

Домашнее задание (модуль 22)

22.6 Практическая работа

Цели практической работы

- Использовать знания о сборе диагностической информации в Linux.
- Самостоятельно изучить алерт-менеджер и синтаксис заведения алертов.
- Попрактиковаться в составлении выражений на promql.

Задание 1

Что нужно сделать

Напишите promql-выражение, которое покажет:

- свободное место на диске;
- свободную память (не забывайте о памяти, которая занята пейджкешем);
- loadaverage за минуту.

Что оценивается

Корректность выражений.

Как отправить задание на проверку

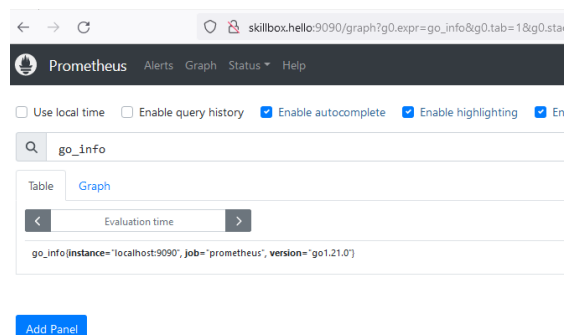
Пришлите через форму ниже скриншоты, на которых видны выражения и график, построенный по полученным значениям для каждого случая.

Ответ:

Установка prometheus

```
vgusev2007@skillbox-vgusev2007:~$ docker run -d --name prometheus-vgusev2007 -p 9090:9090 prom/prometheus
c9b5e07f3c6e0a64cabae85d44ee344ad97d071ff8a8c8afc42e55de86dc05d6
vgusev2007@skillbox-vgusev2007:~$
```

Программа работает:



```

vgusev2007@skillbox-vgusev2007:~$ docker run -d --name prometheus-node-exporter-vgusev2007 prom/node-exporter
Unable to find image 'prom/node-exporter:latest' locally
latest: Pulling from prom/node-exporter
d5c4df21b127: Pull complete
2f5f7d8898a1: Pull complete
2b6642e6c59e: Pull complete
Digest: sha256:81f94e50ea37a88dfce849d0f4acadb25b96b397061f59e5095905f6bc5829637
Status: Downloaded newer image for prom/node-exporter:latest
dc60ff83e34f74a686bbbf6e351e35bef25e6bb56b44da81658498d9190b81b
vgusev2007@skillbox-vgusev2007:~$

```

```
vgusev2007@skillbox-vgusev2007:~$ docker inspect --format='{{.NetworkSettings.IPAddress}}' prometheus-node-exporter-vgusev2007
172.17.0.6
```

```
vgusev2007@skillbox-vgusev2007:~$ curl -v http://172.17.0.6:9100 | head
* Trying 172.17.0.6:9100...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed

0         0      0     0    0      0     0      0  --:--:-- --:--:-- --:--:--    0* Connected to 172.17.0.6 (172.17.0.6) port 9100 (#0)
> GET / HTTP/1.1
> Host: 172.17.0.6:9100
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: text/html; charset=UTF-8
< Date: Fri, 22 Sep 2023 05:48:31 GMT
< Content-Length: 961
<
{ [961 bytes data]
100    961    100    961     0     0   619k      0  --:--:-- --:--:-- --:--:--   938k
* Connection #0 to host 172.17.0.6 left intact
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Node Exporter</title>
    <style>body {
      font-family: -apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Helvetica Neue,Arial,Noto Sans,Liberation Sans,sans-serif,Apple Color
oe UI Symbol,Noto Color Emoji;
      margin: 0;
    }
  </head>
  <body>
    <div class="card">
      <div class="card-body">
        <div class="text-center">
          <h1>Node Exporter</h1>
          <p>Node.js version: v18.12.0
```

```

vgusev@200708@killbox-vgusev2007:~$ docker exec -it prometheus-vgusev2007: sh
/prometheus $ su -
su: must be suid to work properly
/prometheus $ sudo
sh: sudo: not found
/prometheus $ vi /etc/prometheus/prometheus.yml
/prometheus $ more /etc/prometheus/prometheus.yml
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  job_name: node
  static_configs:
    - targets: ['172.17.0.6:9100']

```

Перезапускаем контейнер:

```
vgusev2007@skillbox-vgusev2007:~$ docker restart prometheus-vgusev2007
prometheus-vgusev2007
vgusev2007@skillbox-vgusev2007:~$
```

Проверяем, что узел добавился:

Prometheus Alerts Graph Status Help

Targets

All scrape pools All Unhealthy Collapse All Filter by endpoint or labels

node (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://172.17.0.6:9100/metrics	UP	instance="172.17.0.6:9100" job="node"	13.887s ago	24.318ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	8.472s ago	9.389ms	

- Делаем запрос (свободное место на диске):

skillbox:hello:9090/graph?g0.expr=node_filesystem_avail_bytes(mountpoint%3D%2Fetc%2Fhostname)%2F1024%2F1024%2F1024&g0.t...

Prometheus Alerts Graph Status Help

Use local time Enable query history Enable autocomplete Enable highlighting Enable linter

node_filesystem_avail_bytes(mountpoint="/etc/hostname") / 1024 / 1024 / 1024

Execute

Table Graph

Load time: 320ms Resolution: 14s Result series: 1

Evaluation time

{device="/dev/sda2", fstype="ext4", instance="172.17.0.6:9100", job="node", mountpoint="/etc/hostname"} 203.99044799804688

