云原生社区
Cloud Native Community

OBSERVABILITY
SUMMIT 2023
可观测性峰会 第 1 届

基于 Prometheus 的 SLO 告警实战

宋佳洋<KLLK

大纲

告警基础知识

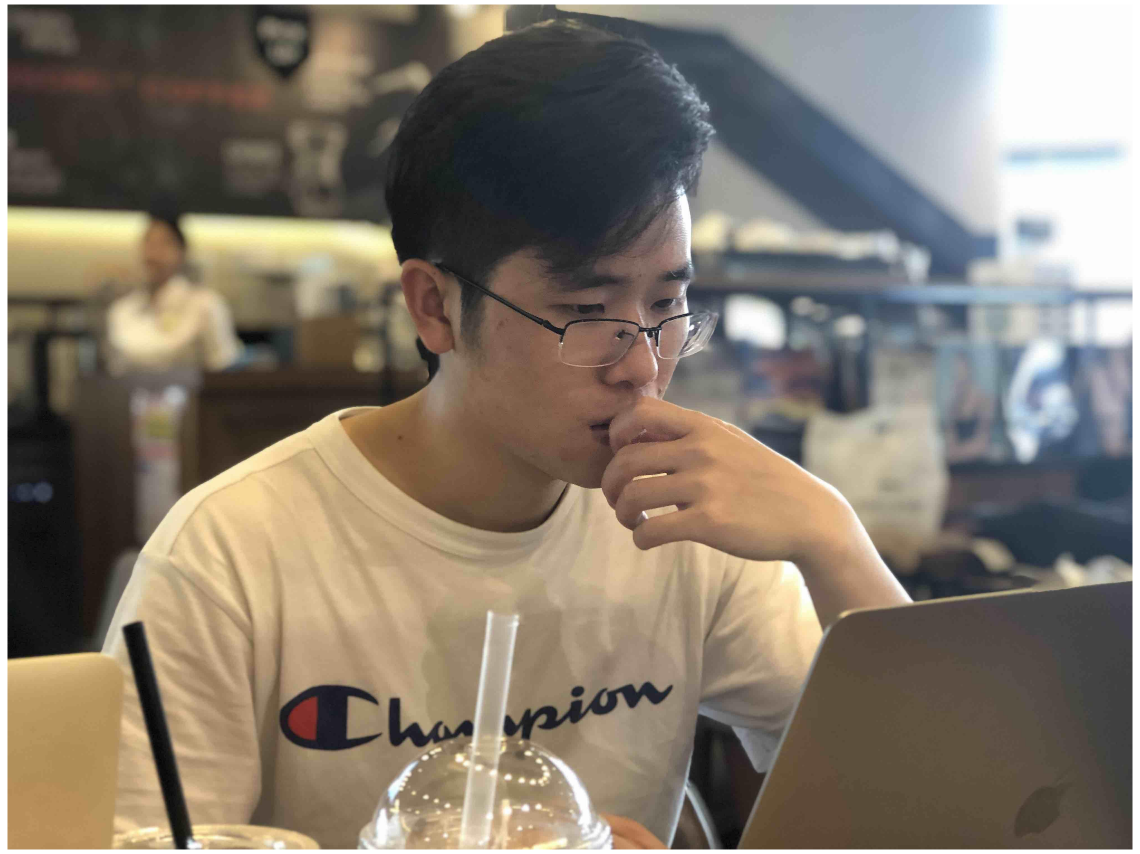
唤基于!ш/Х警项目构建

噬基于!л偶面!项目构建

唏多租户!щεE!服务构建

云原生社区
Cloud Native Community

关于我



先后就职于七牛云、京东云等公司，目前在! E л л E!从事云计算相关工作。

爱好开源，目前主要关注! ± X和!云原生可观测领域、是开源项目! [illegible]的代码贡献值。"

[illegible]

微信公众号："° [illegible]爱好者

为什么基于 SLO 告警

~~100%~~

- 梳理内容
- 优先级告警
- 利益方认可
- 持续迭代

没有 ɯɛE，就没有!ɯɲf ！

SLO 相关概念

SLI

- 状态码!埚
- 请求延迟!埚
- 进程运行非!嗍状态码退出

时间窗口

- Ik ꜝX
- Hꜝ Ikx
- HK

告警级别

- ꜝ Hb
- □ ·b□

错误预算

- 时间周期：
- İ »„ ：
- 错误预算：
- HK 天总请求数：
- 允许的错误请求数：

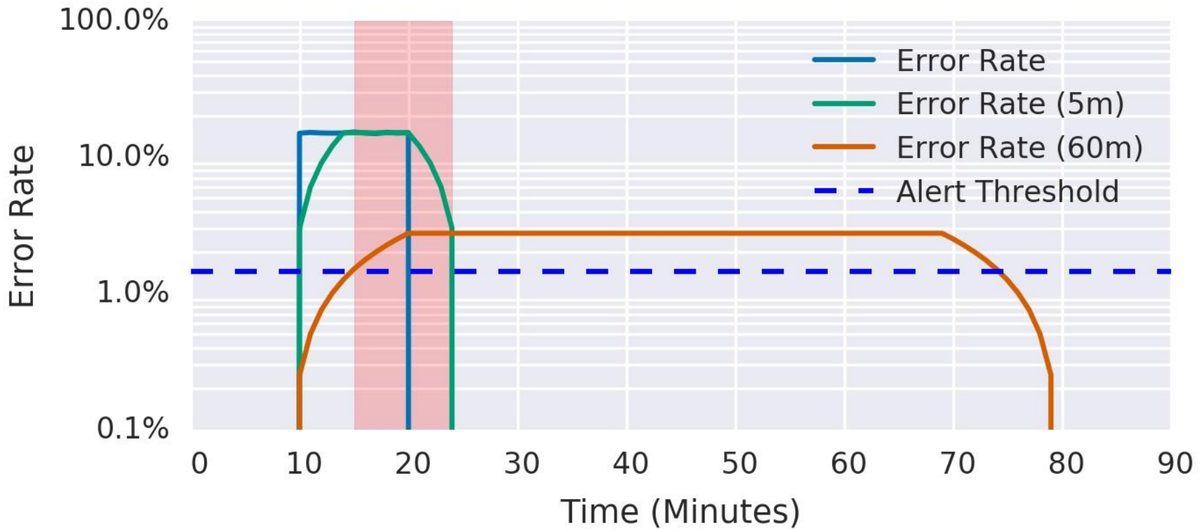
燃烧率

燃烧率	100%错误预算燃烧时间
	天
	天
	小时
	分钟

SLO 告警指导思想-MWMR

- ý↑ב, ר□ŷ□: 准确率
- ×ב, ⊓₩: 召回率（故障漏过未告警）
- \ בב, □ŷ□□ב: 投递延迟
- ×ב□□□ב: 告警重置时长

Severity	Long window	Short window	Burn rate	Error budget consumed



参考! ?

