Установка Prometheus + Alertmanager + node\_exporter на Linux

Обновлено Обновлено: 10.01.2021 Опубликовано Опубликовано: 12.08.2019

В двух словах, Prometheus — система мониторинга, обладающая возможностями тонкой настройки метрик. Она будет полезна для отслеживания состояния работы сервисов на низком уровне.

Данная инструкция позволит установить prometheus как на системы RPM (Red Hat, CentOS), так и deb (Debian, Ubuntu). Помимо Prometheus мы установим Alertmanager для возможности отправлять тревоги и node\_exporter для мониторинга сервера Linux.

[**Готовим сервер**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prepare)  
    [Правильное время](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prepare-time)  
    [Фаервол](https://www.dmosk.ru/instruktions.php?object=prometheus-linux" \l "prepare-firewall)  
    [Безопасность SELinux](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prepare-selinux)  
[**Установка и настройка Prometheus**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prometheus)  
    [Загрузка](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prometheus-download)  
    [Установка](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prometheus-install)  
    [Назначение прав](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prometheus-rights)  
    [Запуск и проверка](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prometheus-run)  
    [Автозапуск](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#prometheus-autorun)  
[**Установка и настройка Alertmanager**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#alertmanager)  
    [Загрузка](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#alertmanager-download)  
    [Установка](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#alertmanager-install)  
    [Назначение прав](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#alertmanager-rights)  
    [Автозапуск](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#alertmanager-autoruns)  
[**node\_exporter**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#node)  
    [Загрузка](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#node-download)  
    [Установка](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#node-install)  
    [Назначение прав](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#node-rights)  
    [Автозапуск](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#node-autoruns)  
[**Отображение метрик с node\_exporter в консоли prometheus**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#metric)  
[**Отображение тревог**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#alerts)  
[**Отправка уведомлений по почте**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#notifications)[**Мониторинг сервисов Linux**](https://www.dmosk.ru/instruktions.php?object=prometheus-linux#systemd)

Подготовка сервера

Настроим некоторые параметры сервера, необходимые для правильно работы системы.

Время

Для отображения событий в правильное время, необходимо настроить его синхронизацию. Для этого установим chrony:

а) если на системе **CentOS / Red Hat**:

yum install chrony

systemctl enable chronyd

systemctl start chronyd

б) если на системе **Ubuntu / Debian**:

apt-get install chrony

systemctl enable chrony

systemctl start chrony

Брандмауэр

На фаерволе, при его использовании, необходимо открыть порты:

* TCP 9090 — http для сервера прометеус.
* TCP 9093 — http для алерт менеджера.
* TCP и UDP 9094 — для алерт менеджера.
* TCP 9100 — для node\_exporter.

а) с помощью firewalld:

firewall-cmd --permanent --add-port=9090/tcp --add-port=9093/tcp --add-port=9094/{tcp,udp} --add-port=9100/tcp

firewall-cmd --reload

б) с помощью iptables:

iptables -I INPUT 1 -p tcp --match multiport --dports 9090,9093,9094,9100 -j ACCEPT

iptables -A INPUT -p udp --dport 9094 -j ACCEPT

в) с помощью ufw:

ufw allow 9090,9093,9094,9100/tcp

ufw allow 9094/udp

ufw reload

SELinux

По умолчанию, SELinux работает в операционный системах на базе Red Hat. Проверяем, работает ли она в нашей системе:

getenforce

Если мы получаем в ответ:

Enforcing

... необходимо отключить его командами:

setenforce 0

sed -i 's/^SELINUX=.\*/SELINUX=disabled/g' /etc/selinux/config

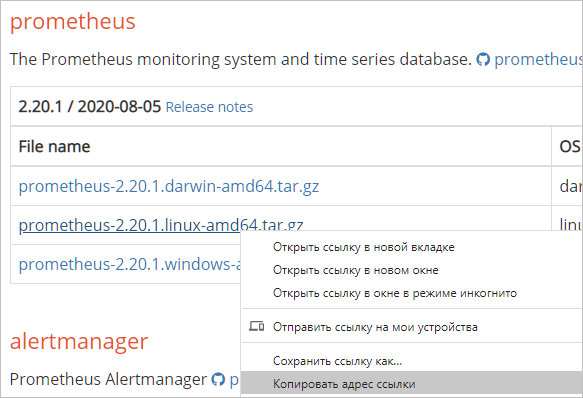
*\* если же мы получим ответ****The program 'getenforce' is currently not installed****, то SELinux не установлен в системе.*

Prometheus

Prometheus не устанавливается из репозитория и имеет, относительно, сложный процесс установки. Необходимо скачать исходник, создать пользователя, вручную скопировать нужные файлы, назначить права и создать юнит для автозапуска.

Загрузка

Переходим на [официальную страницу загрузки](https://prometheus.io/download/) и копируем ссылку на пакет для Linux:



... и используем ее для загрузки пакета на Linux:

wget https://github.com/prometheus/prometheus/releases/download/v2.20.1/prometheus-2.20.1.linux-amd64.tar.gz

*\* если система вернет ошибку, необходимо установить пакет wget.*

Установка (копирование файлов)

После того, как мы скачали архив prometheus, необходимо его распаковать и скопировать содержимое по разным каталогам.

Для начала создаем каталоги, в которые скопируем файлы для prometheus:

mkdir /etc/prometheus

mkdir /var/lib/prometheus

Распакуем наш архив:

tar zxvf prometheus-\*.linux-amd64.tar.gz

... и перейдем в каталог с распакованными файлами:

cd prometheus-\*.linux-amd64

Распределяем файлы по каталогам:

cp prometheus promtool /usr/local/bin/

cp -r console\_libraries consoles prometheus.yml /etc/prometheus

Назначение прав

Создаем пользователя, от которого будем запускать систему мониторинга:

useradd --no-create-home --shell /bin/false prometheus

*\* мы создали пользователя****prometheus****без домашней директории и без возможности входа в консоль сервера.*

Задаем владельца для каталогов, которые мы создали на предыдущем шаге:

chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus

Задаем владельца для скопированных файлов:

chown prometheus:prometheus /usr/local/bin/{prometheus,promtool}

Запуск и проверка

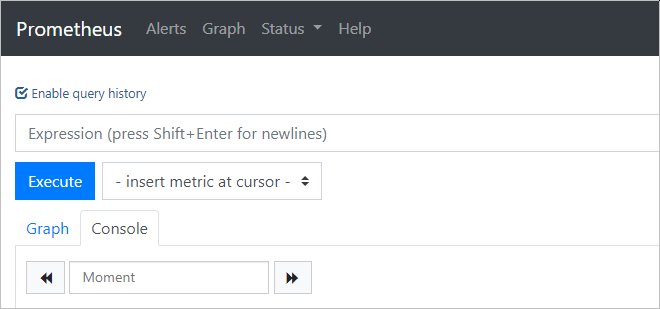
Запускаем prometheus командой:

/usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/ --web.console.templates=/etc/prometheus/consoles --web.console.libraries=/etc/prometheus/console\_libraries

... мы увидим лог запуска — в конце «Server is ready to receive web requests»:

level=info ts=2019-08-07T07:39:06.849Z caller=main.go:621 msg="Server is ready to receive web requests."

Открываем веб-браузер и переходим по адресу http://<IP-адрес сервера>:9090 — загрузится консоль Prometheus:



Установка завершена.

Автозапуск

Мы установили наш сервер мониторинга, но его необходимо запускать вручную, что совсем не подходит для серверных задач. Для настройки автоматического старта Prometheus мы создадим новый юнит в systemd.

