

Мониторинг: Prometheus



Артур
Сагутдинов



Артур Сагутдинов

Инженер DevOps
департамента голосовых
цифровых технологий

Banks Soft Systems



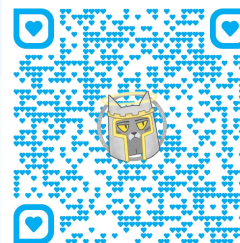
15+ лет в сфере ИТ



Разрабатываю и внедряю
линуксовую инфраструктуру



[Сисадминский блог](#)

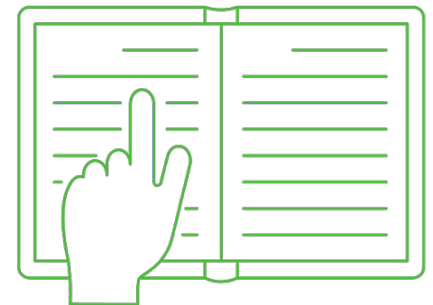


https://t.me/belf_igor

Предисловие

На этом занятии мы:

- познакомимся с Prometheus;
- узнаем, что такое Grafana;
- установим Prometheus и Grafana;
- подключим Grafana к Prometheus;
- получим данные о состоянии сервера в Prometheus.



План занятия

1. [Что такое Prometheus?](#)
2. [Историческая справка](#)
3. [Как устроен Prometheus?](#)
4. [Сравнение с Zabbix](#)
5. [Установка Prometheus](#)
6. [Интеграция Grafana и Prometheus](#)
7. [Итоги](#)
8. [Домашнее задание](#)



Что такое Prometheus?

Что такое Prometheus?

— это open source набор инструментов, предназначенный для оповещения о каких-либо событиях, происходящих на хостах, находящихся у него на мониторинге.

Подходит для:

- обработки данных, представленных во временных рядах;
- мониторинга аппаратных ресурсов;
- мониторинга динамичной сервис-ориентированной архитектуры.



Историческая справка



Историческая справка

Впервые разработан для внутреннего использования проектом SoundCloud. С момента своего появления в 2012 году, Prometheus был взят на вооружение многими крупными компаниями, например DigitalOcean, Ericsson, CoreOS, Weaveworks, Red Hat, и Google.

Имеет активную команду разработчиков и [КОМЬЮНИТИ](#).

В данный момент проект развивается независимо от какой-либо компании.



Как устроен Prometheus?



Как устроен Prometheus?

Данные в Prometheus хранятся в виде имён метрик и пар ключей с их значениями.

Каждый экземпляр Prometheus является независимой и автономной единицей.

Данные собираются через pull запросы по HTTP.

Для сбора информации о чём-либо библиотеки Prometheus должны быть интегрированы в этот продукт либо необходимо использовать отдельные экспортеры, являющиеся посредником между наблюдаемым объектом и Prometheus.



Сравнение с Zabbix

Сравнение с Zabbix

В отличие от Zabbix, который является законченным продуктом, имеющим из коробки весь базовый необходимый функционал, Prometheus требует доработок:

- Интерфейс с графиками и дашбордами является сторонней разработкой, подключаемой к Prometheus;
- Ограниченная работа Prometheus с оповещениями;
- Недоступны авторизация и контроль пользователей для Prometheus.

Сравнение с Zabbix

Изначально Prometheus рассчитан на активную доработку со стороны эксплуатирующей организации, что делает его более удобным для специфических задач:

- Программисты не ограничены уже существующими механизмами авторизации, способами извлечения данных из приложений, интерфейсом и чем-либо вообще;
- Эксплуатирующие подразделения для доведения Prometheus до кондиции могут использовать уже имеющиеся, готовые и работающие решения.



Установка Prometheus

Устанавливаем Prometheus

Создаём пользователя prometheus:

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

Последнюю версию ищем на [гитхабе](https://github.com/prometheus/prometheus/releases/download/v2.28.1/prometheus-2.28.1.linux-amd64.tar.gz).

```
wget  
https://github.com/prometheus/prometheus/releases/download/v2.28.1/prometheus-2.28.1.linux-amd64.tar.gz
```

Извлекаем архив и копируем файлы в необходимые директории:

```
tar xvfz prometheus-2.28.1.linux-amd64.tar.gz  
cd prometheus-2.28.1.linux-amd64  
mkdir /etc/prometheus  
mkdir /var/lib/prometheus  
cp ./prometheus promtool /usr/local/bin/  
cp -R ./console_libraries /etc/prometheus  
cp -R ./consoles /etc/prometheus  
cp ./prometheus.yml /etc/prometheus
```