node_filesystem_avail_bytes(mountpoint="/etc/hostname") / 1024 / 1024 / 1024

Execute

Load time: 463ms Resolution: 14s Result series: 1

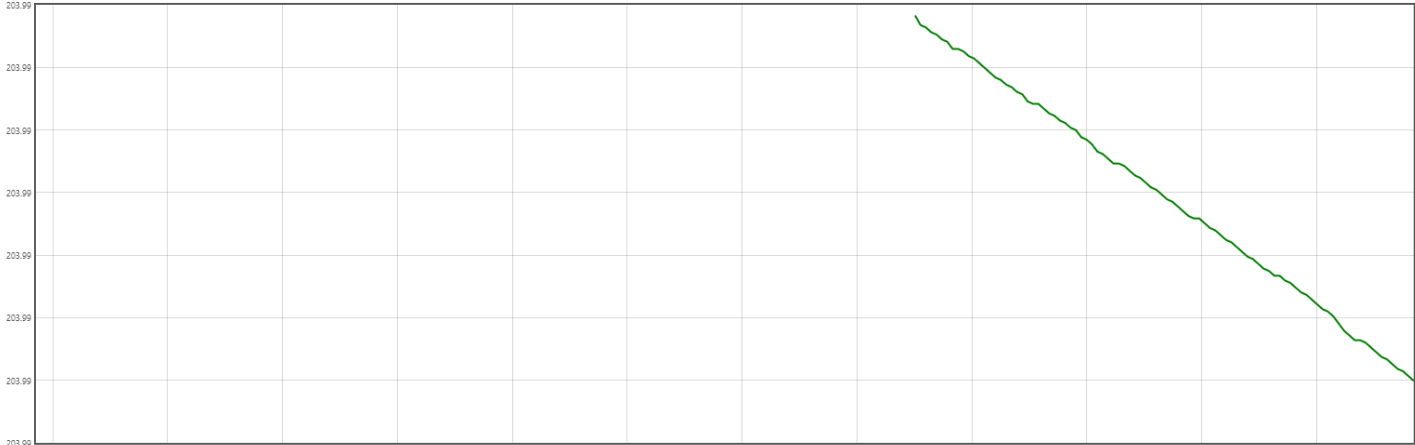
Table Graph

1h

End time

Res. (s)

Show Exemplars



{device="/dev/sda2", fstype="ext4", instance="172.17.0.6:9100", job="node", mountpoint="/etc/hostname"}

Всё правильно в контейнере, указан размер хост файловой системы, свободно 204 гб:

```
vgusev2007@skillbox-vgusev2007: ~/skillbox
/ $ df -h /etc/hosts
Filesystem                Size      Used Available Use% Mounted on
/dev/sda2                  231.2G    15.4G     204.0G   7% /etc/resolv.conf
/ $
```

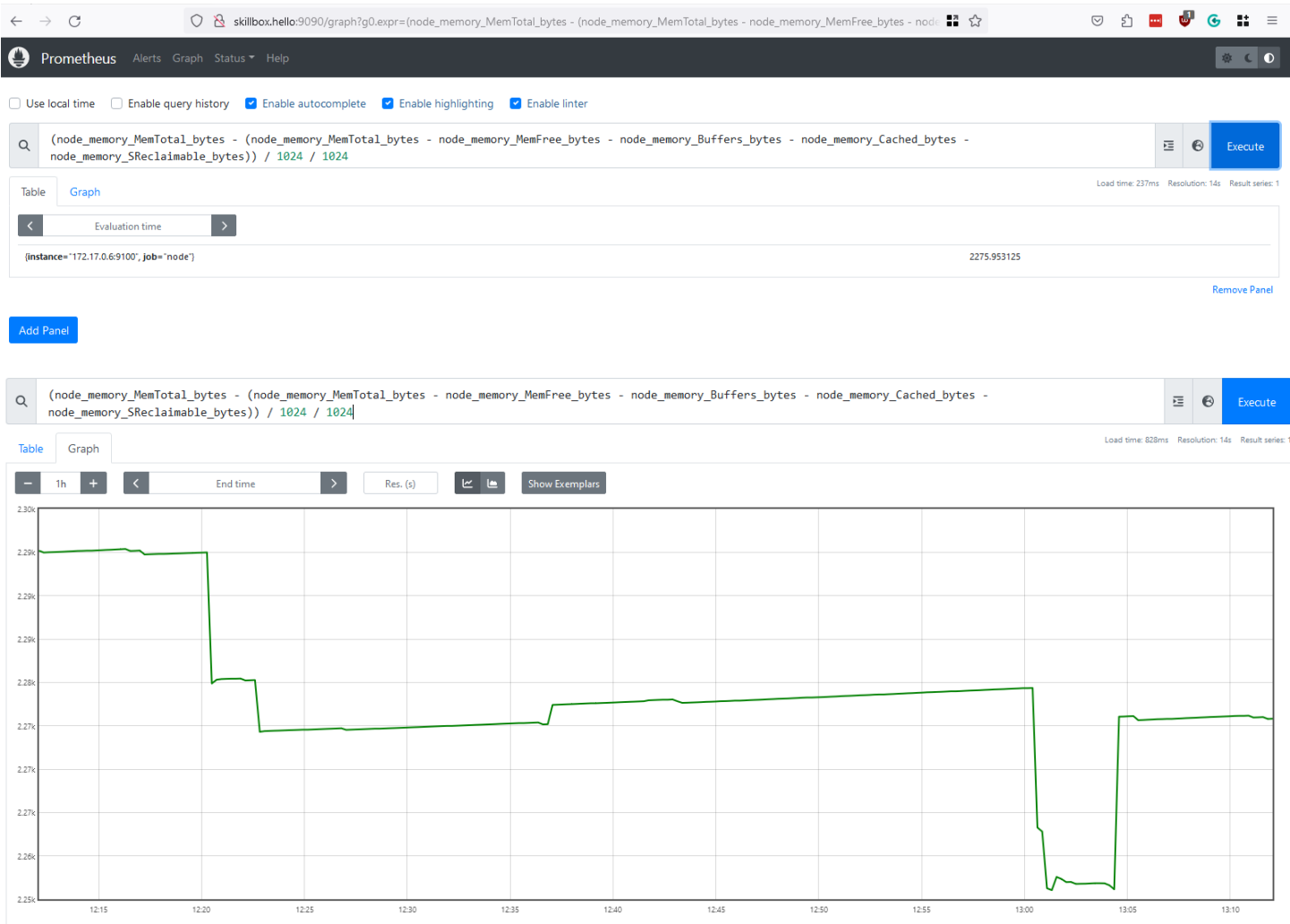


```
vgusev2007@skillbox-vgusev2007:~/skillbox$ df -h /
Filesystem                Size      Used Avail Use% Mounted on
/dev/sda2                  232G     16G    204G   8% /
vgusev2007@skillbox-vgusev2007:~/skillbox$
```

- **Находим свободную память (не забываем о памяти, которая занята пейджкешем);**