Возвращаемся к консоли сервера и прерываем работу Prometheus с помощью комбинации Ctrl + C. Создаем файл prometheus.service:

vi /etc/systemd/system/prometheus.service

[Unit]  
Description=Prometheus Service  
After=network.target  
  
[Service]  
User=prometheus  
Group=prometheus  
Type=simple  
ExecStart=/usr/local/bin/prometheus \  
--config.file /etc/prometheus/prometheus.yml \  
--storage.tsdb.path /var/lib/prometheus/ \  
--web.console.templates=/etc/prometheus/consoles \  
--web.console.libraries=/etc/prometheus/console\_libraries  
ExecReload=/bin/kill -HUP $MAINPID  
Restart=on-failure  
  
[Install]  
WantedBy=multi-user.target

Перечитываем конфигурацию systemd:

systemctl daemon-reload

Разрешаем автозапуск:

systemctl enable prometheus

После ручного запуска мониторинга, который мы делали для проверки, могли сбиться права на папку библиотек — снова зададим ей владельца:

chown -R prometheus:prometheus /var/lib/prometheus

Запускаем службу:

systemctl start prometheus

... и проверяем, что она запустилась корректно:

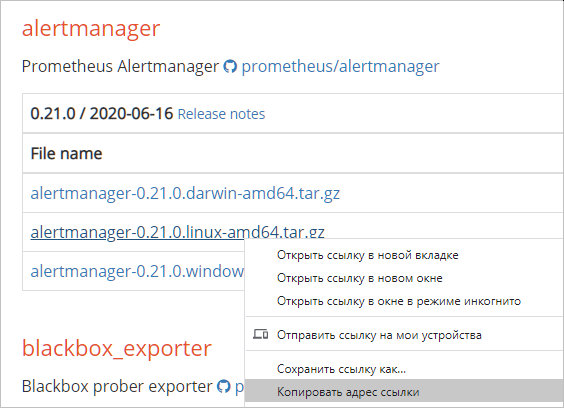
systemctl status prometheus

Alertmanager

Alertmanager нужен для сортировки и группировки событий. Он устанавливается по такому же принципу, что и prometheus.

Загрузка

На той же [официальной странице загрузки](https://prometheus.io/download/#alertmanager) копируем ссылку на Alertmanager для Linux:



После предыдущей установки мы должны были остаться в каталоге прометеуса — выходим на уровень выше:

cd ..

Теперь используем ссылку для загрузки alertmanager:

wget https://github.com/prometheus/alertmanager/releases/download/v0.21.0/alertmanager-0.21.0.linux-amd64.tar.gz

Установка

Создаем каталоги для alertmanager:

mkdir /etc/alertmanager /var/lib/prometheus/alertmanager

Распакуем наш архив:

tar zxvf alertmanager-\*.linux-amd64.tar.gz

... и перейдем в каталог с распакованными файлами:

cd alertmanager-\*.linux-amd64

Распределяем файлы по каталогам:

cp alertmanager amtool /usr/local/bin/

cp alertmanager.yml /etc/alertmanager

Назначение прав

Создаем пользователя, от которого будем запускать alertmanager:

useradd --no-create-home --shell /bin/false alertmanager

*\* мы создали пользователя****alertmanager****без домашней директории и без возможности входа в консоль сервера.*

Задаем владельца для каталогов, которые мы создали на предыдущем шаге:

chown -R alertmanager:alertmanager /etc/alertmanager /var/lib/prometheus/alertmanager

Задаем владельца для скопированных файлов:

chown alertmanager:alertmanager /usr/local/bin/{alertmanager,amtool}

Автозапуск

Создаем файл alertmanager.service в systemd:

vi /etc/systemd/system/alertmanager.service

[Unit]  
Description=Alertmanager Service  
After=network.target  
  
[Service]  
EnvironmentFile=-/etc/default/alertmanager  
User=alertmanager  
Group=alertmanager  
Type=simple  
ExecStart=/usr/local/bin/alertmanager \  
          --config.file=/etc/alertmanager/alertmanager.yml \  
          --storage.path=/var/lib/prometheus/alertmanager \  
          $ALERTMANAGER\_OPTS  
ExecReload=/bin/kill -HUP $MAINPID  
Restart=on-failure  
  