基于 Prometheus SLO 告警基础和挑战

开箱即用的 record and alert rule

[illegible]
[illegible]
[illegible]

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

加载和热更新

- 默认从本地文件加载。
- [illegible]
- [illegible]

Prometheus 告警基础

与时间窗口相关的多个 SLI rules

- [illegible]
- [illegible]

[illegible]

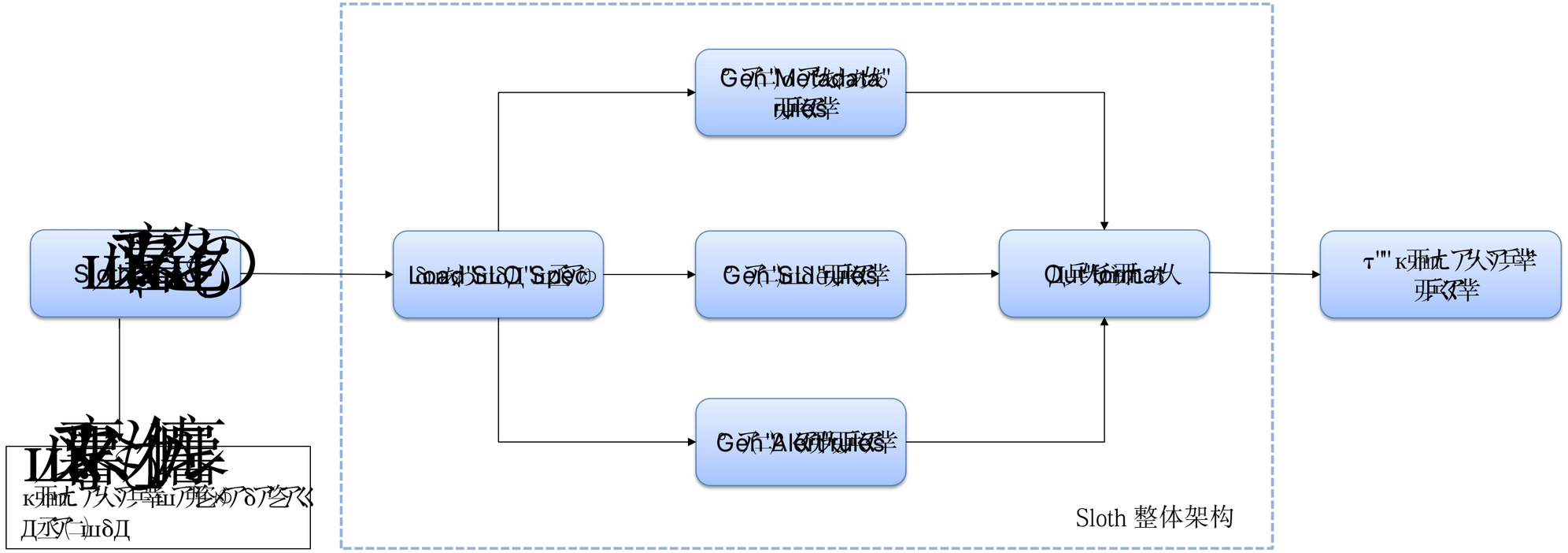
Alert rule 复杂, 需要考虑不同时间窗口和告警级别

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

Prometheus SLO 告警挑战

开源项目 sloth 简介

ıı/ 是一个简单易用的! ⊓ 自动生成器、支持!命令行和! 两种使用方式，支持自定义告警窗口配置、提供开箱即用的! 看板。



Sloth SLO 配置

```
version: "prometheus/v1"
service: "myservice"
labels:
  owner: "myteam"
  repo: "myorg/myservice"
  tier: "2"
slos:
  # We allow failing (5xx and 429) 1 request every
  - name: "requests-availability"
    objective: 99.9
    description: "Common SLO based on availability
    sli:
      events:
        error_query: sum(rate(http_request_duration_
        total_query: sum(rate(http_request_duration_
    alerting:
      name: MyServiceHighErrorRate
      labels:
        category: "availability"
      annotations:
        # Overwrite default Sloth SLO alert summmary
        summary: "High error rate on 'myservice' rec
      page_alert:
        labels:
          severity: pageteam
          routing_key: myteam
      ticket_alert:
        labels:
          severity: "slack"
          slack_channel: "#alerts-myteam"
```

以!碁组织，包含多个!碁

公用标签

通用标签

级别告警标签

级别告警标签

X啦碁U π] ！

```
apiVersion: sloth.slok.dev/v1
kind: PrometheusServiceLevel
metadata:
  name: sloth-slo-my-service
  namespace: monitoring
spec:
  service: "myservice"
  labels:
    owner: "myteam"
    repo: "myorg/myservice"
    tier: "2"
  slos:
    - name: "requests-availability"
      objective: 99.9
      description: "Common SLO based on availability
      sli:
        events:
          errorQuery: sum(rate(http_request_duration
          totalQuery: sum(rate(http_request_duration
      alerting:
        name: MyServiceHighErrorRate
        labels:
          category: "availability"
        annotations:
          summary: "High error rate on 'myservice' r
        pageAlert:
          labels:
            severity: pageteam
            routing_key: myteam
        ticketAlert:
          labels:
            severity: "slack"
            slack_channel: "#alerts-myteam"
```

E 2ら4! ш ε E

```
apiVersion: openslo/v1alpha
kind: SLO
metadata:
  name: sloth-slo-my-service
  displayName: Requests Availability
spec:
  service: my-service
  description: "Common SLO based on availability
  budgetingMethod: Occurrences
  objectives:
    - ratioMetrics:
        good:
          source: prometheus
          queryType: promql
          query: sum(rate(http_request_duration_s
        total:
          source: prometheus
          queryType: promql
          query: sum(rate(http_request_duration_s
      target: 0.999
  timeWindows:
    - count: 30
      unit: Day
```

Sloth AlertWindows 配置

```yaml
apiVersion: "sloth.slok.dev/v1"
kind: "AlertWindows"
spec:
  sloPeriod: 30d
  page:
    quick:
      errorBudgetPercent: 2
      shortWindow: 5m
      longWindow: 1h
    slow:
      errorBudgetPercent: 5
      shortWindow: 30m
      longWindow: 6h
  ticket:
    quick:
      errorBudgetPercent: 10
      shortWindow: 2h
      longWindow: 1d
    slow:
      errorBudgetPercent: 10
      shortWindow: 6h
      longWindow: 3d
```

时间周期

不同告警级别

不同错误预算燃烧率

多窗口

```yaml
apiVersion: sloth.slok.dev/v1
kind: AlertWindows
spec:
  sloPeriod: 7d
  page:
    quick:
      errorBudgetPercent: 8
      shortWindow: 5m
      longWindow: 1h
    slow:
      errorBudgetPercent: 12.5
      shortWindow: 30m
      longWindow: 6h
  ticket:
    quick:
      errorBudgetPercent: 20
      shortWindow: 2h
      longWindow: 1d
    slow:
      errorBudgetPercent: 42
      shortWindow: 6h
      longWindow: 3d
```