Настраиваем права на файлы Prometheus

Передаём права на файлы пользователю prometheus:

```
chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus  
chown prometheus:prometheus /usr/local/bin/prometheus  
chown prometheus:prometheus /usr/local/bin/promtool
```

Запускаем и проверяем результат:

```
/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
```


Создаём сервис для работы с Prometheus

```
nano /etc/systemd/system/prometheus.service
```

Вставляем в файл сервиса следующее содержимое:

```
[Unit]
Description=Prometheus Service Netology Lesson 9.4
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
```

Передаём права на файл:

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

Тестируем сервис Prometheus

Прописываем автозапуск:

```
sudo systemctl enable prometheus
```

Запускаем сервис:

```
sudo systemctl start prometheus
```

Проверяем статус сервиса:

```
sudo systemctl status prometheus
```



Установка Node Exporter

Установка Node Exporter

Node Exporter помогает извлекать данные с хоста, на котором установлен.

Качаем архив с Node Exporter и извлекаем его:

```
wget  
https://github.com/prometheus/node_exporter/releases/download/v1.0.0/node_exporter-1.0.0.linux-amd64.tar.gz  
tar xvfz node_exporter-*.tar.gz
```

Переходим в появившуюся папку:

```
cd node_exporter-*.tar.gz  
./node_exporter
```

Можно проверить, что всё работает, перейдя по адресу:
`http://<ip_сервера>:9100/metrics`

Копируем Node Exporter в папку Prometheus

```
mkdir /etc/prometheus/node-exporter  
cp ./* /etc/prometheus/node-exporter
```

Передаём права на папку пользователю prometheus

```
chown -R prometheus:prometheus /etc/prometheus/node-exporter/
```

Создаём сервис для работы с Node Exporter

```
nano /etc/systemd/system/node-exporter.service
```

Вставляем в файл сервиса следующее содержимое:

```
[Unit]
Description=Node Exporter Lesson 9.4
After=network.target
[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/etc/prometheus/node-exporter/node_exporter
[Install]
WantedBy=multi-user.target
```

Тестируем сервис Prometheus

Прописываем автозапуск:

```
sudo systemctl enable node-exporter
```

Запускаем сервис:

```
sudo systemctl start node-exporter
```

Проверяем статус сервиса:

```
sudo systemctl status node-exporter
```

Добавляем Node Exporter в Prometheus

Редактируем конфиг Prometheus:

```
nano /etc/prometheus/prometheus.yml
```

Добавляем в scrape_config адрес экспортера:

```
scrape_configs:  
  - job_name: 'prometheus'  
    scrape_interval: 5s  
    static_configs:  
      - targets: ['localhost:9090', 'localhost:9100']
```

Перезапускаем Prometheus:

```
systemctl restart prometheus
```




Установка Grafana

Устанавливаем Grafana

Grafana необходима для того, чтобы визуализировать данные, получаемые Prometheus.

Скачиваем и устанавливаем DEB пакет:

```
wget https://dl.grafana.com/oss/release/grafana_7.0.3_amd64.deb  
dpkg -i grafana_7.0.3_amd64.deb
```

Включаем автозапуск и запускаем сервер Grafana:

```
systemctl enable grafana-server  
systemctl start grafana-server  
systemctl status grafana-server
```

Проверяем статус подключившись на адрес:

`https://<наш сервер>:3000`

Стандартный логин и пароль admin \ admin.



Интеграция Grafana и Prometheus

Добавляем дашборд в Grafana

- Переходим в раздел **Configuration > Data Sources** и нажимаем на Add data source.
- В появившемся списке выбираем Prometheus.

