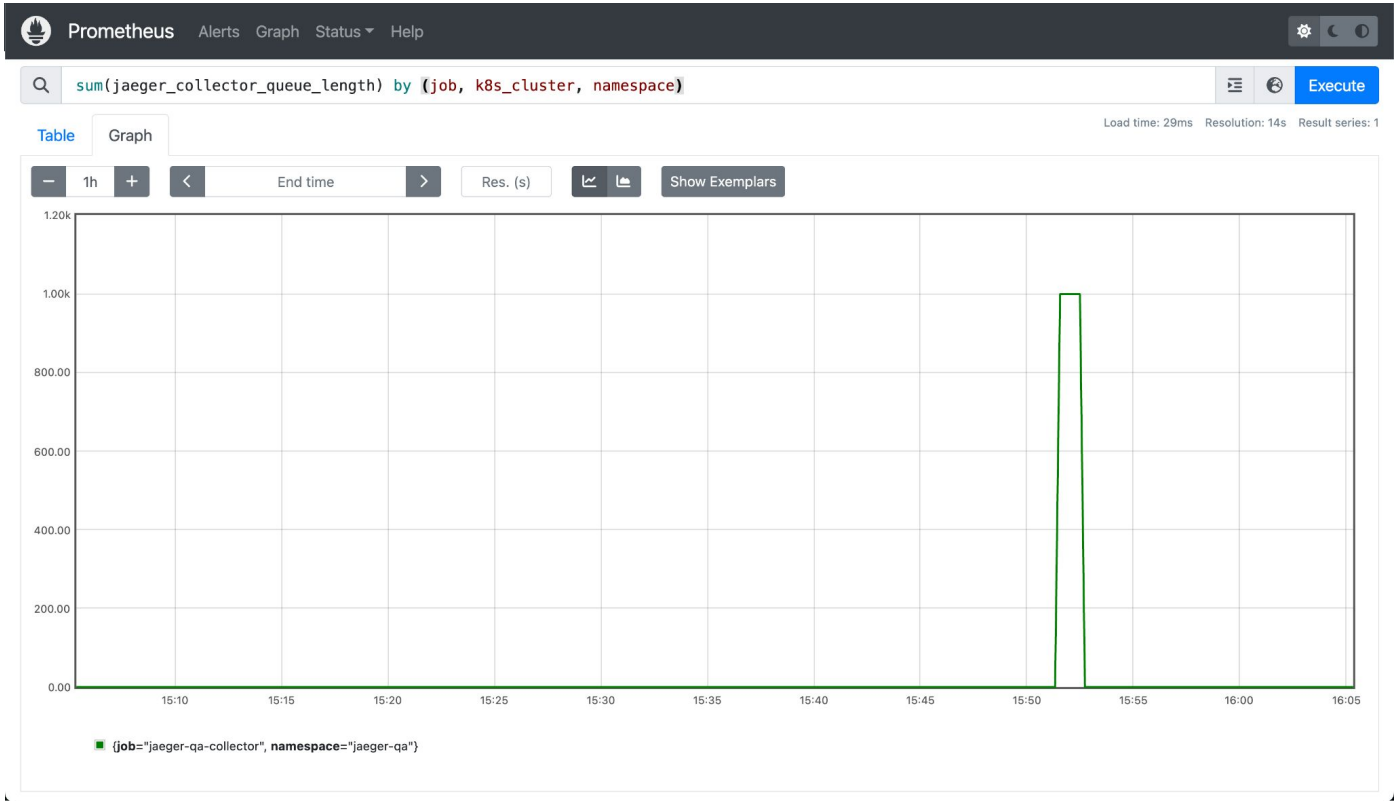
For each combination, Item is generated and tracked in Zabbix

