

SRE 4 (03.12.2021) | Теория SRE | Часть 4: SLO alerts

Для каждого из наших SLO (availability, latency) сделаем по два алерта:

- warning - чувствительный к изменениям, но шумный а следовательно интересен только для ретроспективного анализа.
- page - дольше реагирует на изменения, но менее шумный, а значит подходящий для уведомлений.

Recording rules

Для начала, создадим два основных recording rule. Для этого, создадим объект PrometheusRule. Можно это сделать через файл `rules.yaml` положив туда следующий контент:

Замечание: Важно заменить `$city` на имя вашего города

```
kind: PrometheusRule
apiVersion: monitoring.coreos.com/v1
metadata:
  name: prometheus-slo-rules
  namespace: $city
spec:
  groups:
  - name: rules-slo
    rules:
    - record: $city:job:slo_availability:rate5m
      expr: |
        100
        * sum(rate(http_server_requests_total{app="provider_backend", city="$city", code!~"5.."}[5m]))
```

```
    / sum(rate(http_server_requests_total{app="provider_backend", city="$city"}[5m]))

- record: $city:job:slo_latency:rate5m
  expr: |
    100
    * sum(rate(http_server_request_duration_seconds_bucket{app="provider_backend", city="$city", le="1.0"}[5m]))
    / sum(rate(http_server_request_duration_seconds_count{app="provider_backend", city="$city"}[5m]))
```

Применяем конфигурацию в кластере:

```
kubectl apply -f p8s-rules.yaml
```

Убеждаемся, что prometheus подхватил изменения (процесс может занять до нескольких минут). В закладке Status -> Rules должны появиться созданные recording rules.

prometheus.sre.slurm.io/rules

☆

Prometheus

Alerts

Graph

Status ▾

Help

Rules

rules-slos

Runtime & Build Information

Command-Line Flags

Configuration

Rules

Targets

Service Discovery

| Rule | State | Error | Last Evaluation | Evaluation Time |
|--|-------|-------|-----------------|-----------------|
| <div>record: <code>kemerovo:job:slo_availability:rate5m</code> expr: <code>100 * sum(rate(http_server_requests_total{app="provider_backend",city="kemerovo",code="200"}[5m])) / sum(rate(http_server_requests_total{app="provider_backend",city="kemerovo"}[5m]))</code></div> | OK | | 10.238s ago | 974.7us |
| <div>record: <code>kemerovo:job:slo_latency:rate5m</code> expr: <code>100 * sum(rate(http_server_request_duration_seconds_bucket{app="provider_backend",city="kemerovo",le="1.0"}[5m])) / sum(rate(http_server_request_duration_seconds_count{app="provider_backend",city="kemerovo"}[5m]))</code></div> | OK | | 10.237s ago | 757.9us |

Алерты

После создания recording rule, можно непосредственно приступить к определению алертов. Создается подобным образом. Создаем файл `p8s-alerts.yaml`

Замечание: Не забываем про `$city`

```
kind: PrometheusRule
apiVersion: monitoring.coreos.com/v1
metadata:
  name: prometheus-slo-alerts
  namespace: $city
spec:
```

```
groups:
- name: alerts-slo
  rules:
    - alert: $cityAvailabilitySLOPage
      expr: $city:job:slo_availability:rate5m < 99
      for: 5m
      labels:
        severity: page
        team: $city
      annotations:
        summary: "Availability SLO is low"
        description: "Availability SLO is below target 99% for already 5 minutes (current value: {{ printf \"%.2f\" $value }}%)"

    - alert: $cityAvailabilitySLOWarning
      expr: $city:job:slo_availability:rate5m < 99.5
      for: 1m
      labels:
        severity: warning
        team: $city
      annotations:
        summary: "Availability SLO is low"
        description: "Availability SLO is approaching target 99% (current value: {{ printf \"%.2f\" $value }}%)"

    - alert: $cityLatencySLOPage
      expr: $city:job:slo_latency:rate5m < 99
      for: 5m
      labels:
        severity: page
        team: $city
      annotations:
        summary: "Latency SLO is low"
```

```
    description: "Latency SLO is below target 99% for already 5 minutes (current value: {{ printf \"%.2f\" $value }})"

- alert: $cityLatencySLOWarning
  expr: $city:job:slo_latency:rate5m < 99.5
  for: 1m
  labels:
    severity: warning
    team: $city
  annotations:
    summary: "Latency SLO is low"
    description: "Latency SLO is approaching target 99% (current value: {{ printf \"%.2f\" $value }}%)"
```

Тут мы создали по два алерта для availability и latency SLO.

Применяем конфигурацию в кластере:

```
kubectl apply -f p8s-alerts.yaml
```

После этого, в prometheus в закладке Alerts должны появиться четыре алерта.

Alerts

[🔍 Inactive \(4\)](#) [🔍 Pending \(0\)](#) [🔍 Firing \(0\)](#)

☐ Show annotations

/etc/prometheus/rules/prometheus-students-prometheus-kube-p-prometheus-rulefiles-0/kemerovo-prometheus-slo-alerts.yaml > alerts-slo


KemerovoAvailabilitySLOPage (0 active)

KemerovoAvailabilitySLOWarning (0 active)

KemerovoLatencySLOPage (0 active)

KemerovoLatencySLOWarning (0 active)

Edited 2 hours ago

 snippetfile1.txt 15 bytes

1

see description

Write a comment or drag your files here...

[Markdown](#) is supported