1 [illegible] 主要用于 [illegible] 告警配置，[illegible] 默认
包含了 [illegible] 和 [illegible] 两个配置，可自定义。

[illegible]

Sloth CLI

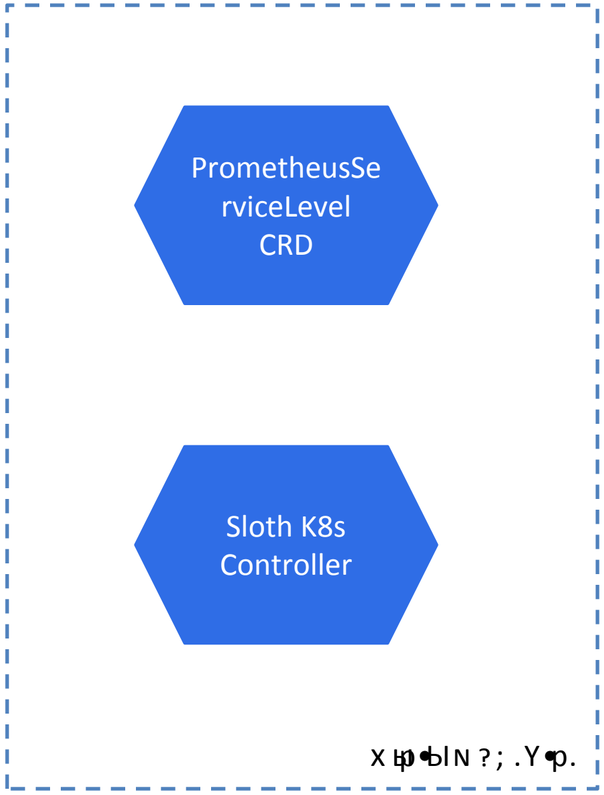
```
INFO[0000] SLI plugins loaded                            plugins=0 svc=storage.FileSLIPlugin version=v0.11.0 window=30d
INFO[0000] Using custom slo period windows catalog       svc=alert.WindowsRepo version=v0.11.0 window=30d
INFO[0000] SLO period windows loaded                     svc=alert.WindowsRepo version=v0.11.0 window=30d windows=2
INFO[0000] Generating from Prometheus spec               version=v0.11.0 window=30d
INFO[0000] Multiwindow-multiburn alerts generated        out=rules slo=myservice-requests-availability svc=generate.prometheu
s.Service version=v0.11.0 window=30d
INFO[0000] SLI recording rules generated                 out=rules rules=8 slo=myservice-requests-availability svc=generate.p
rometheus.Service version=v0.11.0 window=30d
INFO[0000] Metadata recording rules generated            out=rules rules=7 slo=myservice-requests-availability svc=generate.p
rometheus.Service version=v0.11.0 window=30d
INFO[0000] SLO alert rules generated                     out=rules rules=2 slo=myservice-requests-availability svc=generate.p
rometheus.Service version=v0.11.0 window=30d
INFO[0000] Prometheus rules written                      format=yaml groups=3 out=rules svc=storage.IOWriter version=v0.11.0
window=30d
```

```
INFO[0000] SLI plugins loaded                            plugins=0 svc=storage.FileSLIPlugin
INFO[0000] Using custom slo period windows catalog       svc=alert.WindowsRepo version=v0.11
INFO[0000] SLO period windows loaded                     svc=alert.WindowsRepo version=v0.11
error: "generate" command failed: invalid default slo period: window period 672h0m0s missing
```