| <http://localhost:9090>

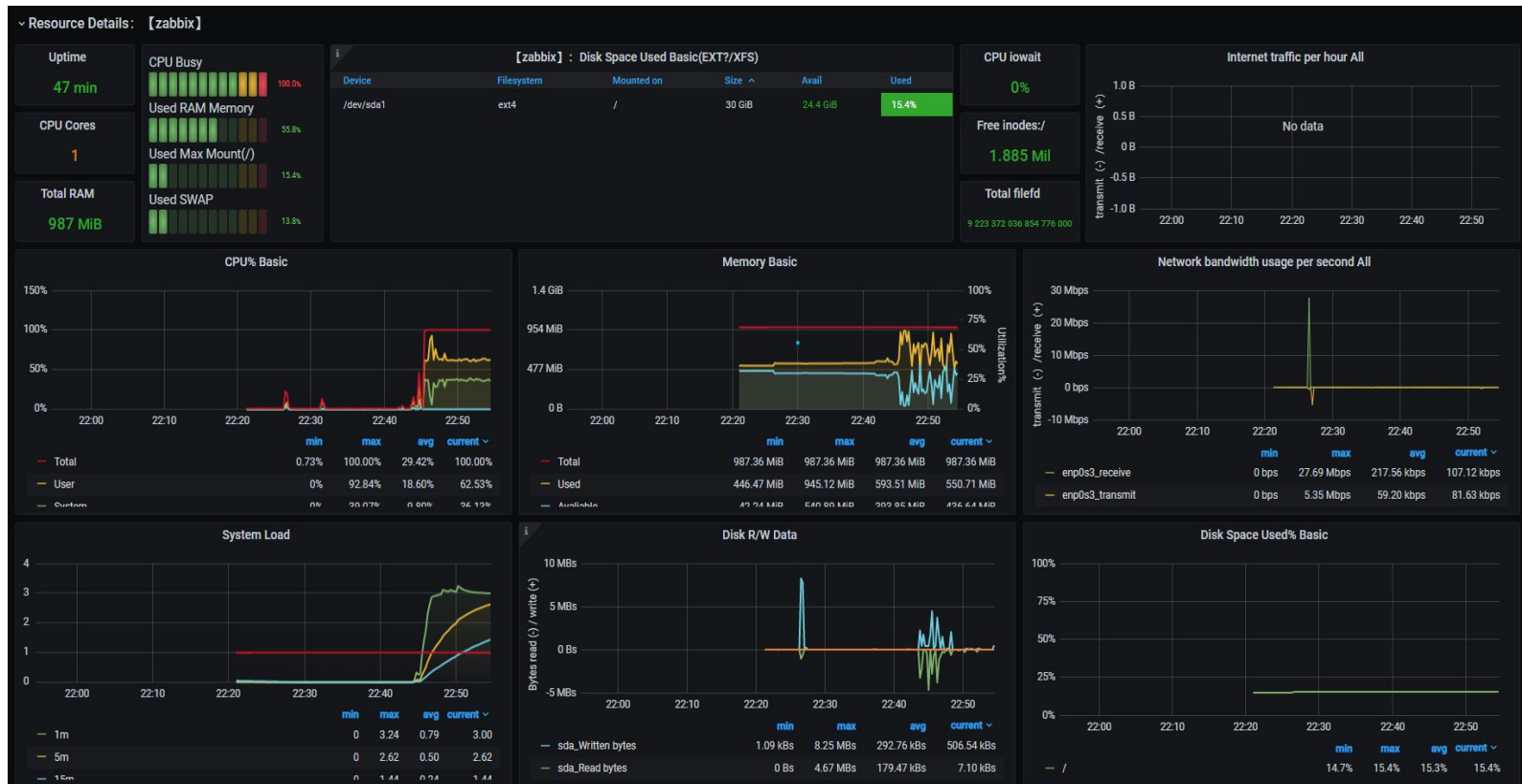
- Нажимаем кнопку Save & Test.
- На сайте grafana.com ищем [нужный дашборд](#) и копируем его ID.
- Переходим в раздел **[+] > Import** и в поле Import via grafana.com вбиваем скопированный ID или ссылку на дашборд.
- В выпадающем списке **VictoriaMetrics** выбираем Prometheus.
- Нажимаем кнопку **Import**.



Просмотр метрик с хоста, где стоит Prometheus

Просмотр метрик

Переходим в раздел **Dashboards > Manage** и выбираем только что импортированный дашборд.





Итоги

Итоги

Сегодня мы узнали как:

- установить Prometheus;
- установить Node Exporter;
- подключить Prometheus к Node Exporter;
- установить Grafana;
- подключить Prometheus к Grafana;
- Импортировать в Grafana шаблон для просмотра метрик, собранных Prometheus с помощью Node Exporter.





Домашнее задание

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачу можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты задача полностью**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Артур Сагутдинов



https://t.me/belf_igor