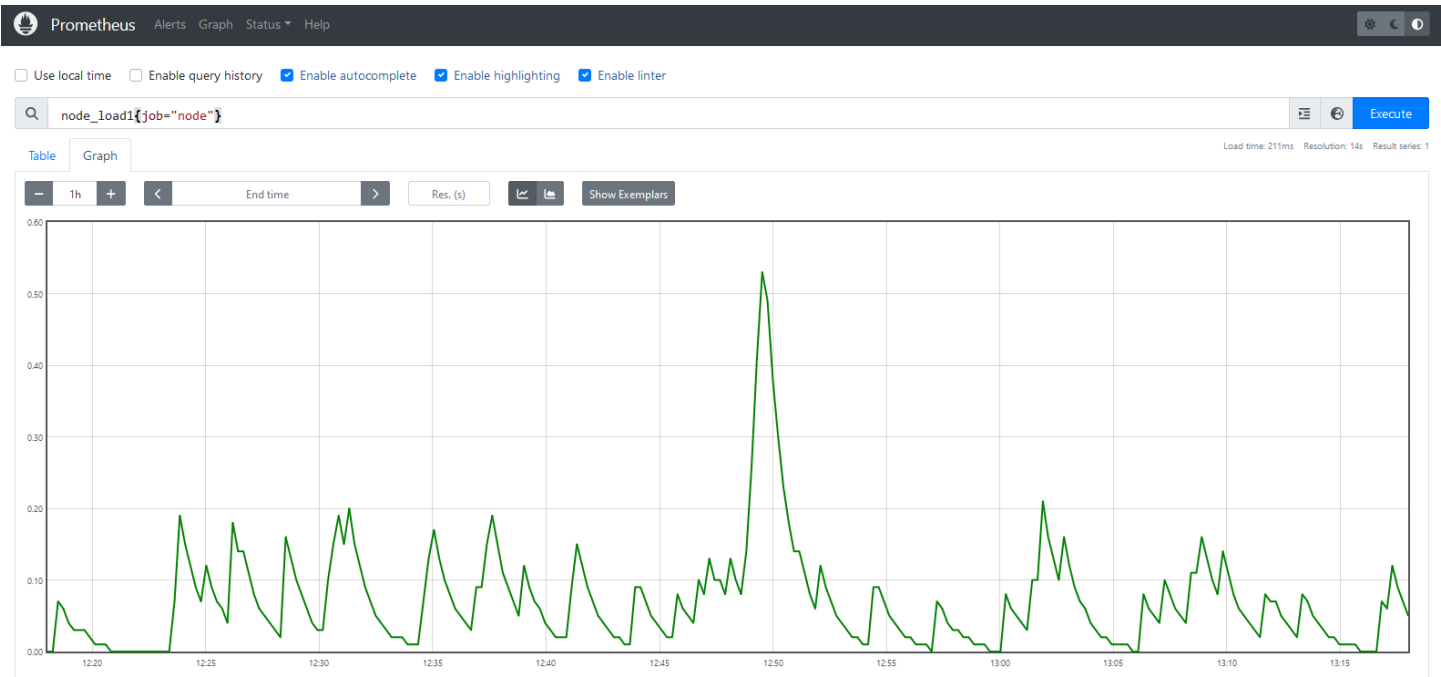
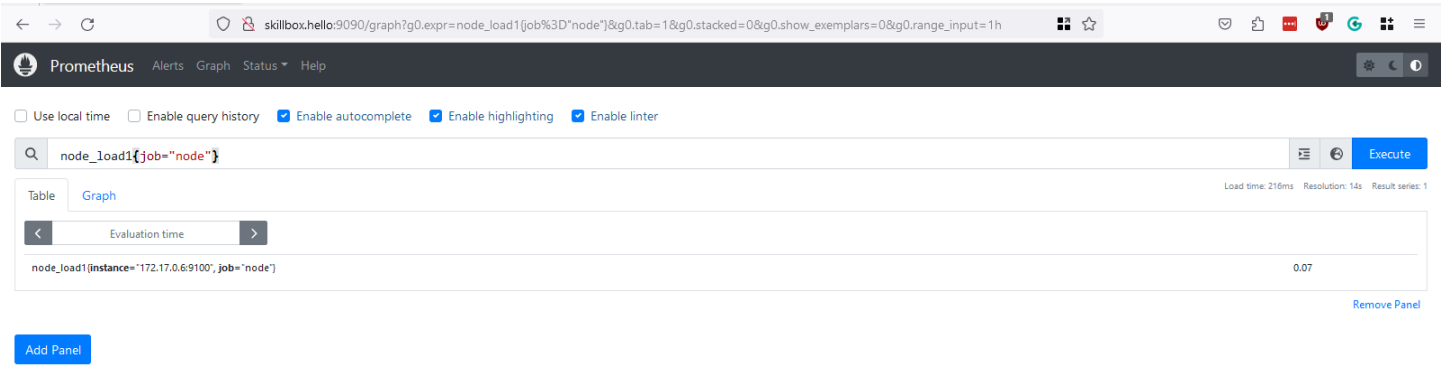
```
vgusev2007@skillbox-vgusev2007:~/skillbox$ free -m
              total        used         free      shared  buff/cache   available
Mem:           2968         693          130           2        2145        2076
Swap:          11393         457        10936
vgusev2007@skillbox-vgusev2007:~/skillbox$
vgusev2007@skillbox-vgusev2007:~/skillbox$ echo "2968 - 693" | bc
2275
vgusev2007@skillbox-vgusev2007:~/skillbox$
```

Реально свободно, 2275 мегабайт. Проверяем:



Всё сходится с выводом утилиты free

loadaverage за минуту:



Задание 2

Что нужно сделать

Постарайтесь составить необходимый и достаточный список мониторингов для Linux-сервера, на котором хостится веб-сайт. Для примера можно рассматривать блог на Wordpress, который мы развернули.

Подумайте о том, какие системные метрики важны, какие метрики, веб-серверы и базы нужно отображать. Не забудьте о мониторинге бэкенда!

Алерт будет показывать нам, что клиенты сайта получают ошибочные ответы, и это может означать проблемы в нашей инфраструктуре.

Что оценивается

Системный подход к проблеме, самостоятельное исследование.

Как отправить задание на проверку

Вы можете прислать текстом описание алертов.

Пример: «Если среди всех ответов веб-сервера больше 5% пятисотых ошибок за минуту — поджигаем алерт».

Советы и рекомендации

Вы можете вдохновляться [сайтом](#). Однако учтите, что алерты приведены в синтаксисе алерт-менеджера. Кроме того, вам понадобится описать, для чего нужен тот или иной алерт.

Ответ:

Что бы, что-то мониторить, нужна экспертность во всех продуктах, конечно. Попробую ответить на своём, базовом уровне понимания работы всех продуктов:

Сперва, я бы стал мониторить все хосты, которые обслуживают сайт, а именно: доступность, утилизацию cpu, памяти, сети, дисков. — Всё при помощи готовых exporter.

В целом, если мы говорим о мониторинге именно wordpress, где у нас может быть, не только отказ, но и разработка новой функциональности, возможно, логично использовать специальную разработку именно для wordpress, или комплекс разработок, например: <https://www.muellerpublic.de/2019/05/19/monitoring-wordpress-with-prometheus-in-a-kubernetes-cluster/>

Можно взять только самое необходимое оттуда.