- 支持单个文件和目录批量生成。
- □会覆盖默认!」
- 对应!」不存在，会报错

Sloth 与 K8s

Prometheus Operator



Ę 部署!彗又豪又ㄑ彳包豪缶

ヤ仇ヒも嘉」ㄑㄅ偶甲ㄆ
ㄑヤ關ㄐヒ包ㄟ喜彗關ㄑ彳關缶嚇又豪彳ㄍく嚇又嚇又豪關」ㄐ關
ㄑヤ關ㄐヒ包ㄟ喜彗關ㄑ彳關缶嚇又豪彳ㄍく嚇又嚇又豪關」ㄐ關

ヤ仇ヒも嘉」ㄑㄅ偶动
ㄑヤ關ㄐヒ包ㄟ喜彗關ㄑ彳關缶嚇又豪彳ㄍく嚇又嚇又豪關」ㄐ關
へㄑ又關ㄐヒ包ㄟ喜彗關側嚇又豪關くル

唎部署!彗又豪ɯεE

ヤ仇ヒも嘉」ㄑㄅ偶动
ㄑヤ關ㄐヒ包ㄟ喜彗關ㄑ彳關缶嚇又豪彳ㄍく嚇又嚇又豪關」ㄐ關
ㄐ角くㄑㄐ關ㄐ包ㄟ喜彗關ㄑ彳關缶嚇又豪彳ㄍく嚇又嚇又豪關」ㄐ關

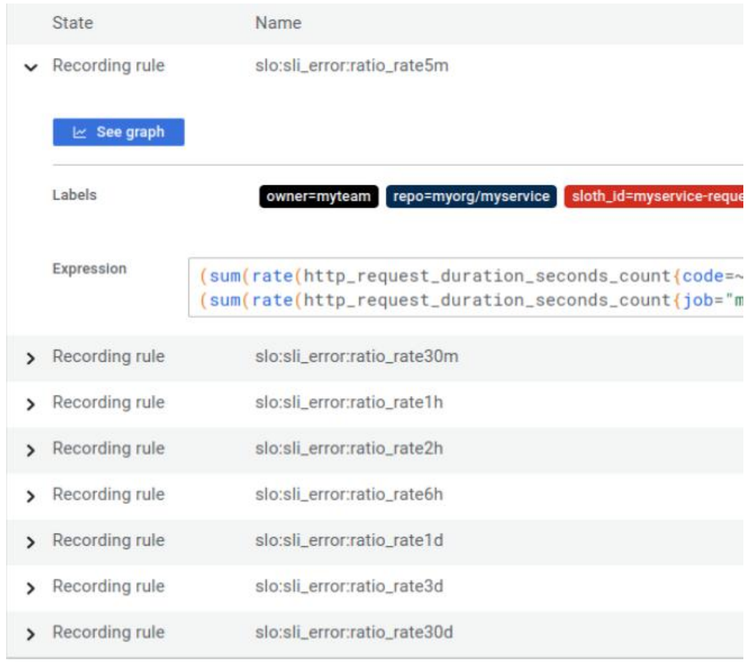
唎查看生成的!彗又甚和!ㄑ缶又くく豪ㄑ仍甚缶的ㄑ甚

ヤ仇ヒも嘉ㄐ!くく又イ暮缶ㄐヤ!ヰ嘉ㄑ又甚
ヤ仇ヒも嘉ㄐ!くく又イ暮缶ㄐヤ!ヰ嘉缶又くく豪ㄑ仍甚缶的ㄑ甚

生成 Prometheus rules 详解

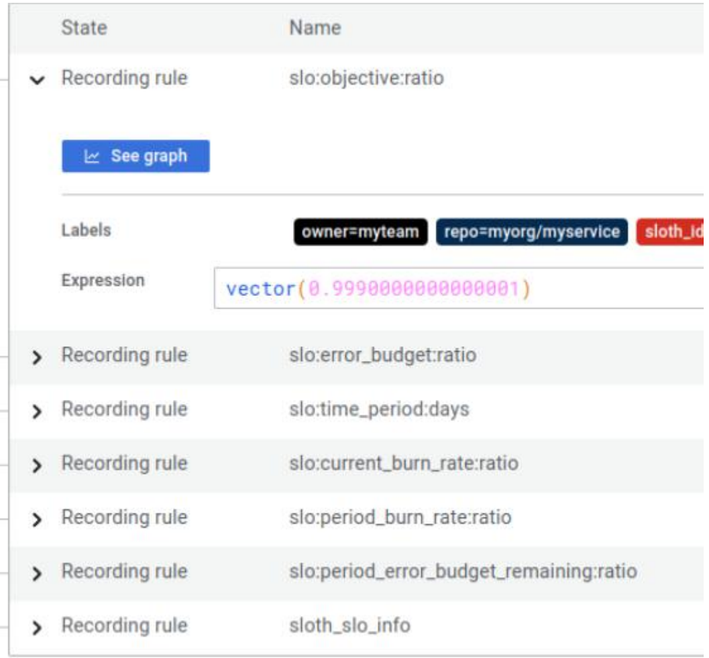
sli rules

- χ 个!!包の文ℓ⊂他ℓ甚
- □ぶ□ゃ□↑ぶ↓□ぶ↑□γ□ ぅHK□ 、信ヾ、修ヾ、徒ヾ倍值



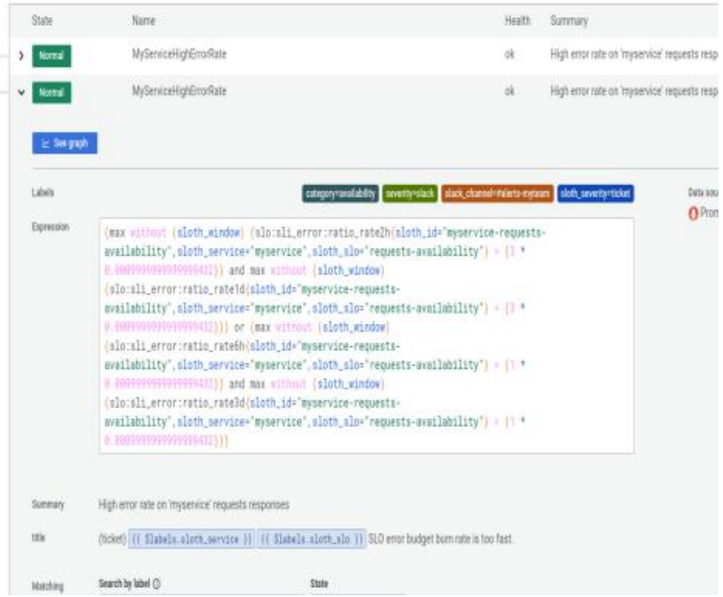
metadata rules

- ¥ 个!!包の文ℓ⊂他ℓ甚
- 包含了!шεE!目标、错误预算、时间周期等

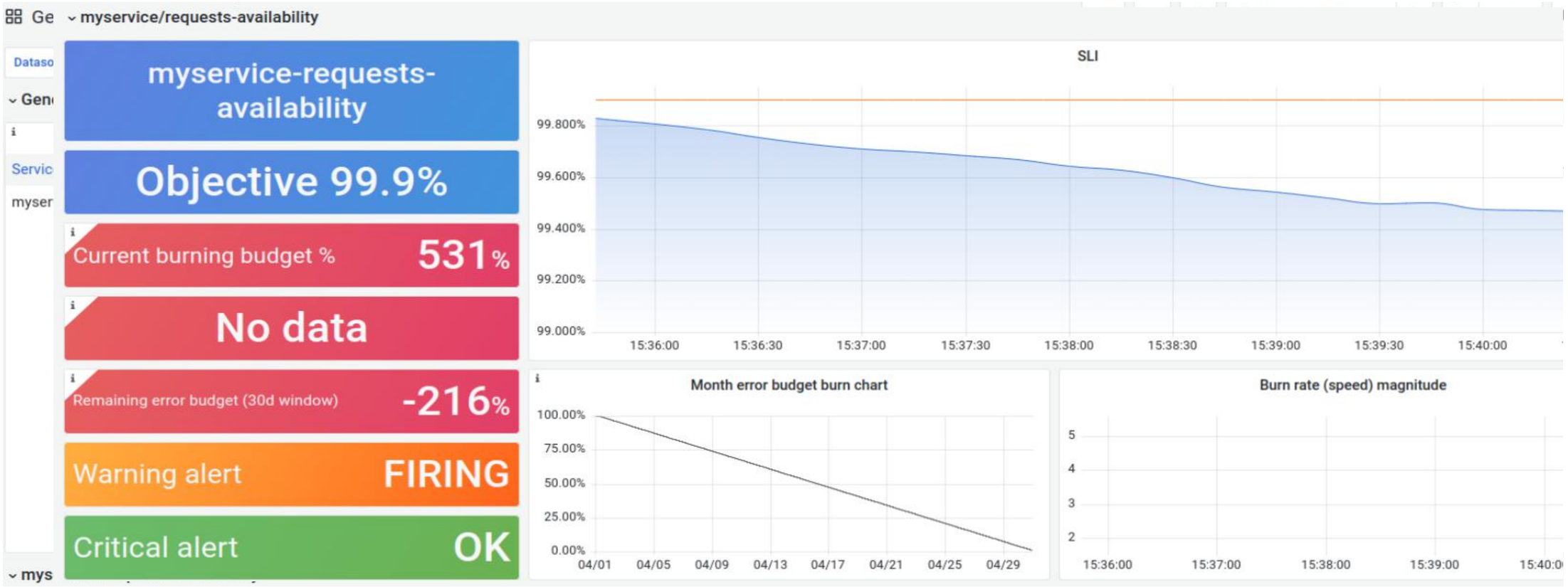


alert rules

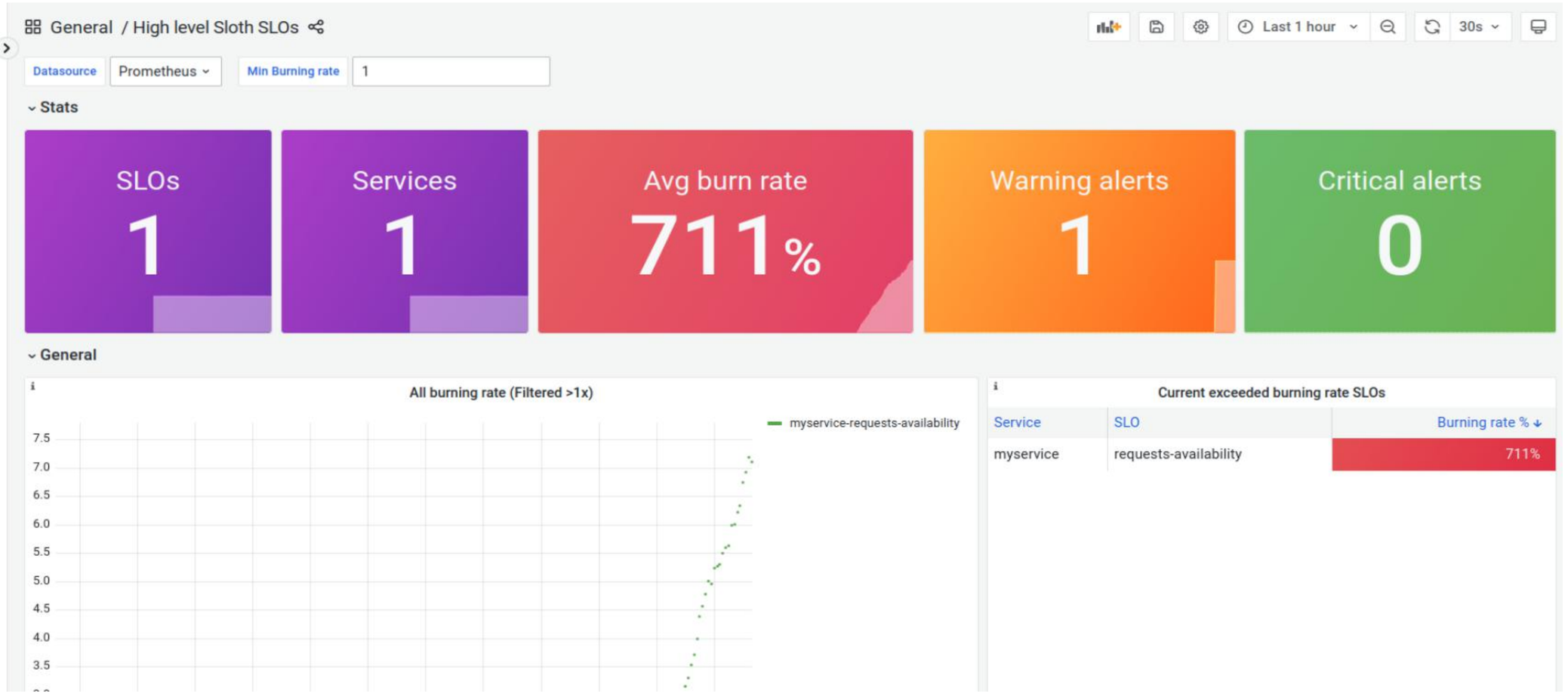
- κ 个!! 」ル 甚の甚ℓ甚
