

1.1 Настраиваем мониторинг: Инструкции для создания простого мониторинга

Подключаем датасурс <https://prometheus.sre.slurm.io>

1. Container CPU usage

```
sum(rate(container_cpu_usage_seconds_total{container!="POD",container!="", namespace="<CITY>"}[5m])) by (pod)
```

2. Container memory usage

```
sum(container_memory_usage_bytes{container!="POD",container!="", namespace="<CITY>"}) by (pod)
```

3. Pods running

```
sum(up{job="<CITY>"}) / count(up{job="<CITY>"})
```

4. Алерт на Pods running

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: prometheus-basic-rules
spec:
  groups:
```

```
- name: rules
  rules:
  - alert: TooManyDown
    expr: (sum(up{job="<CITY>"})) / count(up{job="<CITY>"})) <= 0.6
    for: 3m
    labels:
      team: <CITY>
    annotations:
      summary: "Too many pods down"
      description: "Ratio of running pods is too small (current value: {{ printf \"%.1f\" $value }})"
```

5. Requests

```
sum(rate(http_server_requests_total{app="provider_backend", namespace="<CITY>"}[5m])) by (code)
```

6. Latency

```
histogram_quantile(0.99, sum(rate(http_server_request_duration_seconds_bucket{app="provider_backend", namespace="<CITY>"}[5m])))
histogram_quantile(0.95, sum(rate(http_server_request_duration_seconds_bucket{app="provider_backend", namespace="<CITY>"}[5m])))
histogram_quantile(0.90, sum(rate(http_server_request_duration_seconds_bucket{app="provider_backend", namespace="<CITY>"}[5m])))
histogram_quantile(0.75, sum(rate(http_server_request_duration_seconds_bucket{app="provider_backend", namespace="<CITY>"}[5m])))
histogram_quantile(0.50, sum(rate(http_server_request_duration_seconds_bucket{app="provider_backend", namespace="<CITY>"}[5m])))
```

Edited 2 hours ago

 **foo.bar** 6 bytes

1	foobar
---	--------

Write a comment or drag your files here...

[Markdown](#) is supported