Или что-то подобное, если нужно не так комплексно:

https://www.reddit.com/r/opensource/comments/lnoh6q/prometheus_wordpress_exporter/

Можно запустить экспортер в отдельном докер контейнере, и получать количество постов, тегов, и т.п. — В случае, существенного проседания цифр, можно зажигать тот или иной алерт. — Как минус подобного подхода, мы можем получать ошибочные срабатывания, в период, каникул, праздников и т.п.

По бекенду: имеет смысл мониторить некоторые параметры mysql, для чего так же, имеет смысл воспользоваться готовым экспортером, например, отсюда: https://github.com/prometheus/mysqld_exporter

Так же, имеет смысл, с бекенда собирать статистику по диску, сети, CPU

Php-fpm: https://github.com/hipages/php-fpm_exporter отсюда я бы взял как доску для Grafana, так и метрики для prometheus

По nginx: если установлена платная версия nginxplus, то можно собирать много информации, используя готовое решение, подобное этому: <https://github.com/nginxinc/nginx-prometheus-exporter>

Можно, рассмотреть nginx от Taobao 😊

Можно рассмотреть nginx status page модуль, или аналог

А можно рассмотреть, вот такой пакет:

<https://github.com/martin-helmich/prometheus-nginxlog-exporter>

И настроить всё примерно, вот по такой инструкции: <https://blog.ruanbekker.com/posts/10/>

Текст, описание алертов:

1. Если время ответа от web сервера (код 200, 404), более чем 500 мс. – поджигаем алерт
2. Если любой из хостов недоступен – поджигаем алерт – node-explorer
3. Если перегружена сеть на любом из устройств – поджигаем алерт
4. Если php-fpm недоступен – поджигаем алерт – blackbox-exporter. **Его я буду позже использовать**
https://github.com/prometheus/blackbox_exporter#blackbox-exporter-
5. Если nginx недоступен – поджигаем алерт
https://github.com/prometheus/blackbox_exporter#blackbox-exporter-
6. Если nginx проблемы с сертификатом – поджигаем алерт
https://github.com/prometheus/blackbox_exporter#blackbox-exporter-
7. Если mysql недоступен – поджигаем алерт
https://github.com/prometheus/blackbox_exporter#blackbox-exporter-
8. Если la выше количества ядер в системе – поджигаем алерт. – Например, la 5, при 4ядрах.
9. Если io выше определенного порога – поджигаем алерт

10. Если среди всех ответов веб-сервера больше 5% пятисотых ошибок за минуту — поджигаем алерт. – Можно использовать prometheus-nginxlog-exporter.

Задание 2* (повышенной сложности)

Добавьте к текстовому описанию promql-выражения.

1. Если время ответа от web сервера (любой код 200, 404), более чем 500 мс.

```
_http_upstream_time_seconds - (prometheus-nginxlog-exporter)
```

2. Если la выше количества ядер в системе – поджигаем алерт. – Например, la 5, при 4ядрах. – node explorer

```
100 * avg(1 - rate(node_cpu_seconds_total{mode="idle"}[5m])) by (instance)
```

Или:

```
100 * max(1 - rate(node_cpu_seconds_total{mode="idle"}[5m])) by (instance)
```

Примерно, это может выглядеть так. Что бы привести больше примеров, нужно ставить всё и пробовать настраивать на практике.

Задание 3

Что нужно сделать

Ранее мы установили пакет alertmanager, и настало время его использовать.

Вам потребуется:

- самостоятельно изучить настройку алерт-менеджера;
- запустить его и открыть в браузере веб-интерфейс;
- самостоятельно изучить синтаксис заведения правил;
- создать три алерта и создать ситуацию, чтобы они сработали (вы можете использовать выражения из первого задания, однако помните, что вам понадобится создать условия, при которых алерты сработают, — например, loadaverage превысит пороговое значение. Учтите этот момент при составлении демонстрационных правил).

Используйте [официальную документацию](#) и другие статьи по запросу наподобие prometheus alertmanager configuration.

Обратите внимание, что вам нужна не вся информация, приведенная в подобных статьях. Например, мы уже установили alertmanager из пакета, поэтому ни устанавливать докер, ни настраивать запуск бинарника не требуется. Кроме того, в этом задании не требуется настройка интеграцией.

Что оценивается

Работоспособность алерт-менеджера.

NB: смысловая нагрузка созданных для демонстрации алертов оцениваться не будет!

Как отправить задание на проверку

Пришлите через форму ниже скриншоты, на которых видны:

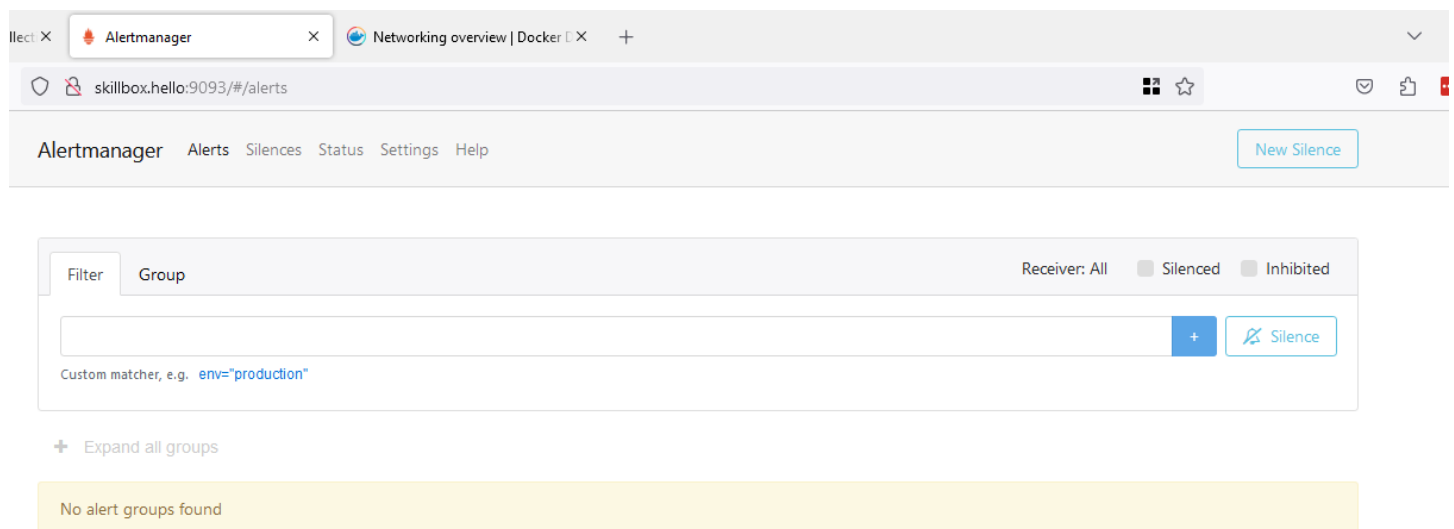
- веб-интерфейс алерт-менеджера;
- настроенные алерты в веб-интерфейсе;
- по крайней мере один сработавший алерт.