[Install]  
WantedBy=multi-user.target

Перечитываем конфигурацию systemd:

systemctl daemon-reload

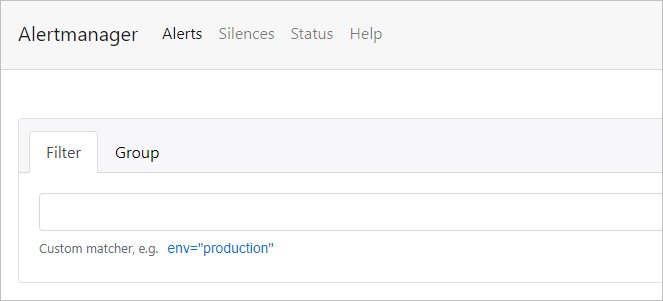
Разрешаем автозапуск:

systemctl enable alertmanager

Запускаем службу:

systemctl start alertmanager

Открываем веб-браузер и переходим по адресу http://<IP-адрес сервера>:9093 — загрузится консоль alertmanager:



Установка завершена.

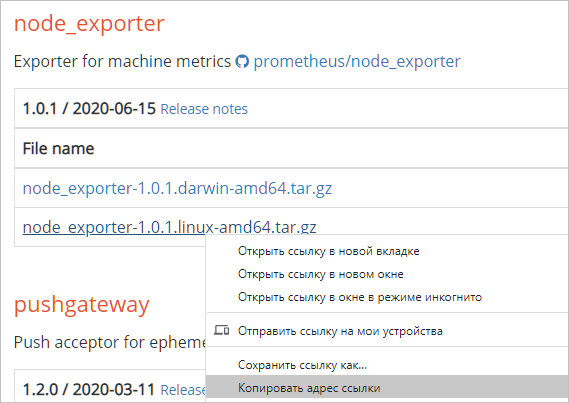
node\_exporter

Для получения метрик от операционной системы, установим и настроим node\_exporter на тот же сервер прометеуса (и на все клиентские компьютеры). Процесс установки такой же, как у Prometheus и Alertmanager.

Если мы устанавливаем node\_exporter на клиента, необходимо проверить наличие брандмауэра и, при необходимости, открыть tcp-порт 9100.

Загрузка

Заходим на [страницу загрузки](https://prometheus.io/download/#node_exporter) и копируем ссылку на node\_exporter:



*\* обратите внимание, что для некоторых приложений есть свои готовые экспортеры.*

После предыдущей установки мы должны были остаться в каталоге алерт менеджера — выходим на уровень выше:

cd ..

Теперь используем ссылку для загрузки node\_exporter:

wget https://github.com/prometheus/node\_exporter/releases/download/v1.0.1/node\_exporter-1.0.1.linux-amd64.tar.gz

Установка

Распакуем скачанный архив:

tar zxvf node\_exporter-\*.linux-amd64.tar.gz

... и перейдем в каталог с распакованными файлами:

cd node\_exporter-\*.linux-amd64

Копируем исполняемый файл в bin:

cp node\_exporter /usr/local/bin/

Назначение прав

Создаем пользователя nodeusr:

useradd --no-create-home --shell /bin/false nodeusr

Задаем владельца для исполняемого файла:

chown -R nodeusr:nodeusr /usr/local/bin/node\_exporter

Автозапуск

Создаем файл node\_exporter.service в systemd:

vi /etc/systemd/system/node\_exporter.service

[Unit]  
Description=Node Exporter Service  
After=network.target  
  
[Service]  
User=nodeusr  
Group=nodeusr  
Type=simple  
ExecStart=/usr/local/bin/node\_exporter  
ExecReload=/bin/kill -HUP $MAINPID  
Restart=on-failure  
  