Fresh data is added with each query from the previous step

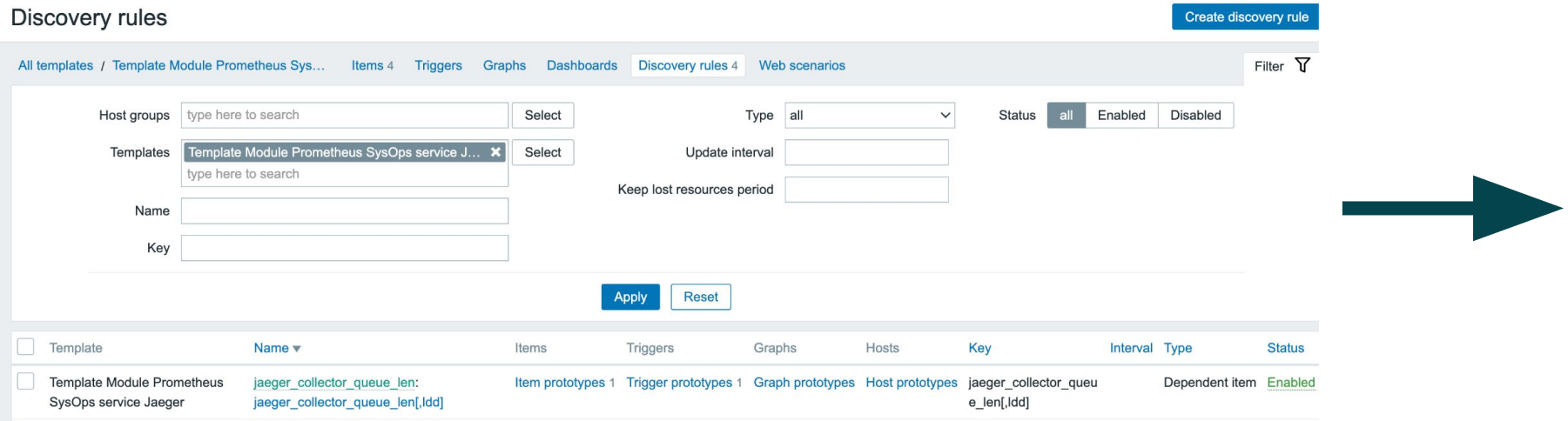
Example


Prometheus
Alerts
Graph
Status
Help
jaeger_collector_queue_length
Execute
Table
Graph
Load time: 251ms
Resolution: 14s
Result series: 3
1h
End time
Res. (s)
Show Exemplars
1.20k
1.00k
800.00
600.00
400.00
200.00
0.00
15:10
15:15
15:20
15:25
15:30
15:35
15:40
15:45
15:50
15:55
16:00
16:05
jaeger_collector_queue_length{container="jaeger-qa-collector", endpoint="admin", host="jaeger-qa-collector-6f7484b468-8z7p9", instance="10.70.200.18:14269", job="jaeger-qa-collector", namespace="jaeger-qa", pod="jaeger-qa-collector-6f7484b468-8z7p9", service="jaeger-qa-collector"}
jaeger_collector_queue_length{container="jaeger-qa-collector", endpoint="admin", host="jaeger-qa-collector-6f7484b468-jpx6t", instance="10.70.206.136:14269", job="jaeger-qa-collector", namespace="jaeger-qa", pod="jaeger-qa-collector-6f7484b468-jpx6t", service="jaeger-qa-collector"}
jaeger_collector_queue_length{container="jaeger-qa-collector", endpoint="admin", host="jaeger-qa-collector-6f7484b468-rtntx", instance="10.70.203.21:14269", job="jaeger-qa-collector", namespace="jaeger-qa", pod="jaeger-qa-collector-6f7484b468-rtntx", service="jaeger-qa-collector"}
Click: select series, CMD + click: toggle multiple series

Example


Prometheus
Alerts
Graph
Status
Help
sum(jaeger_collector_queue_length) by (job, k8s_cluster, namespace)
Execute
Table
Graph
Load time: 29ms
Resolution: 14s
Result series: 1
1h
End time
Res. (s)
Show Exemplars
1.20k
1.00k
800.00
600.00
400.00
200.00
0.00
15:10
15:15
15:20
15:25
15:30
15:35
15:40
15:45
15:50
15:55
16:00
16:05
{job="jaeger-qa-collector", namespace="jaeger-qa"}

Example



Example


Service Jaeger Prod: collector gce-infra-gke-us-w1-01 jaeger-prod has large queue, possible stora…
View as
Graph
Zoom out
Last 1 hour
From
now-1h
To
now
Apply
Last 2 days
Last 7 days
Last 30 days
Last 3 months
Last 6 months
Last 1 year
Last 2 years
Yesterday
Day before yesterday
This day last week
Previous week
Previous month
Previous year
Today
Today so far
This week
This week so far
This month
This month so far
This year
This year so far
Last 5 minutes
Last 15 minutes
Last 30 minutes
Last 1 hour
Last 3 hours
Last 6 hours
Last 12 hours
Last 1 day
Service Jaeger Prod: collector gce-infra-gke-us-w1-01 jaeger-prod has large queue, possible storage problem
10 K
8 K
6 K
4 K
2 K
0
06-25 15:15
15:18
15:20
15:22
15:24
15:26
15:28
15:30
15:32
15:34
15:36
15:38
15:40
15:42
15:44
15:46
15:48
15:50
15:52
15:54
15:56
15:58
16:00
16:02
16:04
16:06
16:08
16:10
16:12
16:14
06-25 16:15
last
min
avg
max
collector gce-infra-gke-us-w1-01 jaeger-prod has large queue, possible storage problem
[all]
0
0
1.17 K
7.63 K
Trigger: collector gce-infra-gke-us-w1-01 jaeger-prod has large queue, possible storage problem
[> 40000]

What Do We Get?

- We can build any Prometheus / exporters / OpenTelemetry / whatever pipeline in front of Prometheus
- We get all important metrics from Prometheus to Zabbix
  - But don’t overload Zabbix
  - 0.001-0.1% of collected data are really important
- We have a single pane of glass and alert history in Zabbix
- We can reuse AlertManager rules and alerts (and other tools)
  - But still use Zabbix Macros logic for thresholds

```yaml
groups:
  - name: recording_rules
    rules:
      - record: elasticsearch_cluster_health_red
        expr: sum(elasticsearch_cluster_health_status{color="red"}) by (cluster)
  - name: alerting_rules
    rules:
      - alert: elasticsearch_cluster_health_red_simple
        expr: elasticsearch_cluster_health_red > 0
        annotations:
          description: "Elasticsearch cluster {{$labels.cluster}} is in RED state"
        labels:
          __zbx_priority: "AVERAGE"

prometheus:
  api:
    url: "http://victoria-metrics:8428/api/v1/query"
  labels_to_zabbix_macros:
    - pattern: '\{\{(?:\s*)\$value(?:\s*)\}\}'
      value: "{ITEM.VALUE1}"
    - pattern: '\{\{(?:\s*)\$labels\.(?P<label>[a-zA-Z0-9\_\-]*)(?:\s*)\}\}'
      value: "{#\\g<label>}"

zabbix:
  template: service_elasticsearch_cluster
  name: "Template Module Prometheus Elasticsearch Cluster"
  hosts:
    - host_name: elasticsearch-cluster-prod
      visible_name: "Service Elasticsearch Cluster PROD"
      host_groups: ["Prometheus pseudo hosts", "Production hosts"]
      link_templates: ["templ_module_promt_service_elasticsearch_cluster"]
      macros:
        - macro: "{$ES.CLUSTER.LLD.MATCHES}"
          value: "^prod-(us|eu)$"
  macros:
    - macro: "{$ES.THRESHOLD}"
      value: "1"
      description: "Elasticsearch cluster health threshold"

# Optional: Documentation for alerts
wiki:
  knowledgebase:
    alerts:
      alertings:
        "elasticsearch_cluster_health_red_simple":
          title: "Elasticsearch cluster in RED state"
          content: "See runbook for troubleshooting steps"
```

1. PromQL query definition (recording rule)
2. Zabbix trigger expression
3. Connection to Zabbix / VictoriaMetrics
4. Pseudo-host definition
5. Macros override ber Pseudo-host
6. (Optional) Documentation

Promabbix Tooling

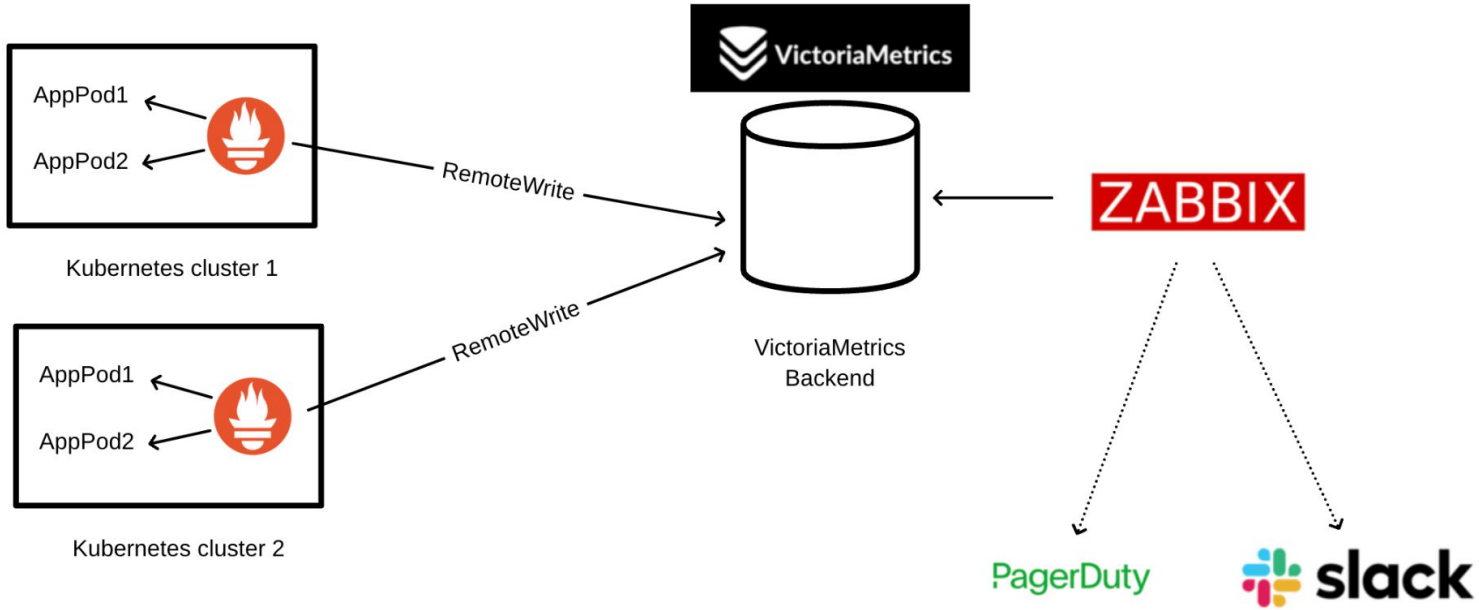
- promabbix.py - simple CLI tool
  - Input: alerting rules
  - Output: Zabbix template
- GitOps for Zabbix - open-sourcing in progress
  - Simple wrapper to manage most of Zabbix config through API from Git -> Zabbix instance (You can also click-ops in parallel, but…)
  - Pseudo-hosts definition using the template from CLI tool

Promabbix is Open-Source

- Tooling using via Ansible in Wrike since 2020
- Now rewritten to simple Python scripts for ease of use and separation from Wrike tooling (“v2.0”)

