

АХІОМ JDK

Инструкция по настройке мониторинга Axiom Runtime Container Pro через Grafana

Axiom JDK | Май 2025

Copyright © 2019-2025 Все права защищены АО "АКСИОМ" (АКСИОМ)

Программное обеспечение АКСИОМ содержит программное обеспечение с открытым исходным кодом. Дополнительная информация о коде сторонних разработчиков доступна на сайте https://axiomjdk.ru/third_party_licenses. Для дополнительной информации о том, как получить копию исходного кода, можно обратиться по адресу info@axiomjdk.ru.

ДАННАЯ ИНФОРМАЦИЯ МОЖЕТ ИЗМЕНЯТЬСЯ БЕЗ ПРЕДВАРИТЕЛЬНОГО УВЕДОМЛЕНИЯ. АКСИОМ ПРЕДОСТАВЛЯЕТ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ "КАК ЕСТЬ" БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, АКСИОМ ПРЯМО ОТКАЗЫВАЕТСЯ ОТ ВСЕХ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ПОДРАЗУМЕВАЕМЫМИ ГАРАНТИЯМИ ТОВАРНОЙ ПРИГОДНОСТИ И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

АКСИОМ НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА ЛЮБЫЕ КОСВЕННЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ШТРАФНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ, ИЛИ УБЫТКИ ОТ ПОТЕРИ ПРИБЫЛИ, ДОХОДА, ДАННЫХ ИЛИ ИСПОЛЬЗОВАНИЯ ДАННЫХ, ПОНЕСЕННЫЕ ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ, БУДЬ ТО В РЕЗУЛЬТАТЕ ДЕЙСТВИЯ ДОГОВОРА ИЛИ ДЕЛИКТА, ДАЖЕ ЕСЛИ АКСИОМ БЫЛО ПРЕДУПРЕЖДЕНО О ВОЗМОЖНОСТИ ТАКИХ УБЫТКОВ.

Использование любого программного продукта АКСИОМ регулируется соответствующим лицензионным соглашением, которое никоим образом не изменяется условиями данного уведомления. Программные продукты и фирменные наименования: Axiom JDK, Axiom JDK Pro, Axiom Runtime Container Pro, Axiom Linux, Libercat, Libercat Certified и АКСИОМ принадлежат АКСИОМ и их использование допускается только с разрешения правообладателя.

Товарный знак Linux® используется в соответствии с сублицензией от Linux Foundation, эксклюзивного лицензиата Линуса Торвальдса, владельца знака на всемирной основе. Java и OpenJDK являются товарными знаками или зарегистрированными товарными знаками компании Oracle и/или ее аффилированных лиц. Другие торговые марки являются собственностью их соответствующих владельцев и используются только в целях идентификации.

Содержание

1. Введение	5
2. Установка.....	6
Grafana.....	6
Windows	6
Linux	7
Prometheus.....	7
3. Конфигурация.....	8
Prometheus	8
4. Запуск	9
Grafana.....	9
Prometheus	9
сAdvisor	9
Axiom Runtime Container Pro	10
Проверка.....	10

5. Настройка Grafana	12
Data source.....	12
Импорт Dashboard	13
Использование.....	13
6. Описание графиков.....	15
Summary	15
CPU.....	15
Memory.....	15
Network	16
I/O	16
7. Добавление метрик.....	17



1. Введение

Данный документ описывает настройку и работу с дашбордом для мониторинга Axiom Runtime Container Pro с помощью Grafana и Prometheus. Дашборд содержит основные метрики, необходимые для мониторинга производительности и состояния контейнера. Метрики считываются при помощи cAdvisor.

2. Установка

Grafana

См. подробную инструкцию по установке на странице [Install Grafana](#).

Windows

Выполните следующие шаги, чтобы скачать и установить Grafana.

1. Перейдите на страницу [загрузки Grafana](#).
2. Выберите версию Grafana, которую вы хотите установить в списке **Version**. По умолчанию выбрана самая последняя версия Grafana.
3. Выберите выпуск в списке **Edition**.
 - Версия с открытым исходным кодом - Эта версия функционально идентична корпоративной версии.
 - Корпоративная версия - Функционально она идентична версии с открытым исходным кодом, но включает функции, которые вы можете разблокировать с помощью лицензии.
4. Выберите Windows. Чтобы использовать установщик Windows, выполните следующие действия:
 - a. Нажмите Download the installer, чтобы загрузить установщик.
 - b. Откройте и запустите установщик.
 - c. Следуйте инструкциям на экране окна установщика. Нажмите **Next** для перехода на следующий шаг.