[Install]  
WantedBy=multi-user.target

Перечитываем конфигурацию systemd:

systemctl daemon-reload

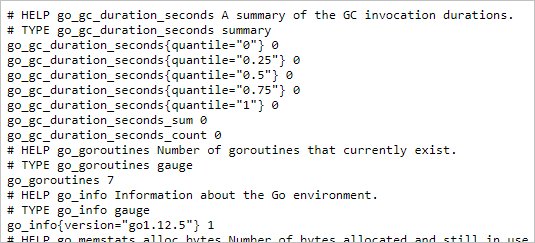
Разрешаем автозапуск:

systemctl enable node\_exporter

Запускаем службу:

systemctl start node\_exporter

Открываем веб-браузер и переходим по адресу http://<IP-адрес сервера или клиента>:9100/metrics — мы увидим метрики, собранные node\_exporter:



Установка завершена.

Отображение метрик с node\_exporter в консоли prometheus

Открываем конфигурационный файл prometheus:

vi /etc/prometheus/prometheus.yml

В разделе scrape\_configs добавим:

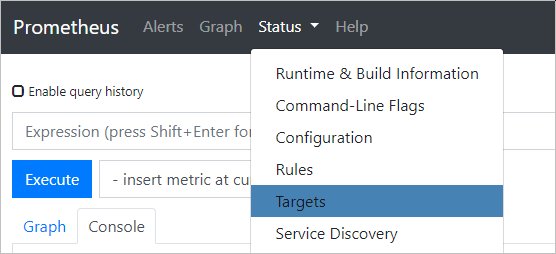
scrape\_configs:  
  ...  
  - job\_name: 'node\_exporter\_clients'  
    scrape\_interval: 5s  
    static\_configs:  
      - targets: ['192.168.0.14:9100','192.168.0.15:9100']

*\* в данном примере мы добавили клиента с IP-адресом****192.168.0.14****, рабочее название для группы клиентов****node\_exporter\_clients****. Для примера, мы также добавили клиента****192.168.0.15****— чтобы продемонстрировать, что несколько клиентов добавляется через запятую.*

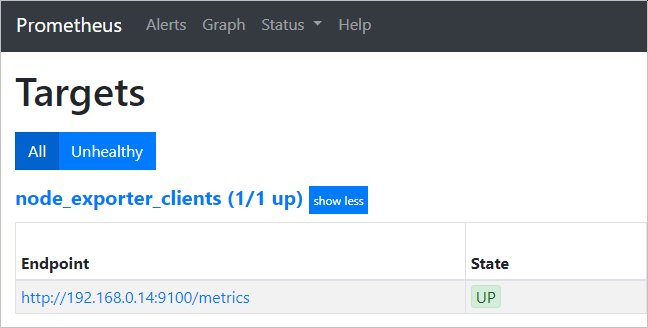
Чтобы настройка вступила в действие, перезагружаем наш сервис prometheus:

systemctl restart prometheus

Заходим в веб-консоль prometheus и переходим в раздел **Status** - **Targets**:



... в открывшемся окне мы должны увидеть нашу группу хостов и сам компьютер с установленной node\_exporter:



*\* статус также должен быть****UP****.*

Отображение тревог

Создадим простое правило, реагирующее на недоступность клиента.

Создаем файл с правилом:

vi /etc/prometheus/alert.rules.yml

groups:  
- name: alert.rules  
  rules:  
  - alert: InstanceDown  
    expr: up == 0  
    for: 1m  
    labels:  
      severity: critical  
    annotations:  
      description: '{{ $labels.instance }} of job {{ $labels.job }} has been down  
        for more than 1 minute.'  
      summary: Instance {{ $labels.instance }} down