- 支持!π→π π



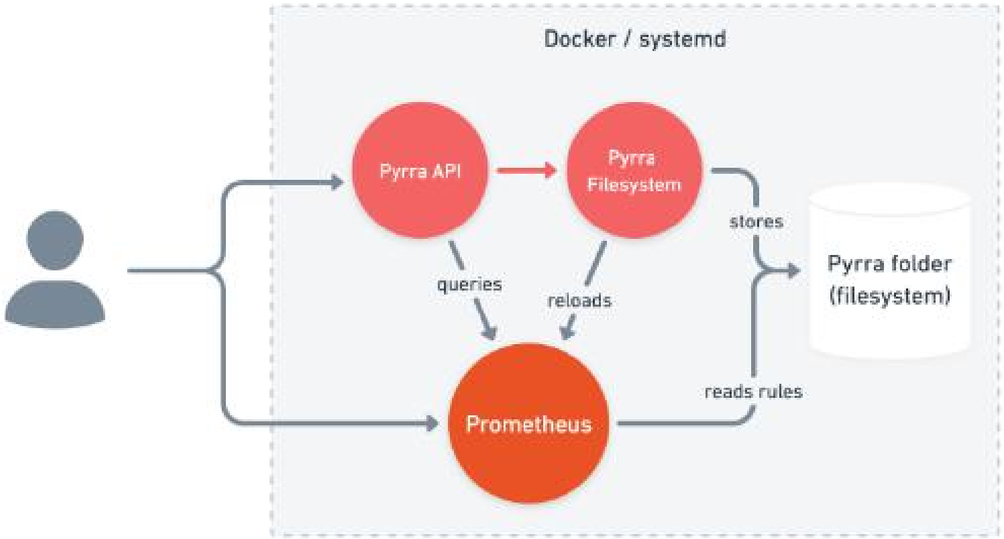
Sloth Dashboard-Details

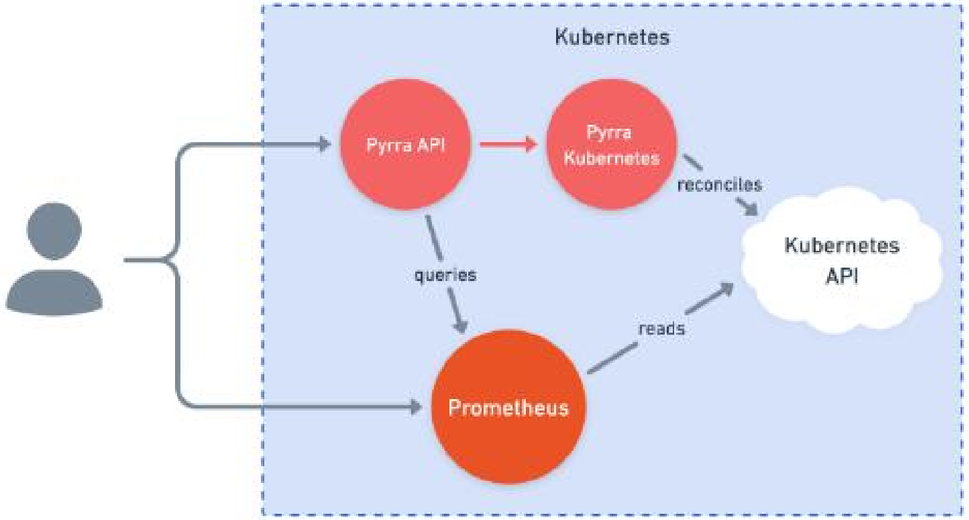


Sloth Dashboard-High Level



开源项目 Pyrra 介绍





Pyrra SLO 配置

```yaml
apiVersion: pyrra.dev/v1alpha1
kind: ServiceLevelObjective
metadata:
  name: pyrra-api-errors
  namespace: monitoring
  labels:
    prometheus: k8s
    role: alert-rules
    pyrra.dev/team: operations # Any labels prefixed with 'pyrra.dev/' will be propaga
spec:
  target: "99"
  window: 2w
  description: Pyrra's API requests and response errors over time grouped by route.
  indicator:
    ratio:
      errors:
        metric: http_requests_total{job="pyrra",code=~"5.."}
      total:
        metric: http_requests_total{job="pyrra"}
      grouping:
        - route
```

- 统一配置格式，无论是
- 一个
- 通过标签进行过滤，缺少
这层概念，按照配置文件名进行组织
- 不支持自定义告警窗口配置

云原生社区
Cloud Native Community

Filesystem 和 K8s 模式使用

主要使用!乏彻屿!ｸｶ基甼墓巜!命令

```
/ # pyrra filesystem --help
Usage: pyrra filesystem

Runs Pyrra's filesystem operator and backend for the API.

Flags:
  -h, --help                                   Show context-se
      --config-files="/etc/pyrra/*.yaml"       The folder wher
      --prometheus-url=http://localhost:9090   The URL to the
      --prometheus-folder="/etc/prometheus/pyrra/"  The folder wher
      --generic-rules                          Enabled generic
/ #
```

- 与!ｍ佨巜ㆫ豪ｼ們!…处于同一实例，…加载目录配置，生成!…能够确保被!ｍ佨巜ㆫ豪ｼ們…加载。
- 通过配置的!…进行!…让!ｍ佨巜ㆫ豪ｼ們重新加载配置。

Filesystem 模式

唻部署!乏彻屿!…

…

唻部署!乏彻屿!…

!!

…

唻查看生成的!…和!…

…

K8s 模式

Pyrra 生成 Prometheus rules 详解

单个[illegible]包括三个[illegible]告警分组，分别为[illegible]、[illegible]、[illegible]

[illegible]"包含[illegible][illegible]和[illegible]

Recording rule	http_request_duration_seconds:burnrate5m
Recording rule	http_request_duration_seconds:burnrate32m
Recording rule	http_request_duration_seconds:burnrate1h4m
Recording rule	http_request_duration_seconds:burnrate2h9m
Recording rule	http_request_duration_seconds:burnrate6h26m
Recording rule	http_request_duration_seconds:burnrate1d1h43m
Recording rule	http_request_duration_seconds:burnrate4d6h51m
Normal	ErrorBudgetBurn
Normal	ErrorBudgetBurn
Normal	ErrorBudgetBurn
Normal	ErrorBudgetBurn

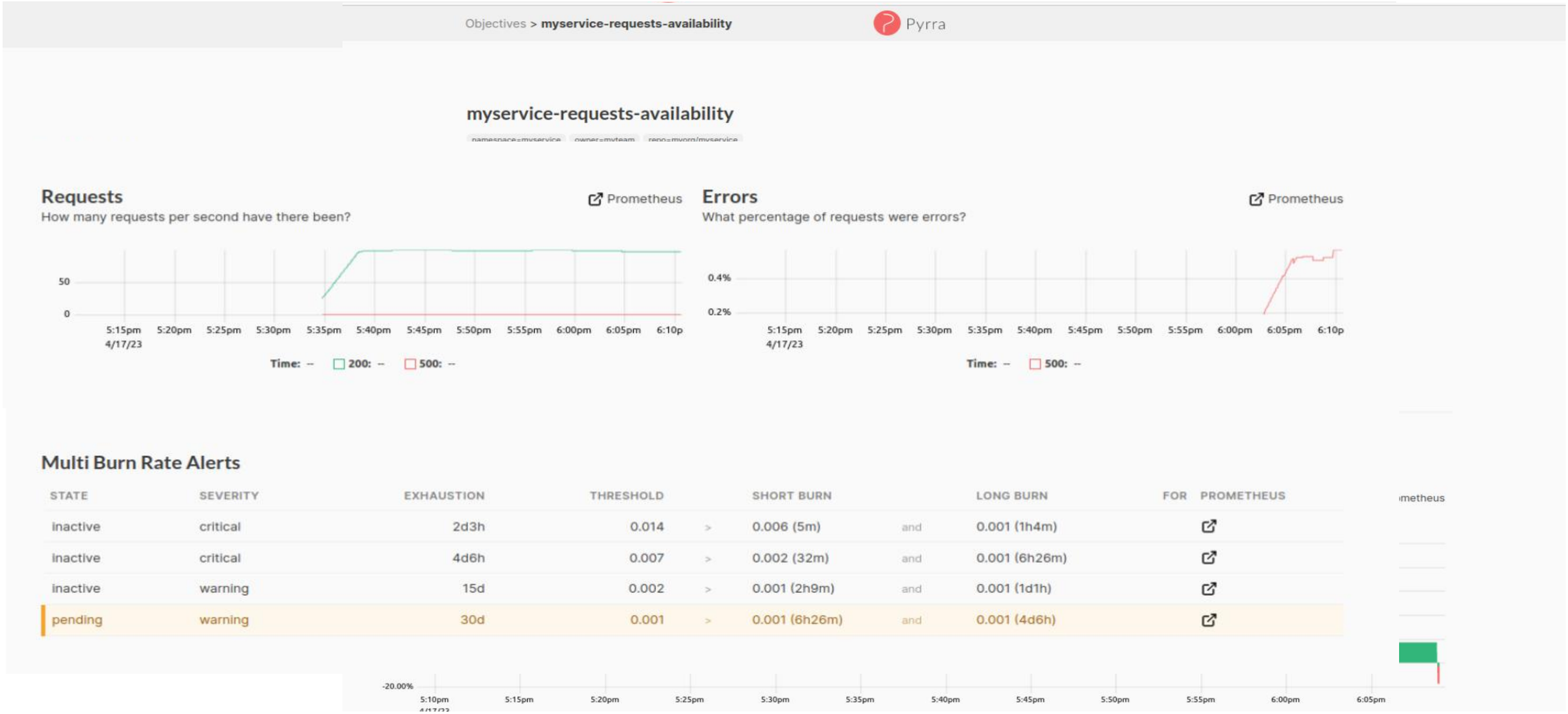
[illegible]："[illegible]"主要用于看板的[illegible]汇总统计（"og^"）。

State	Name
Recording rule	pyrra_objective
Recording rule	pyrra_window
Recording rule	pyrra_availability
Recording rule	pyrra_requests_total
Recording rule	pyrra_errors_total