Чтобы установить Grafana из архива, выполните следующие действия:

1. На странице [загрузки Grafana](#) выберите Windows и далее нажмите Download the zip file.
2. Извлеките ZIP-файл в любую папку.

При необходимости сначала разблокируйте файл следующим образом: щелкните загруженный файл правой кнопкой мыши, выберите "Свойства", установите флажок "Разблокировать" и нажмите "ОК".

См. страницу [Установка на Windows](#) для дополнительной информации.

Linux

Выполните следующие шаги, чтобы скачать и установить Grafana как отдельный пакет на Linux.

1. Перейдите на страницу [загрузки Grafana](#).
2. Выберите версию Grafana, которую вы хотите установить в списке **Version**. По умолчанию выбрана самая последняя версия Grafana.
3. Выберите выпуск в списке **Edition**.
 - Версия с открытым исходным кодом - Эта версия функционально идентична корпоративной версии.
 - Корпоративная версия - Функционально она идентична версии с открытым исходным кодом, но включает функции, которые вы можете разблокировать с помощью лицензии.
4. Нажмите Linux и выберите вашу версию Linux.
5. Скопируйте и вставьте код со страницы загрузки под необходимой версией Linux в командную строку и запустите. Например, чтобы скачать и установить Grafana как отдельный пакет на Linux выполните следующие команды:

```
wget https://dl.grafana.com/oss/release/grafana-11.5.1.linux-amd64.tar.gz
tar -zxvf grafana-11.5.1.linux-amd64.tar.gz
```

См. страницу [Установка на Linux](#) для дополнительной информации.

Prometheus

Чтобы установить Prometheus, выполните следующие шаги:

1. Откройте страницу [загрузки](#).
2. Для скачивания выберите и кликните на архив подходящий для вашей операционной системы.
3. Разархивируйте программу в каталог на диске. Например, чтобы разархивировать файл на Linux запустите следующую команду:

```
tar -zxvf prometheus-3.2.0-rc.1.linux-amd64.tar.gz
```

3. Конфигурация

Prometheus

Создайте файл конфигурации `prometheus.yml`:

```
global:
  scrape_interval: 5s
  evaluation_interval: 5s

scrape_configs:
  - job_name: 'cadvisor'
    scheme: http
    static_configs:
      - targets: ['localhost:8080']
```

Обратите внимание:

- Значение параметра `job_name` "cadvisor" будет в дальнейшем использоваться в Grafana для считывания метрик.
- В поле `targets` указан endpoint cAdvisor, порт 8080 будет повторно указан при запуске cAdvisor (следующий шаг).



Примечание:

Официальная документация по конфигурационному файлу Prometheus: [Configuration file](#).

4. Запуск

Grafana

Запустите Grafana:

```
grafana-server --homepath='/usr/share/grafana'  
--config='/usr/share/grafana/conf/defaults.ini'
```



Примечание:

Официальная документация по запуску Grafana: [Запуск Grafana](#).

Prometheus

Запустите Prometheus, указав конфигурационный файл `prometheus.yaml` в параметре `--config.file`:

```
prometheus --config.file=prometheus.yaml --web.enable-admin-api
```



Примечание:

Официальная документация по запуску Prometheus: [Запуск Prometheus](#).

cAdvisor

Запустите cAdvisor (поставляется в виде Docker image, готового для запуска):

```
sudo docker run \  
  --volume=/:/rootfs:ro \  
  --volume=/var/run:/var/run:ro \  
  --volume=/sys:/sys:ro \  
  --volume=/var/lib/docker/:/var/lib/docker:ro \  
  --volume=/dev/disk/:/dev/disk:ro \  
  --publish=8080:8080 \  
  --
```

```
--detach=true \  
--name=cadvisor \  
--privileged \  
--device=/dev/kmsg \  
gcr.io/cadvisor/cadvisor:latest
```

**Примечание:**

Официальная документация по cAdvisor: [cAdvisor](#).

Axiom Runtime Container Pro

Запустите Axiom Runtime Container Pro. Пример:

```
docker run -it --name axiom-runtime-container-pro  
cr.yandex/crpgua9ba7h8red2hulb/axiom-runtime-container-pro
```

- Параметр "--name" указывать не обязательно. Он позволяет присвоить контейнеру фиксированное имя для удобства отображения на графиках.