Теперь подключим наше правило в конфигурационном файле prometheus:

vi /etc/prometheus/prometheus.yml

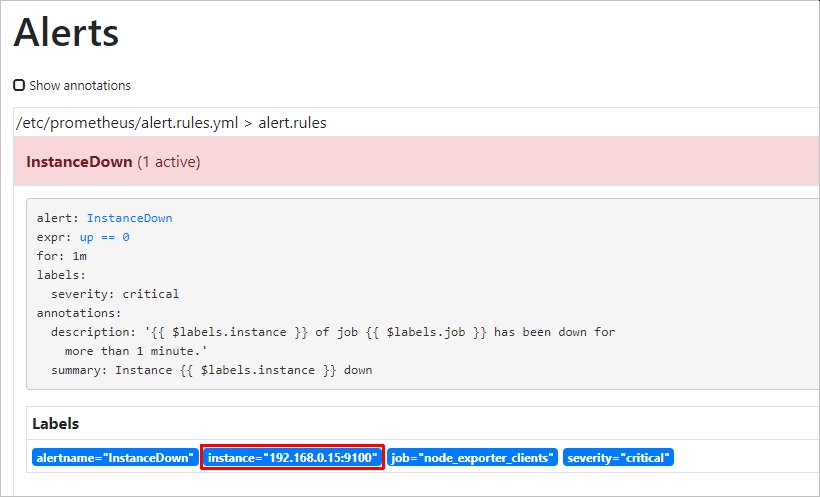
...  
rule\_files:  
  # - "first\_rules.yml"  
  # - "second\_rules.yml"  
  - "alert.rules.yml"  
...

*\* в данном примере мы добавили наш файл****alert.rules.yml****в секцию****rule\_files****. Закомментированные файлы****first\_rules.yml****и****second\_rules.yml****уже были в файле в качестве примера.*

Перезапускаем сервис:

systemctl restart prometheus

Открываем веб-консоль прометеуса и переходим в раздел **Alerts**. Если мы добавим клиента и попробуем его отключить для примера, мы увидим тревогу:



Отправка уведомлений

Теперь настроим связку с алерт менеджером для отправки уведомлений на почту.

Настроим alertmanager:

vi /etc/alertmanager/alertmanager.yml

В секцию **global** добавим:

global:  
  ...  
  smtp\_from: monitoring@dmosk.ru

*\* мы будем отправлять сообщения от email****monitoring@dmosk.ru****.*

Приведем секцию **route** к виду:

route:  
  group\_by: ['alertname', 'instance', 'severity']  
  group\_wait: 10s  
  group\_interval: 10s  
  repeat\_interval: 1h  
  receiver: 'web.hook'  
  
  routes:  
    - receiver: send\_email  
      match:  
        alertname: InstanceDown

*\* в данном примере нами был добавлен маршрут, который отлавливает событие****InstanceDown****и запускает ресивер****send\_email****.*

... далее добавим еще один ресивер:

receivers:  
...  
- name: send\_email  
  email\_configs:  
  - to: alert@dmosk.ru  
    smarthost: localhost:25  
    require\_tls: false

*\* в данном примере мы отправляем сообщение на почтовый ящик****alert@dmosk.ru****с локального сервера. Обратите внимание, что для отправки почты наружу у нас должен быть корректно настроенный почтовый сервер (в противном случае, почта может попадать в СПАМ).*

Перезапустим сервис для алерт менеджера:

systemctl restart alertmanager

Теперь настроим связку prometheus с alertmanager — открываем конфигурационный файл сервера мониторинга:

vi /etc/prometheus/prometheus.yml

Приведем секцию **alerting** к виду:

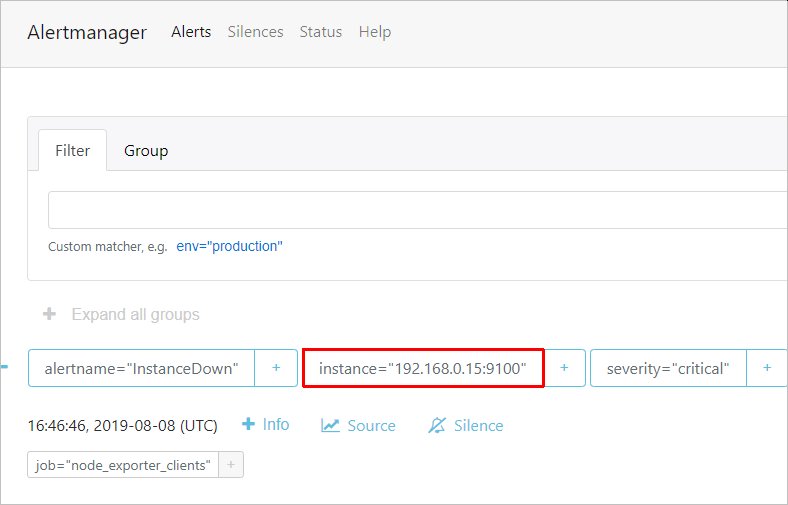
alerting:  
  alertmanagers:  
  - static\_configs:  
    - targets:  
      - 192.168.0.14:9093

*\* где****192.168.0.14****— IP-адрес сервера, на котором у нас стоит alertmanager.*

Перезапускаем сервис:

systemctl restart prometheus

Немного ждем и заходим на веб интерфейс алерт менеджера — мы должны увидеть тревогу:



... а на почтовый ящик должно прийти письмо с тревогой.

Мониторинг служб Linux

Для мониторинга сервисов с помощью Prometheus мы настроим сбор метрик и отображение тревог.

Сбор метрие с помощью node\_exporter

Открываем сервис, созданный для node\_exporter:

vi /etc/systemd/system/node\_exporter.service

... и добавим к ExecStart:

...  
ExecStart=/usr/local/bin/node\_exporter --collector.systemd  
...

*\* данная опция указывает экспортеру мониторить состояние каждой службы.*

При необходимости, мы можем либо мониторить отдельные службы, добавив опцию **collector.systemd.unit-whitelist**:

ExecStart=/usr/local/bin/node\_exporter --collector.systemd --collector.systemd.unit-whitelist="(chronyd|mariadb|nginx).service"

*\* в данном примере будут мониториться только сервисы****chronyd****,****mariadb****и****nginx****.*

... либо наоборот — мониторить все службы, кроме отдельно взятых:

ExecStart=/usr/local/bin/node\_exporter --collector.systemd --collector.systemd.unit-blacklist="(auditd|dbus|kdump).service"

*\* при такой настройке мы запретим мониторинг сервисов****auditd****,****dbus****и****kdump****.*

Чтобы применить настройки, перечитываем конфиг systemd:

systemctl daemon-reload

Перезапускаем node\_exporter:

systemctl restart node\_exporter

Отображение тревог

Настроим мониторинг для службы NGINX.

Создаем файл с правилом:

vi /etc/prometheus/services.rules.yml

groups:  
- name: services.rules  
  rules:  
    - alert: nginx\_service  
      expr: node\_systemd\_unit\_state{name="nginx.service",state="active"} == 0  
      for: 1s  
      annotations:  
        summary: "Instance {{ $labels.instance }} is down"  
        description: "{{ $labels.instance }} of job {{ $labels.job }} is down."

Подключим файл с описанием правил в конфигурационном файле prometheus:

vi /etc/prometheus/prometheus.yml

...  
rule\_files:  
  # - "first\_rules.yml"  
  # - "second\_rules.yml"  
  - "alert.rules.yml"  
  - "services.rules.yml"  
...

*\* в данном примере мы добавили наш файл****services.rules.yml****к уже ранее добавленному****alert.rules.yml****в секцию****rule\_files****.*

Перезапускаем prometheus:

systemctl restart prometheus

Для проверки, остановим наш сервис:

systemctl stop nginx

В консоли Prometheus в разделе Alerts мы должны увидеть тревогу:

