

Добавление правил уведомлений

Практика выполняется на monitoring сервере, доступ к нему осуществляется по IP адресу и выполняется с root привилегиями.

1. Сохраняем правила в файл: rules_alert.yml :

```
cat <<EOF > /etc/prometheus/rules_alert.yml
```

```
groups:
```

```
- name: PrometheusGroup
```

```
rules:
```

```
- alert: PrometheusConfigurationReload
```

```
  expr: prometheus_config_last_reload_successful != 1
```

```
  for: 1m
```

```
  labels:
```

```
    severity: critical
```

```
    service: prom
```

```
  annotations:
```

```
    summary: "Prometheus configuration reload (instance {{ \${labels.instance} }}"
```

```
    description: "Prometheus configuration reload error\n VALUE = {{ \${value} }}\n LABELS: {{ \${labels} }}"
```

```
- name: NodeExporterGroup
```

```
rules:
```

```
- alert: ExporterDown
```

```
  expr: up == 0
```

```
  for: 1m
```

```
  labels:
```

```
    severity: error
```

```
    service: prom
```

```
  annotations:
```

```
    summary: "Exporter down (instance {{ \${labels.instance} }}"
```

```
    description: "Prometheus exporter down\n VALUE = {{ \${value} }}\n LABELS: {{ \${labels} }}"
```

```
- alert: HighCpuLoad
```

```
  expr: 100 - (avg by(instance) (irate(node_cpu_seconds_total{mode="idle"}[5m])) * 100) > 80
```

```
  for: 5m
```

```
  labels:
```

```
    severity: warning
```

```
  annotations:
```

```
    summary: "High CPU load (instance {{ \${labels.instance} }}"
```

```
    description: "CPU load is > 80%\n VALUE = {{ \${value} }}\n LABELS: {{ \${labels} }}"
```

```
- alert: SystemdServiceCrashed
  expr: node_systemd_unit_state{state="failed"} == 1
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "SystemD service crashed (instance {{ \$labels.instance }})"
    description: "SystemD service crashed\n VALUE = {{ \$value }}\n LABELS: {{ \$labels }}"
EOF
```

2. Проверяем правила.

Для проверки правил также используется утилита `promtool`, отличие только в ключах запуска.

```
promtool check rules /etc/prometheus/rules_alert.yml
```

3. Добавляем данные о Rules в основной конфигурационный файл Prometheus.

Открываем на редактирование файл: `/etc/prometheus/prometheus.yml` и после `rule_files` добавляем `"- rules_*.yml"`. В результате у вас должно получиться:

```
rule_files:
- rules.yml
- rules_*.yml
```

4. Проверяем корректность внесенных изменений:

```
promtool check config /etc/prometheus/prometheus.yml
```

5. Выполните reload для Prometheus, чтобы применить новые настройки:

```
systemctl reload prometheus.service
```

NB! Для применения изменений в `prometheus.yml` достаточно выполнить `reload` для службы. Этот способ предпочтительней, так как в случае ошибки в конфигурационном файле просто не применяются изменения. Если же выполнить `restart`, то из-за ошибки весь Prometheus станет недоступен до устранения ошибки.

6. Проверяем, что правила появились.

Открываем UI Prometheus, раздел rules. Для этого открываем в браузере: <http://<адрес сервера monitoring>:9090/rules>. Должны отображаться правила:

NodeExporterGroup			718ms ago	996.3us
Rule	State	Error	Last Evaluation	Evaluation Time
<pre>alert: ExporterDown expr: up == 0 for: 5m labels: severity: warning annotations: description: - Prometheus exporter down VALUE = {{ \$value }} LABELS: {{ \$labels }} summary: Exporter down (instance {{ \$labels.instance }})</pre>	OK		718ms ago	721.5us

Теперь переходим: <http://адрес сервера monitoring:9090/alerts>. На этой странице отображается состояние всех правил.

Alerts

Show annotations

/etc/prometheus/rules_alert.yml > NodeExporterGroup

ExporterDown (0 active)

```
alert: ExporterDown
expr: up == 0
for: 5m
labels:
  severity: warning
annotations:
  description: |-
    Prometheus exporter down
    VALUE = {{ $value }}
    LABELS: {{ $labels }}
summary: Exporter down (instance {{ $labels.instance }})
```

HighCpuLoad (0 active)

SystemServiceCrashed (0 active)

/etc/prometheus/rules_alert.yml > PrometheusGroup

PrometheusConfigurationReload (0 active)

7. Тестирование работы правил.

Отключаем `node_exporter`, выполнив команду:

```
systemctl stop node_exporter.service
```

Возвращаемся на страницу: <http://<адрес сервера monitoring>:9090/alerts>. Одно из alert rule должно перейти в состояние PENDING

Alerts

Show annotations

/etc/prometheus/rules_alert.yml > NodeExporterGroup

ExporterDown (1 active)

```
alert: ExporterDown
expr: up == 0
for: 5m
labels:
  severity: warning
annotations:
  description: |-
    Prometheus exporter down
    VALUE = {{ $value }}
    LABELS: {{ $labels }}
summary: exporter down (instance {{ $labels.instance }})
```

Labels	State	Active Since	Value
alertname="ExporterDown" instance="localhost:9100" job="static_config" sd="static" severity="warning"	PENDING	2019-11-15 12:21:30.427863294 +0000 UTC	0

Annotations

description
Prometheus exporter down VALUE = 0 LABELS: map[_name__up instance:localhost:9100 job:static_config sd:static]

summary
Exporter down (instance localhost:9100)

По истечению времени, указанного в for(5m), alert rule перейдет в состояние: FIRING

Alerts

Show annotations

/etc/prometheus/rules_alert.yml > NodeExporterGroup

ExporterDown (1 active)

```
alert: ExporterDown
expr: up == 0
for: 5m
labels:
  severity: warning
annotations:
  description: |-
    Prometheus exporter down
    VALUE = {{ $value }}
    LABELS: {{ $labels }}
summary: exporter down (instance {{ $labels.instance }})
```

Labels	State	Active Since	Value
alertname="ExporterDown" instance="localhost:9100" job="static_config" sd="static" severity="warning"	FIRING	2019-11-15 12:21:30.427863294 +0000 UTC	0

Annotations

description
Prometheus exporter down VALUE = 0 LABELS: map[_name__up instance:localhost:9100 job:static_config sd:static]

summary
Exporter down (instance localhost:9100)

NB! Обратите внимание: Prom сам не обновляет страницы. Чтобы увидеть добавленные данные, необходимо обновить страницу (нажать F5).

NB! Установленная галочка: "Show annotations" позволяет просмотреть Annotations labels с уже подставленными в шаблон значениями.