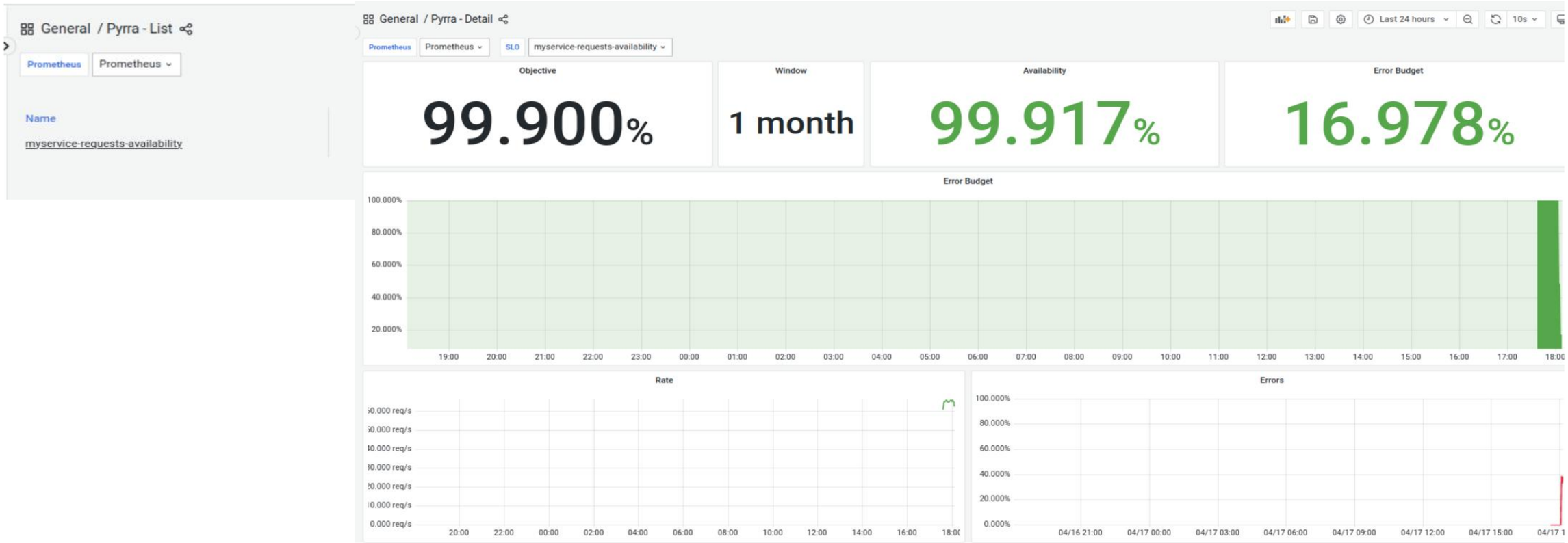
[illegible]："[illegible]和[illegible]"主要用于统计整个窗口周期总请求数和[illegible]告警。

Recording rule	http_request_duration_seconds:increase30d
Normal	SLOMetricAbsent

Pyrra Dashboard-API



Pyrra Dashboard-Grafana



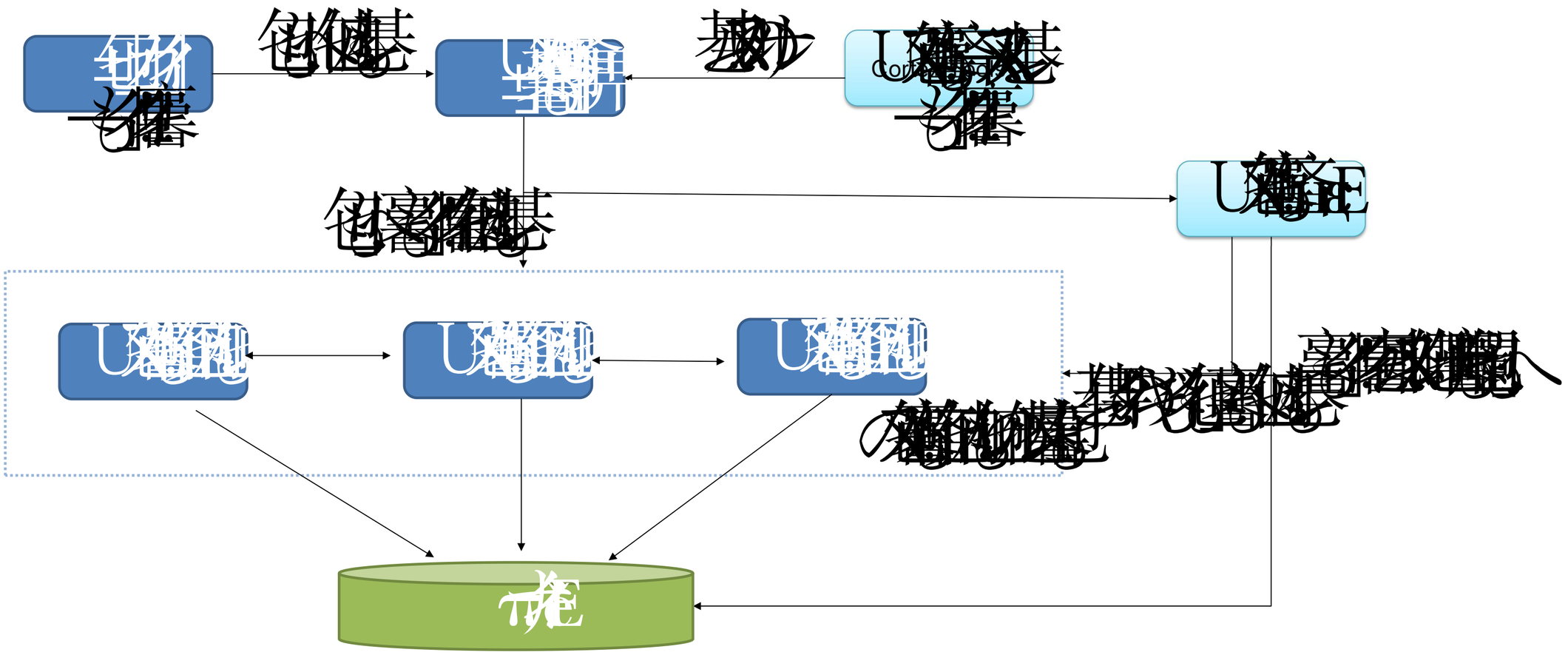
Sloth Vs Pyrra

	Sloth	Pyrra
	支持	支持
	不支持	支持
	支持	不支持
	支持	不支持
可读性	高	一般
		、

简单总结：

- 两个都是优秀的！
- 因为！

基于 Cortex 的多租户 SLO 服务构建

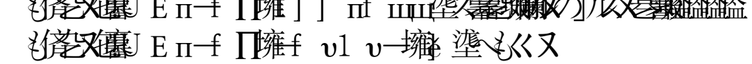


Cortex-tools 扩展

统一封装到!基于子命令

```
$ ./cortextool slos --help
usage: cortextool slos [<flags>] <command> [<args> ...]

View & edit slos stored in cortex.

Flags:
  --help          Show context-sensitive help (also try --help
  --authToken=""  Authentication token for bearer token or JWT

Subcommands:
  slos list [<flags>]
    List the slos currently in the telnant.

  slos load [<flags>] <files>...
    Load a set of slos to a designated cortex endpoint.

  slos get [<flags>] <service>
    Retrieve the sevice slos from the cortex.

  slos delete [<flags>] <service>
    Remove the sevice slos from the cortex.

  slos list-windows [<flags>]
    List slo windows currently in the telnant.

  slos load-windows [<flags>] <files>...
    Load a set of windows to a designated cortex endpoint.

  slos get-windows [<flags>] [<windows>]
    Get windows to from remote store.

  slos delete-windows [<flags>] [<windows>]
    Get windows to from remote store.
```

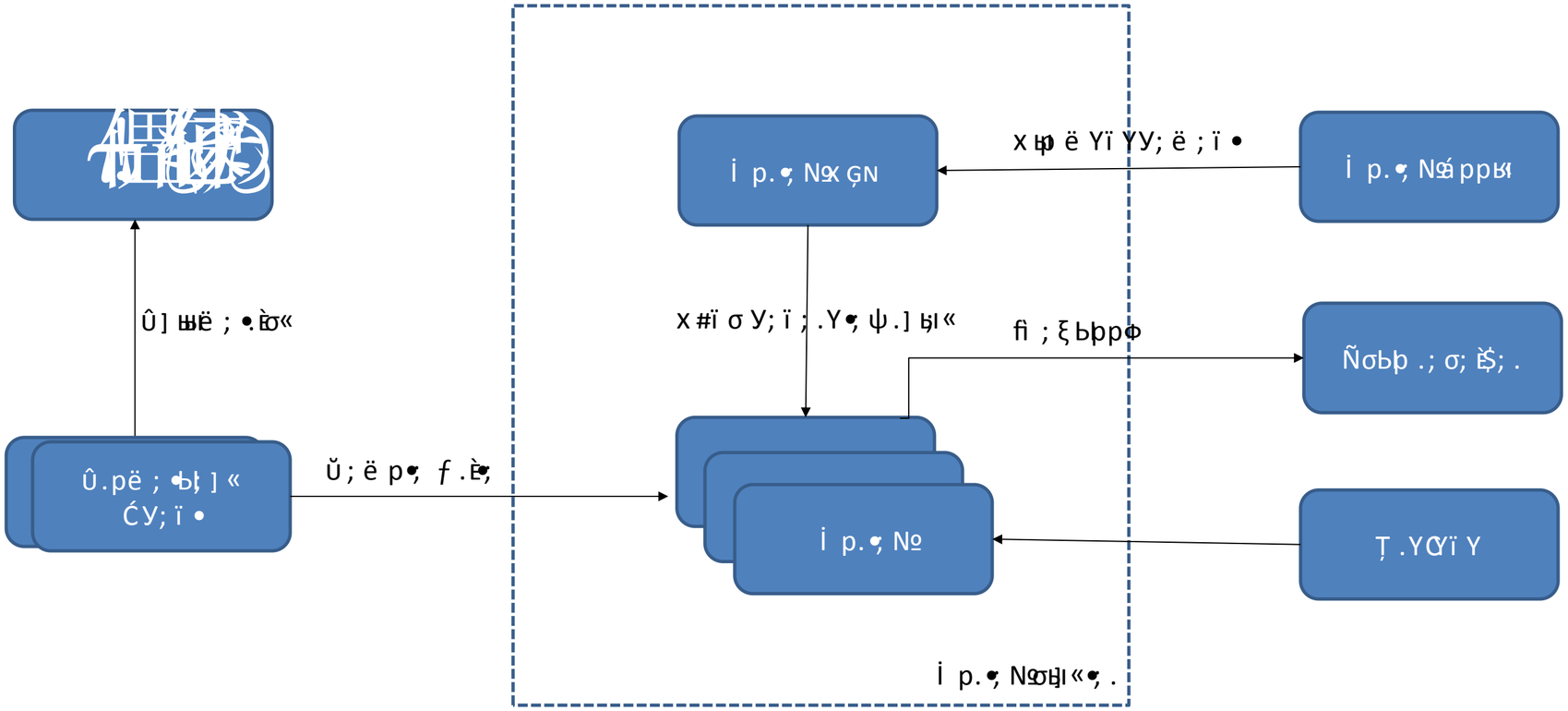
咔配置!的地址以及租户信息



咔寻人!和查询!御

```
11:44 $ cortextool slos load-windows ./config/slos/windows/*
INFO[0000] 7d.yaml windows loaded
INFO[0000] google-30d.yaml windows loaded
✔ /home/service/workspace/tower/play-with-cortex-slo [main|✚ 2…5]
$ cortextool slos list-windows

INFO[0000] Windows:
7d
google-30d
✔ /home/service/workspace/tower/play-with-cortex-slo [main|✚ 2…5]
$ cortextool slos load ./config/slos/*.yml --windows google-30d
INFO[0000] myservice.yml slos loaded with google-30d alert windows
✔ /home/service/workspace/tower/play-with-cortex-slo [main|✚ 2…5]
$ cortextool slos list

INFO[0000] Slos:
myservice
✔ /home/service/workspace/tower/play-with-cortex-slo [main|✚ 2…5]
$ cortextool slos get myservice

INFO[0000] myservice Slos:

version: "prometheus/v1"
service: "myservice"
labels:
  owner: "myteam"
  repo: "myorg/myservice"
slos:
```

Demo with docker-compose



参考!±-喜仰ゞ仓库!

资料

- Ñ ȷ ȷ ƛ İ × e ɾ ȷ ٠ Γ ȷ ȷ ٠ ä □ □ ٿ ב ל ȷ ȷ ƛ ت ȷ ٠ Γ ȷ ȷ ٠ ت ⊓ ⊓ □ ڀ ز خ ت ڀ □ ٨ ȷ □
- İ ٨ ȷ ä Ñ ר ç ¦ Γ ä □ □ ٿ ב ת ¦ Γ ɾ ȷ □ ت ٨ ȷ ٠ ت ٨ ȷ ä
- ý ث ⊓ Ñ ר ç ¦ Γ ä □ □ ٿ ב ת ¦ Γ ɾ ȷ □ ت ث ⊓ خ ב خ ت ث ⊓
- ý ٨ ⊔ ص ר ä □ ٨ ȷ ä ä □ □ ٿ ב ת ¦ Γ ɾ ȷ □ ت ٨ ⊓ ⊓ ⊔ ⊓ ⊔ پ ت ٨ ⊓ ض ר ä ב خ ٨ ȷ ä
- ý ٨ ⊔ ص ר ä ¦ ث ⊓ ä □ □ ٿ ב ת ¦ Γ ɾ ȷ □ ت ٨ ⊓ ⊓ ⊔ ⊓ ⊔ پ ت ٨ ⊓ ض ר ä ב خ ث ⊓

云原生社区
Cloud Native Community

OBSERVABILITY
SUMMIT 2023
可观测性峰会 第1届

Thank you

关注我们获取更多云原生资讯