Ответ:

Запуск сервиса:

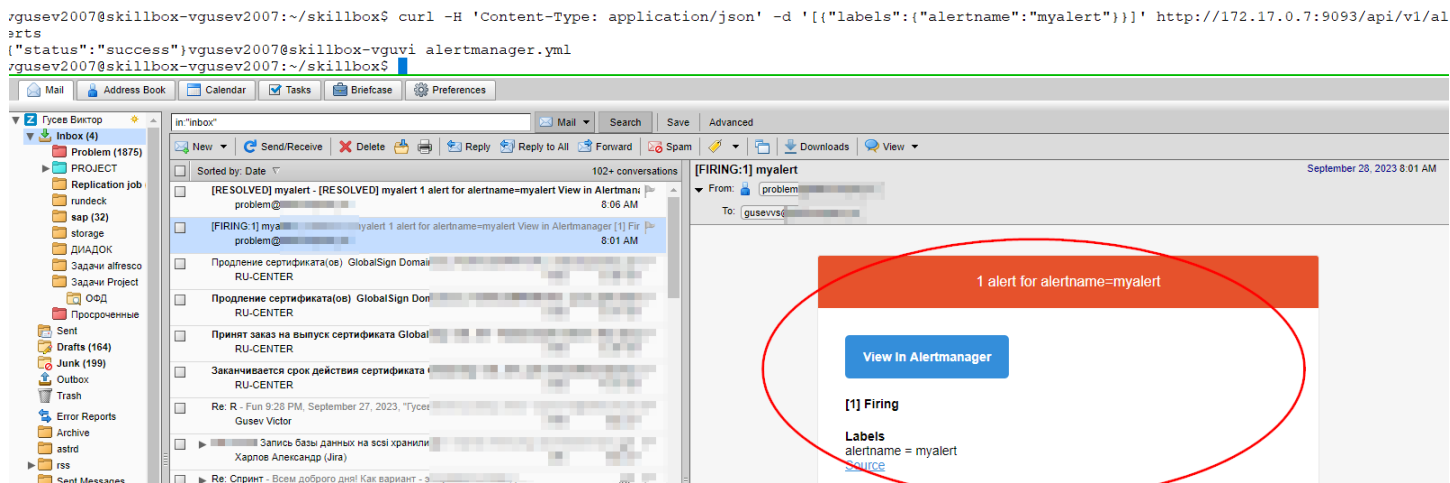
```
vgusev2007@skillbox-vgusev2007:~/skillbox$ docker run -d -p 9093:9093 --name prometheus-alertmanager-vgusev2007 prom/alertmanager  
ff3d4c657d1db8df0f534d43e773b49b24ff92e5bb240ec5b61ee4d74656e1b5  
vgusev2007@skillbox-vgusev2007:~/skillbox$ █
```

Скриншот работы сервиса:



Отправляем тестовое сообщение на email:

```
vgusev2007@skillbox-vgusev2007: ~/skillbox
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'email'
receivers:
- name: 'web.hook'
  webhook_configs:
    - url: 'http://127.0.0.1:5001/'
- name: 'email'
  email_configs:
    - to: 'gusevvs@skillbox.ru'
      from: 'problem@skillbox.ru'
      smarthost: 'mail.skillbox.ru'
      auth_username: 'skillbox'
      auth_identity: 'skillbox'
      auth_password: 'skillbox'
      send_resolved: true
inhibit_rules:
- source_match:
    severity: 'critical'
  target_match:
    severity: 'warning'
  equal: ['alertname', 'dev', 'instance']
~
```



Отлично! Всё работает.

Настроим связь, между prometheus и alertmanager:

```
vgusev2007@skillbox-vgusev2007: ~/skillbox
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds by default
  evaluation_interval: 15s # Evaluate rules every 15 seconds by default
  # scrape_timeout is set to the global default (10s)

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            - 172.17.0.7:9093
```

Создадим реальные задачи на мониторинг:

Укажем файл, где будут лежать правила:

```
/prometheus $ grep first -A5 -B5 /etc/prometheus/prometheus.yml
- targets:
  - 172.17.0.7:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
- "first_rules.yml"
# - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
```

Добавим правила

```
vgusev2007@skillbox-vgusev2007: ~/skillbox
groups:
- name: node.rules
  rules:
  - alert: InstanceDown
    expr: up{job="node"} == 0
    for: 1m
~
~

vgusev2007@skillbox-vgusev2007:~/skillbox$ docker restart prometheus-vgusev2007
prometheus-vgusev2007
```

skillbox.hello:9090/alerts?search=

Prometheus

Alerts

Graph

Status ▾

Help

✓ Inactive (1)

✓ Pending (0)

✓ Firing (0)

/etc/prometheus/first_rules.yml > node.rules

▼ InstanceDown (0 active)

name: InstanceDown

expr: up{job="node"} == 0

for: 1m

Зажигаем алерт:

```
vgusev2007@skillbox-vgusev2007:~/skillbox$ docker stop prometheus-node-exporter-vgusev2007
prometheus-node-exporter-vgusev2007
vgusev2007@skillbox-vgusev2007:~/skillbox$
```

Ещё немного, и полыхнет...

✓ Inactive (0)

✓ Pending (1)

✓ Firing (0)

🔍

Filter by name or labels

☐ Show annotations

/etc/prometheus/first_rules.yml > node.rules

pending (1)

▼ InstanceDown (1 active)

name: InstanceDown

expr: up{job="node"} == 0

for: 1m

Labels	State	Active Since	Value
alertname=InstanceDown instance=172.17.0.6:9100 job=node	PENDING	2023-09-28T05:41:46.548306082Z	0

Полыхнуло...

Inactive (0)

Pending (0)

Firing (1)

Filter by name or labels

Show annotations

/etc/prometheus/first_rules.yml > node.rules

firing (1)

▼ InstanceDown (1 active)

name: InstanceDown
expr: up{job="node"} == 0
for: 1m

Labels	State	Active Since	Value
alertname=InstanceDown instance=172.17.0.6:9100 job=node	FIRING	2023-09-28T05:41:46.548306082Z	0

Ушло в alertmanager, теперь и почта горит...:

skillbox.hello:9093/#/alerts?silenced=false&inhibited=false&active=true

☆

Alertmanager Alerts Silences Status Settings Help

New Silence

Filter Group

Receiver: All ☐ Silenced ☐ Inhibited

+

Silence

Custom matcher, e.g. env="production"

+ Expand all groups

• email

alertname="InstanceDown" + 1 alert

2023-09-28T05:42:46.548Z

Source Silence Link

instance="172.17.0.6:9100" + job="node" +

up{job="node"} == 0

Execute

Table Graph

Load time: 242ms Resolution: 14s Result series: 1

1h

End time

Res. (s)

Show Exemplars

■ up(instance="172.17.0.6:9100", job="node")

Search Save Advanced

Forward Spam Downloads View

101+ conversations [FIRING:1] InstanceDown (172.17.0.6:9100 node) September 28, 2023 8:43 AM

From: problem@ To: s@

1 alert for alertname=InstanceDown

[View In Alertmanager](#)

[1] Firing


Labels
 alertname = InstanceDown
 instance = 172.17.0.6:9100
 job = node
[Source](#)

Тушим..!



```
vgusev2007@skillbox-vgusev2007:~/skillbox$ docker restart prometheus-node-exporter-vgusev2007
prometheus-node-exporter-vgusev2007
vgusev2007@skillbox-vgusev2007:~/skillbox$
```

skillbox.hello:9090/alerts?search=

 Prometheus

Alerts

Graph

Status ▾

Help

✓ Inactive (1)

✓ Pending (0)

✓ Firing (0)

Q

Filter by name or labels

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

skillbox.hello:9093/#/alerts

Alertmanager Alerts Silences Status Settings Help

Filter

Group

Custom matcher, e.g. `env="production"`

+ Expand all groups

No alert groups found

Advanced

✎

📁

📥 Downloads

💬 View

[RESOLVED] InstanceDown (172.17.0.6:9100 node) September 28, 2023 8:53 AM

From: probl

To: gusevvs@

1 alert for alertname=InstanceDown

View In Alertmanager

[1] Resolved

Labels

alertname = InstanceDown

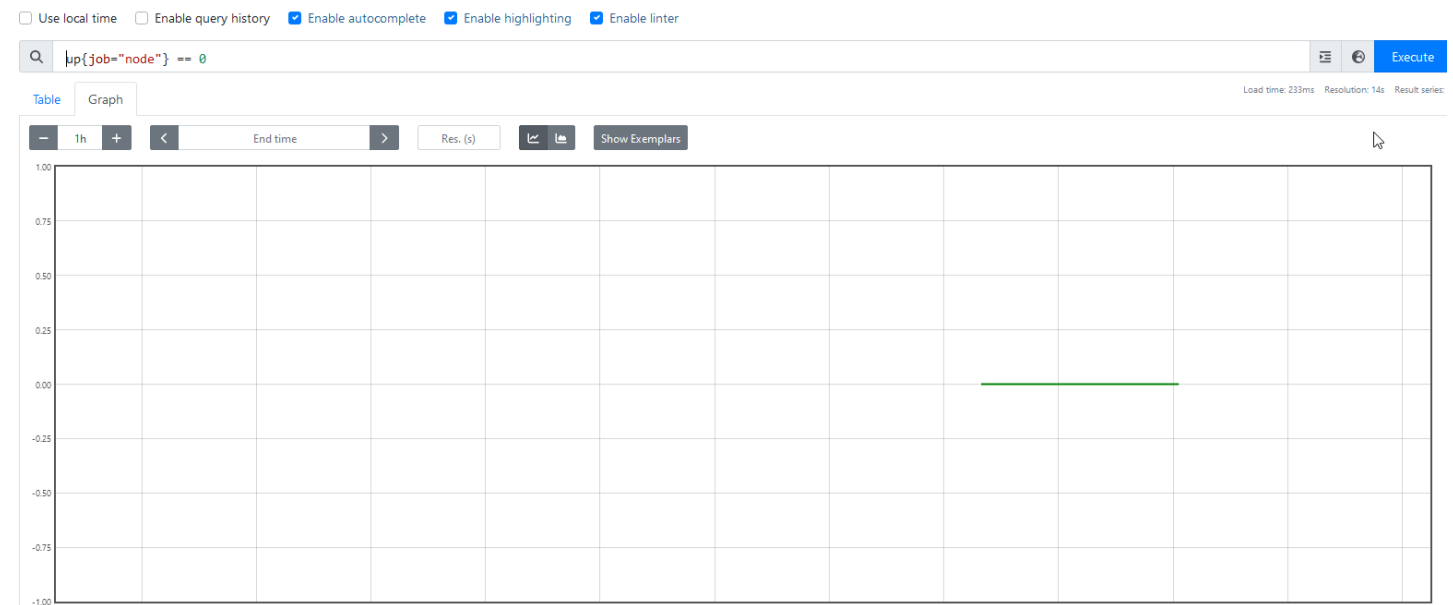
instance = 172.17.0.6:9100

job = node

[Source](#)

Кажется, всё хорошо!

Смотрим, как долго мы горели:



Добавим ещё одно правило (будем поджигаться при нагрузке на CPU 80% в течение 2х минут):

/prometheus \$ vi /etc/prometheus/first_rules.yml

```
vgusev2007@skillbox-vgusev2007: ~/skillbox
groups:
- name: node.rules
  rules:
  - alert: InstanceDown
    expr: up{job="node"} == 0
    for: 1m
  - alert: HostHighCpuLoad
    expr: (sum by (instance) (avg by (mode, instance) (rate(node_cpu_seconds_total{mode!="idle"}[2m]))) > 0.8) * on(instance) group_left (nodename) node_uname
    for: 1m
    labels:
      severity: warning
    annotations:
      summary: Host high CPU load (instance {{ $labels.instance }})
      description: "CPU load is > 80%\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"
```

Перечитаем файл с правилами:

```
PID PPID USER STAT VSZ %VSZ CPU %CPU COMMAND
1 0 nobody S 1690m 56.7 1 0.0 /bin/prometheus --config.fi
15 0 nobody S 1348 0.0 1 0.0 sh
23 15 nobody R 1340 0.0 0 0.0 top
/prometheus $ kill -s 1 1
```

Inactive (2)

Pending (0)

Firing (0)

Filter by name or labels

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

▼ HostHighCpuLoad (0 active)

name: HostHighCpuLoad

expr: (sum by (instance) (avg by (mode, instance) (rate(node_cpu_seconds_total{mode!="idle"}[2m]))) > 0.8) * on (instance) group_left (nodename) node_uname_info{nodename=~".+"}

for: 1m

labels:

severity: warning

annotations:

description: CPU load is > 80%

VALUE = {{ \$value }}

LABELS = {{ \$labels }}

summary: Host high CPU load (instance {{ \$labels.instance }})

Поджигаемся:

```
vgusev2007@skillbox-vgusev2007:~/skillbox$ docker exec -it prometheus-node-exporter-vgusev2007 sh
/ $ yes > /dev/null &
```

```
Mem: 2948140K used, 91960K free, 2864K shrd, 495952K buff, 1502412K cached
CPU: 50.1% usr 0.7% sys 0.0% nic 49.1% idle 0.0% io 0.0% irq 0.0% sirq
Load average: 0.37 0.13 0.06 5/578 18
PID PPID USER STAT VSZ %VSZ CPU %CPU COMMAND
17 12 nobody R 1328 0.0 0 50.3 yes
1 0 nobody S 708m 23.8 0 0.2 /bin/node_exporter
12 0 nobody S 1352 0.0 1 0.0 sh
18 12 nobody R 1340 0.0 1 0.0 top
```

Ещё подкинем дровишек:

```
/ $ yes > /dev/null &
```

Mem: 2948140K used, 91960K free, 2864K shrd, 495972K buff, 1502428K cached
CPU: 99.5% usr 0.4% sys 0.0% nic 0.0% idle 0.0% io 0.0% irq 0.0% sirq
Load average: 0.73 0.25 0.11 3/579 20

PID	PPID	USER	STAT	VSZ	%VSZ	CPU	%CPU	COMMAND
17	12	nobody	R	1328	0.0	0	49.9	yes
19	12	nobody	R	1328	0.0	1	49.5	yes
1	0	nobody	S	708m	23.8	0	0.0	/bin/node_exporter
12	0	nobody	S	1352	0.0	1	0.0	sh
20	12	nobody	R	1340	0.0	0	0.0	top

Задымилось:

Prometheus Alerts Graph Status Help

Inactive (1) Pending (1) Firing (0)

Filter by name or labels

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

HostHighCpuLoad (1 active)

name: HostHighCpuLoad

expr: (sum by (instance) (avg by (mode, instance) (rate(node_cpu_seconds_total{mode!="idle"}[2m]))) > 0.8) * on (instance) group_left (nodename) node_uname_info{nodename=~".+"}

for: 1m

labels: severity: warning

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

summary: Host high CPU load (instance {{ \$labels.instance }})

Labels	State	Active Since
alertname=HostHighCpuLoad instance=172.17.0.6:9100 nodename=dc60ff83e34f severity=warning	PENDING	2023-09-29T03:31:31.548306082Z

Annotations

description CPU load is > 80% VALUE = 0.90666666666666757 LABELS = map[instance:172.17.0.6:9100 nodename:dc60ff83e34f]

summary Host high CPU load (instance 172.17.0.6:9100)

Полыхнуло:

Inactive (1) Pending (0) Firing (1)

Filter by name or labels

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

HostHighCpuLoad (1 active)

name: HostHighCpuLoad

expr: (sum by (instance) (avg by (mode, instance) (rate(node_cpu_seconds_total{mode!="idle"}[2m]))) > 0.8) * on (instance) group_left (nodename) node_uname_info{nodename=~".+"}

for: 1m

labels: severity: warning

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

summary: Host high CPU load (instance {{ \$labels.instance }})

Labels	State	Active Since
alertname=HostHighCpuLoad instance=172.17.0.6:9100 nodename=dc60ff83e34f severity=warning	FIRING	2023-09-29T03:31:31.548306082Z

Annotations

description CPU load is > 80% VALUE = 0.999238095238091 LABELS = map[instance:172.17.0.6:9100 nodename:dc60ff83e34f]

summary Host high CPU load (instance 172.17.0.6:9100)

В почту тоже пришло:

[FIRING:1] HostHighCpuLoad (172.17.0.6:9100 dc60ff83e34f warning)September 29, 2023 6:33 AM

From: problem

To: gusev

1 alert for alertname=HostHighCpuLoad

View In Alertmanager

[1] Firing

Labels

alertname = HostHighCpuLoad
instance = 172.17.0.6:9100
nodename = dc60ff83e34f
severity = warning

Annotations

description = CPU load is > 80% VALUE = 0.9989523809523574 LABELS = map[instance:172.17.0.6:9100 nodename:dc60ff83e34f]
summary = Host high CPU load (instance 172.17.0.6:9100)

[Source](#)

Тушим:

```
/ $ killall yes
/ $
[2]+  Terminated                  yes 1>/dev/null
[1]+  Terminated                  yes 1>/dev/null
/ $
```

Mem: 2940024K used, 100076K free, 2872K shrd, 494444K buff, 1494784K cached
CPU: 0.0% usr 0.0% sys 0.0% nic 100% idle 0.0% io 0.0% irq 0.0% sirq
Load average: 1.22 1.43 0.77 1/579 22

PID	PPID	USER	STAT	VSZ	%VSZ	CPU	%CPU	COMMAND
1	0	nobody	S	708m	23.8	1	0.0	/bin/node_exporter
12	0	nobody	S	1352	0.0	0	0.0	sh
22	12	nobody	R	1340	0.0	0	0.0	top

☒ Inactive (2) ☒ Pending (0) ☒ Firing (0)

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

> HostHighCpuLoad (0 active)

[RESOLVED] HostHighCpuLoad (172.17.0.6:9100 dc60ff83e34f warning) September 29, 2023 6:38 AM

From: [prc] To: [gusev]

1 alert for alertname=HostHighCpuLoad

[View In Alertmanager](#)

[1] Resolved

Labels
alertname = HostHighCpuLoad
instance = 172.17.0.6:9100
nodename = dc60ff83e34f
severity = warning

Annotations
description = CPU load is > 80% VALUE = 0.8387142857142548 LABELS = map[instance:172.17.0.6:9100 nodename:dc60ff83e34f]
summary = Host high CPU load (instance 172.17.0.6:9100)
[Source](#)

Делаем третью проверку, ставим blackbox-exporter

```
vgusev2007@skillbox-vgusev2007:~/skillbox$ mkdir -p 22/6/task3 ; cd $_
vgusev2007@skillbox-vgusev2007:~/skillbox/22/6/task3$
```

```
vgusev2007@skillbox-vgusev2007:~/skillbox/22/6/task3$ docker run --rm -d -p 9115 --name prometheus-blackbox-exporter-vgusev2007 quay.io/prometheus/blackbox-exporter:latest
Unable to find image 'quay.io/prometheus/blackbox-exporter:latest' locally
latest: Pulling from prometheus/blackbox-exporter
e6b9e25f5d01: Pull complete
0c6a06713be8: Pull complete
b4bd2c7f1f55: Pull complete
f1b939a4905b: Pull complete
Digest: sha256:3af31f8bd1ad2907b4b0f7c485fde3de0a8ee0b498d42fc971f0698885c03acb
Status: Downloaded newer image for quay.io/prometheus/blackbox-exporter:latest
589996abd0a4c5bd154daf596ed87b22c90964f12c527228a9c4880e2d3c73db
vgusev2007@skillbox-vgusev2007:~/skillbox/22/6/task3$
```

Создаем новую цель:

```
/prometheus $ grep black -A13 /etc/prometheus/prometheus.yml
- job_name: 'blackbox'
  metrics_path: /probe
  params:
    module: [http_2xx] # Look for a HTTP 200 response.
  static_configs:
    - targets:
      - https://ya.ru/foo
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 172.17.0.8:9115
```

Targets

All scrape pools ▾

All

Unhealthy

Expand All

node (1/1 up)

[show more](#)

prometheus (1/1 up)

[show more](#)

```
/prometheus $ kill -s 1 1
/prometheus $
```

Targets

All scrape pools ▾

AllUnhealthyExpand All

Q

Filter by endpoint or labels

blackbox (1/1 up)

show less

Endpoint	State	Labels
http://172.17.0.8:9115/probe <div>module="http_2xx"target="https://ya.ru/foo"</div>	UP	<div>instance="https://ya.ru/foo"job="blackbox"</div>

node (1/1 up)

show more


prometheus (1/1 up)

show more

Создаем alerts:

```
/prometheus $ grep BlackboxProbeHttpFailure -A8 /etc/prometheus/first_rules.yml
- alert: BlackboxProbeHttpFailure
  expr: probe_http_status_code <= 199 OR probe_http_status_code >= 400
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: Blackbox probe HTTP failure (instance {{ $labels.instance }})
    description: "HTTP status code is not 200-399\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
```

Применяем их:

 Prometheus

AlertsGraphStatus

✓ Inactive (2)

✓ Pending (0)

✓ Firing (0)

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

> HostHighCpuLoad (0 active)

```
/prometheus $ kill -s 1 1
/prometheus $
```

Ой... Кажется, мы сразу же задымились, поскольку <https://ya.ru/foo> возвращает 404:

Снимок экрана браузера:

https://ya.ru/foo

Яндекс

Найдётся всё

Найти

Ошибка 404. Нет такой страницы

Если вы считаете, что страницы нет по нашей вине, [напишите нам](#).

Установите [Punto Switcher](#), чтобы легко исправлять адрес с одного языка на другой.

InspectorConsoleDebuggerNetworkStyle EditorPerformanceMemoryStorageAccessibilityA

Filter URLs

Status	Method	Domain	File	Initiator
404	GET	ya.ru	foo	document
	GET	mc.yandex.ru	watchjs	script

Inactive (2)Pending (1)Firing (0)

Filter by name or labels

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

> HostHighCpuLoad (0 active)

> BlackboxProbeHttpFailure (1 active)

name: BlackboxProbeHttpFailure

expr: probe_http_status_code <= 199 or probe_http_status_code >= 400

for: 1m

labels:

> severity: critical

annotations:

> description: HTTP status code is not 200-399

> VALUE = {{ \$value }}

> LABELS = {{ \$labels }}

> summary: Blackbox probe HTTP failure (instance {{ \$labels.instance }}

Labels

State

alertname=BlackboxProbeHttpFailure

instance=https://ya.ru/foo

job=blackbox

severity=critical

PENDING

Annotations

description

HTTP status code is not 200-399 VALUE = 403 LABELS = map[_name__probe_http_status_code instance=https://ya.ru/foo job=blackbox]

summary

Blackbox probe HTTP failure (instance https://ya.ru/foo)

Кажется, снова горим:

Prometheus Alerts Graph Status

Inactive (2)

Pending (0)

Firing (1)

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

> HostHighCpuLoad (0 active)

> BlackboxProbeHttpFailure (1 active)

[FIRING:1] BlackboxProbeHttpFailure (https://ya.ru/foo blackbox critical)

September 29, 2023 7:43 AM

From: [redacted]

To: gusev@[redacted]

1 alert for alertname=BlackboxProbeHttpFailure

View in Alertmanager

[1] Firing

Labels

alertname = BlackboxProbeHttpFailure
instance = https://ya.ru/foo
job = blackbox
severity = critical

Annotations

description = HTTP status code is not 200-399 VALUE = 403 LABELS = map[
name __probe_http_status_code instance=https://ya.ru/foo job=blackbox
summary = Blackbox probe HTTP failure (instance https://ya.ru/foo)

Source

Тушим!!!!

Будем на главную ходить Яндекса, как все нормальные люди:

```
/prometheus $ grep black -A13 /etc/prometheus/prometheus.yml
- job_name: 'blackbox'
  metrics_path: /probe
  params:
    module: [http_2xx] # Look for a HTTP 200 response.
  static_configs:
    - targets:
      - https://ya.ru
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 172.17.0.8:9115

/prometheus $ kill -s 1 1
/prometheus $
```

Targets

All scrape pools

All Unhealthy Expand All

Filter by endpoint or labels

blackbox (1/1 up) show less

Endpoint	State	Labels
http://172.17.0.8:9115/probe module="http_2xx" target="https://ya.ru"	UP	instance="https://ya.ru" job="blackbox"

Кажется, потухло... Хотя, конечно, мы обманули систему, и просто поправили url

Inactive (3)

Pending (0)

Firing (0)

Q

/etc/prometheus/first_rules.yml > node.rules

> InstanceDown (0 active)

> HostHighCpuLoad (0 active)

> BlackboxProbeHttpFailure (0 active)

name: BlackboxProbeHttpFailure

expr: probe_http_status_code <= 199 or probe_http_status_code >= 400

for: 1m

labels:

severity: critical

annotations:

description: HTTP status code is not 200-399

VALUE = {{ \$value }}

LABELS = {{ \$labels }}

summary: Blackbox probe HTTP failure (instance {{ \$labels.instance }})

[RESOLVED] BlackboxProbeHttpFailure (https://ya.ru/foo blackbox critical)

September 29, 2023 7:53 AM

From: pr

To: gusev

1 alert for alertname=BlackboxProbeHttpFailure

View In Alertmanager

[1] Resolved

Labels

alertname = BlackboxProbeHttpFailure

instance = https://ya.ru/foo

job = blackbox

severity = critical

Annotations

description = HTTP status code is not 200-399 VALUE = 403 LABELS = map[__name__:probe_http_status_code instance:https://ya.ru/foo job:blackbox]

summary = Blackbox probe HTTP failure (instance https://ya.ru/foo)

Source