Promabbix in Wrike

Prometheus -> VictoriaMetrics -> Central Zabbix

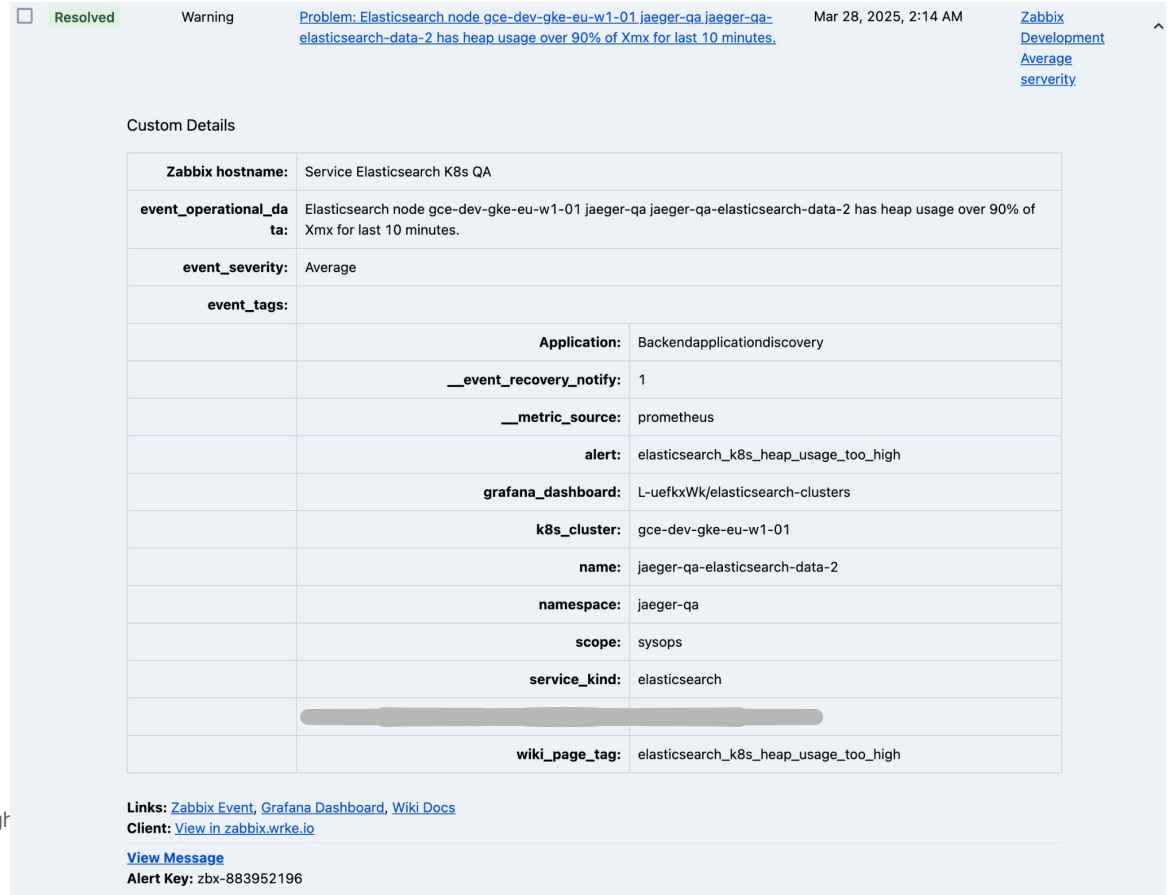


Scale of Promabbix Usage in Wrike

Definitions:

- 56 types of infrastructure services with 205 alert types on 117 pseudo-hosts
- 38 types of application services with 138 alert types on 41 pseudo-hosts

PagerDuty Integration (Infrastructure monitoring)



Resolved | Warning | Problem: Elasticsearch node gce-dev-gke-eu-w1-01 jaeger-qa jaeger-qa-elasticsearch-data-2 has heap usage over 90% of Xmx for last 10 minutes. | Mar 28, 2025, 2:14 AM | Zabbix Development Average serverity

Custom Details

Zabbix hostname:	Service Elasticsearch K8s QA
event_operational_data:	Elasticsearch node gce-dev-gke-eu-w1-01 jaeger-qa jaeger-qa-elasticsearch-data-2 has heap usage over 90% of Xmx for last 10 minutes.
event_severity:	Average
event_tags:	
Application:	Backendapplicationdiscovery
__event_recovery_notify:	1
__metric_source:	prometheus
alert:	elasticsearch_k8s_heap_usage_too_high
grafana_dashboard:	L-uefkxWk/elasticsearch-clusters
k8s_cluster:	gce-dev-gke-eu-w1-01
name:	jaeger-qa-elasticsearch-data-2
namespace:	jaeger-qa
scope:	sysops
service_kind:	elasticsearch
[redacted]	
wiki_page_tag:	elasticsearch_k8s_heap_usage_too_high

Links: Zabbix Event, Grafana Dashboard, Wiki Docs
Client: View in zabbix.wrke.io

View Message
Alert Key: zbx-883952196

PagerDuty Integration (Infrastructure monitoring)

Event details

Trigger details

Host	Service ArgoCD QA
Trigger	Application meta-gce-dev-gke-eu-w1-01 in gce-dev-gke-eu-w1-01 is OutOfSync.
Severity	Warning
Problem expression	**last**(/argocd-qa/argocd_unsynced_apps_count[,gce-dev-gke-eu-w1-01,meta-gce-dev-gke-eu-w1-01,default])>=1 and **nodata**(/argocd-qa/argocd_unsynced_apps_count[,gce-dev-gke-eu-w1-01,meta-gce-dev-gke-eu-w1-01,default],4m)=0
Recovery expression	
Event generation	Normal
Allow manual close	Yes
Enabled	Yes

Event details

Event	Application meta-gce-dev-gke-eu-w1-01 in gce-dev-gke-eu-w1-01 is OutOfSync.
Operational data	[redacted]
Severity	Warning
Time	2025-05-21 09:41:20
Acknowledged	No
Tags	alert: argocd_unsynce… Application: Backend … grafana_dashboard: a… •••
Description	Wiki: [redacted]

Actions

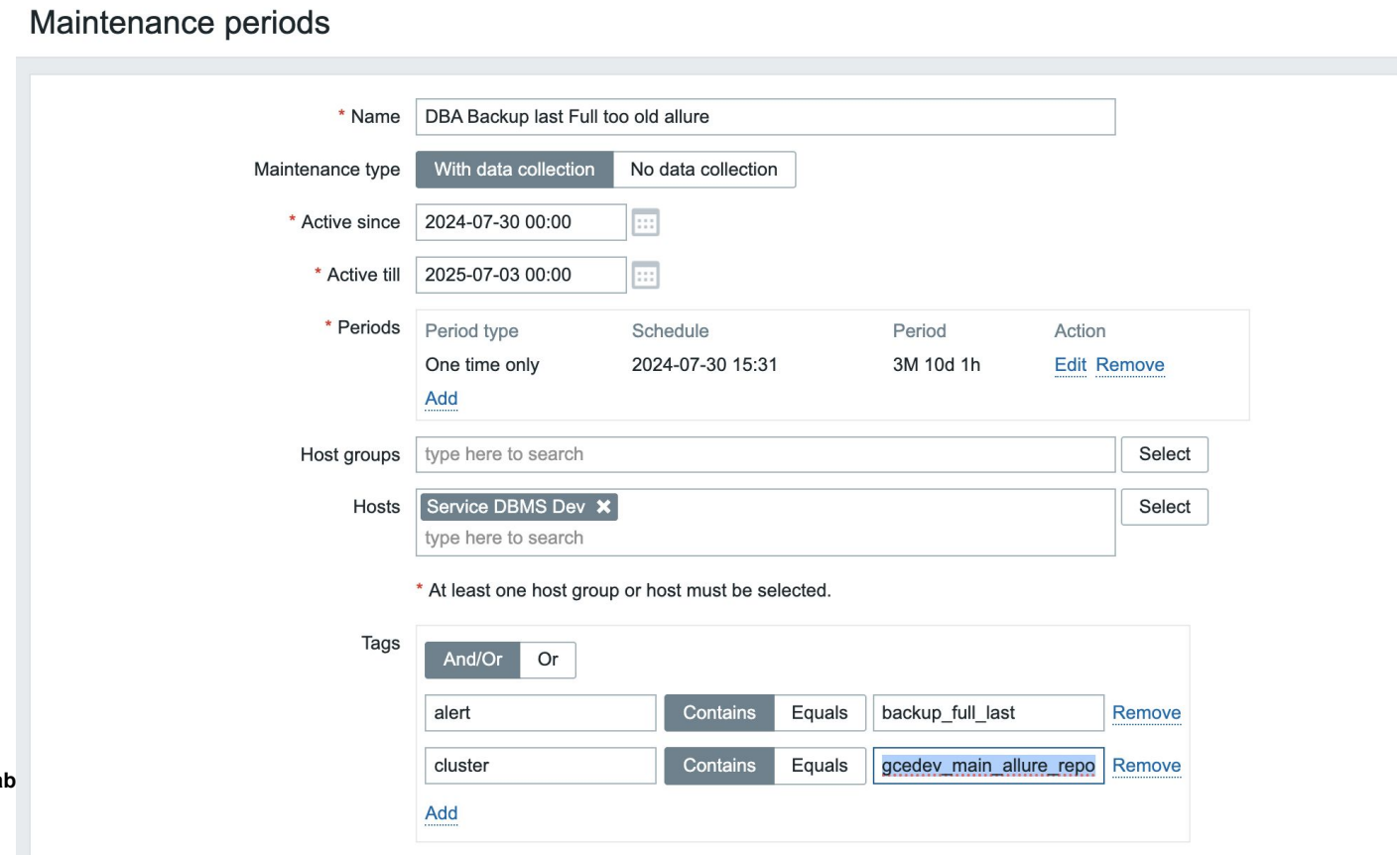
Step	Time	User/Recipient	Action	Message/Command	Status	Info
	2025-05-21 09:41:20					

Event list [previous 20]

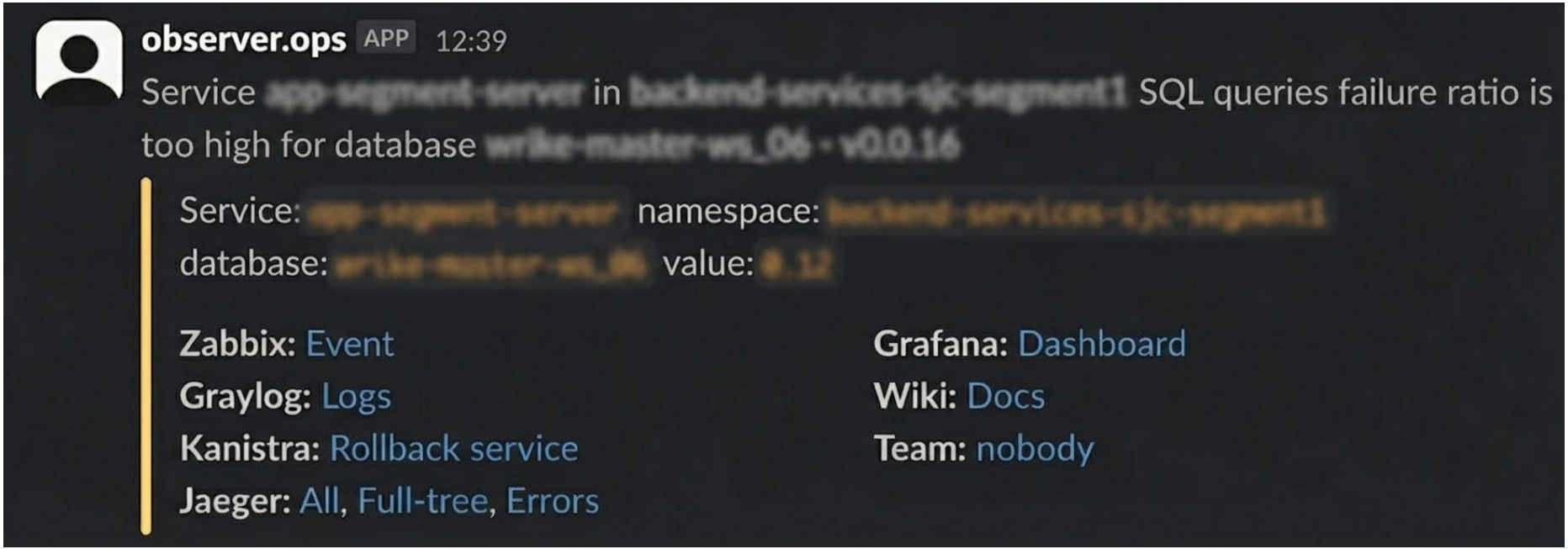
Time	Recovery time	Status	Age	Duration	Ack	Actions
2025-05-21 09:41:20		PROBLEM	2h 43m 44s	2h 43m 44s	No	

alert: argocd_unsynce… Application: Backend … grafana_dashboard: a… graylog_search_query… graylog_url_stream: gr…
k8s_cluster: gce-dev-g… name: meta-gce-dev-g… project: default scope: sysops slack_alarm_channel_… wiki_page_tag: argocd…
__event_delay: 4h __event_recovery_not… __metric_source: pro…

PagerDuty Integration (Infrastructure monitoring)



Slack Integration (APM)



Try Promabbix Yourself!

Q/A