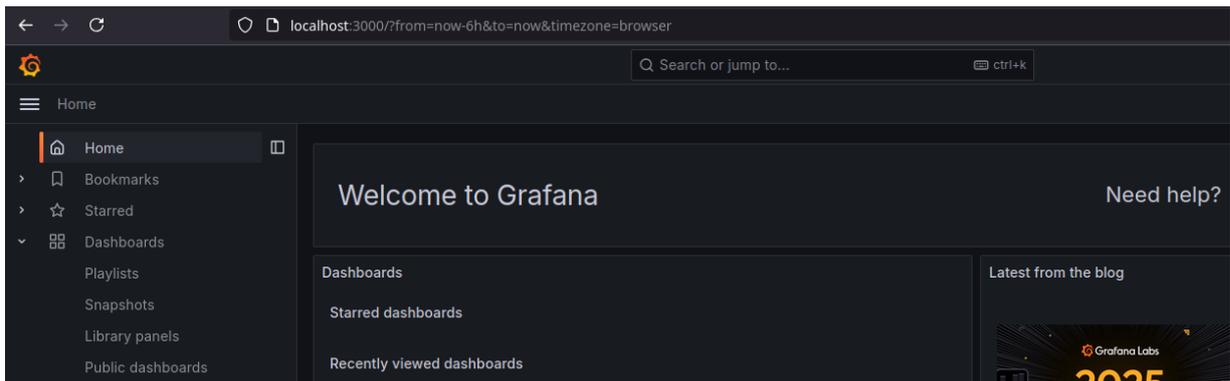
**Примечание:**

Официальная документация по запуску Axiom Runtime Container Pro: [Запуск Axiom Runtime Container Pro](#).

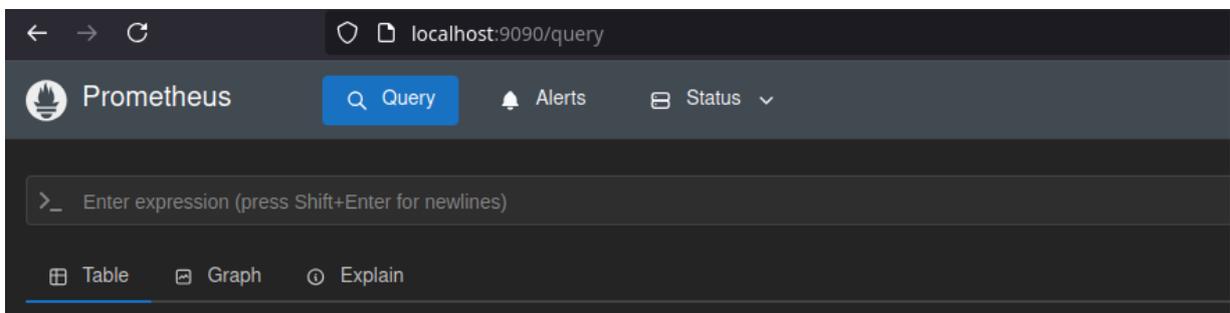
Проверка

На данном шаге должны быть доступны следующие URL:

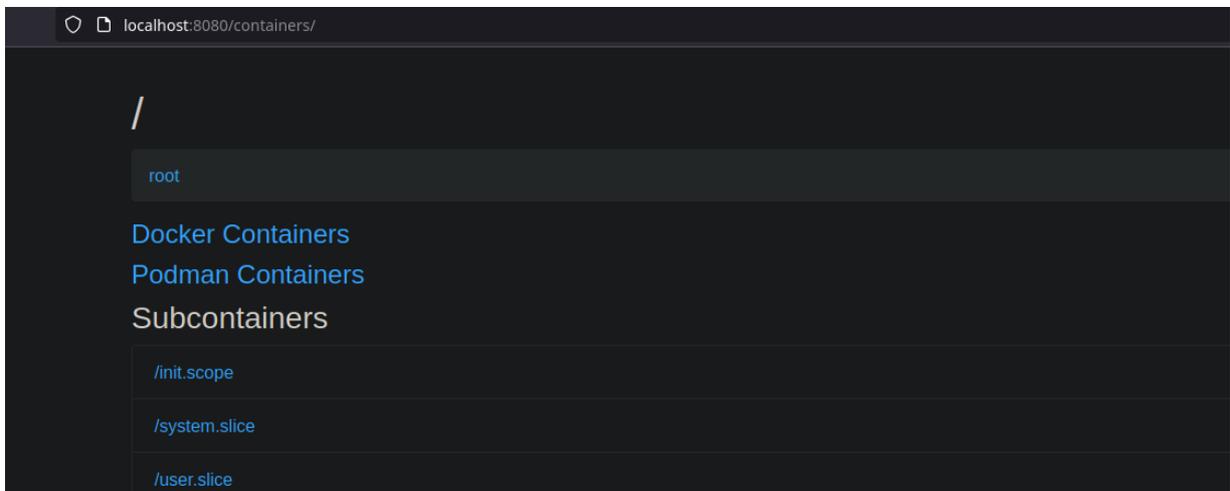
- Grafana: <http://localhost:3000>



- Prometheus: <http://localhost:9090>



- cAdvisor: <http://localhost:8080/metrics>



5. Настройка Grafana

Data source

1. Добавьте Prometheus data source, нажав **Add new data source** на данной странице: <http://localhost:3000/connections/datasources/prometheus>

Settings Dashboards

Name Default

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

Fields marked with * are required

Connection

Prometheus server URL *

2. В поле **Prometheus server URL** укажите <http://localhost:9090> и нажмите **Save & test**. Должен отображаться успешный статус "Successfully queried the Prometheus API."

✓ Successfully queried the Prometheus API.

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the Explore view.

Delete Save & test

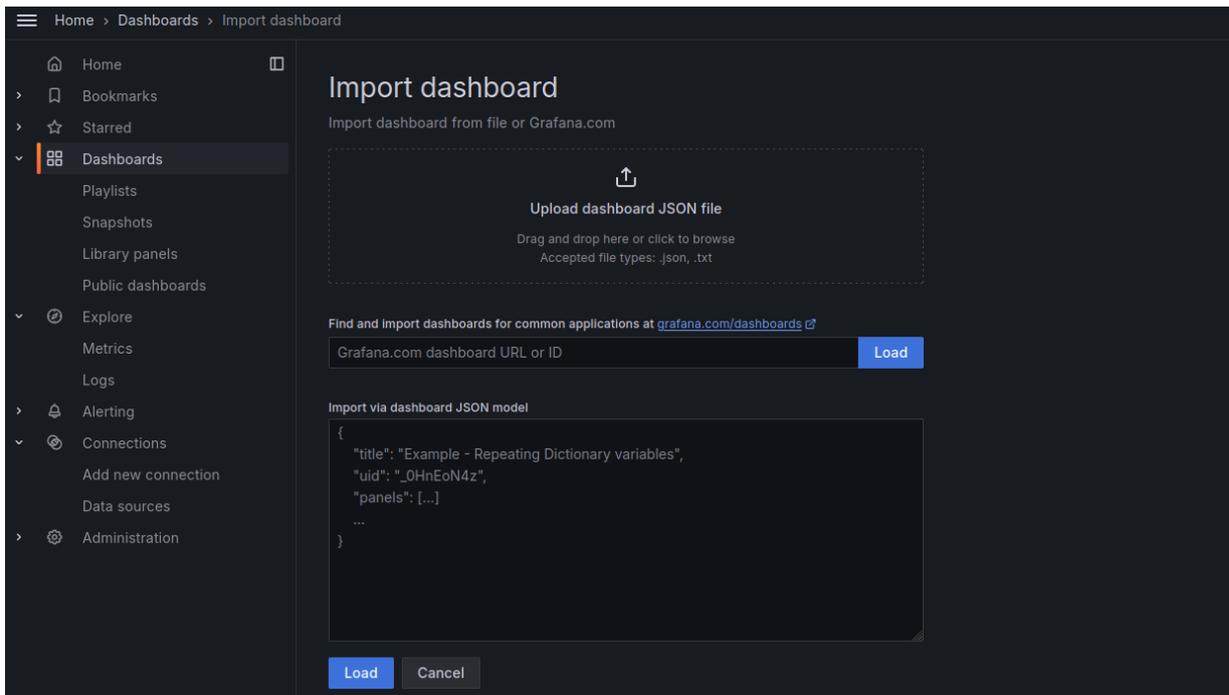


Примечание:

Официальная документация по конфигурации Prometheus data source: [Configure Prometheus](#).

Импорт Dashboard

1. Откройте страницу импорта Dashboards: <http://localhost:3000/dashboard/import>
2. Скачайте и загрузите JSON файл дашборда, нажав **Upload dashboard JSON file**: <https://download.axiomjdk.ru/grafana/axiom-runtime-container-pro-dashboard.json>

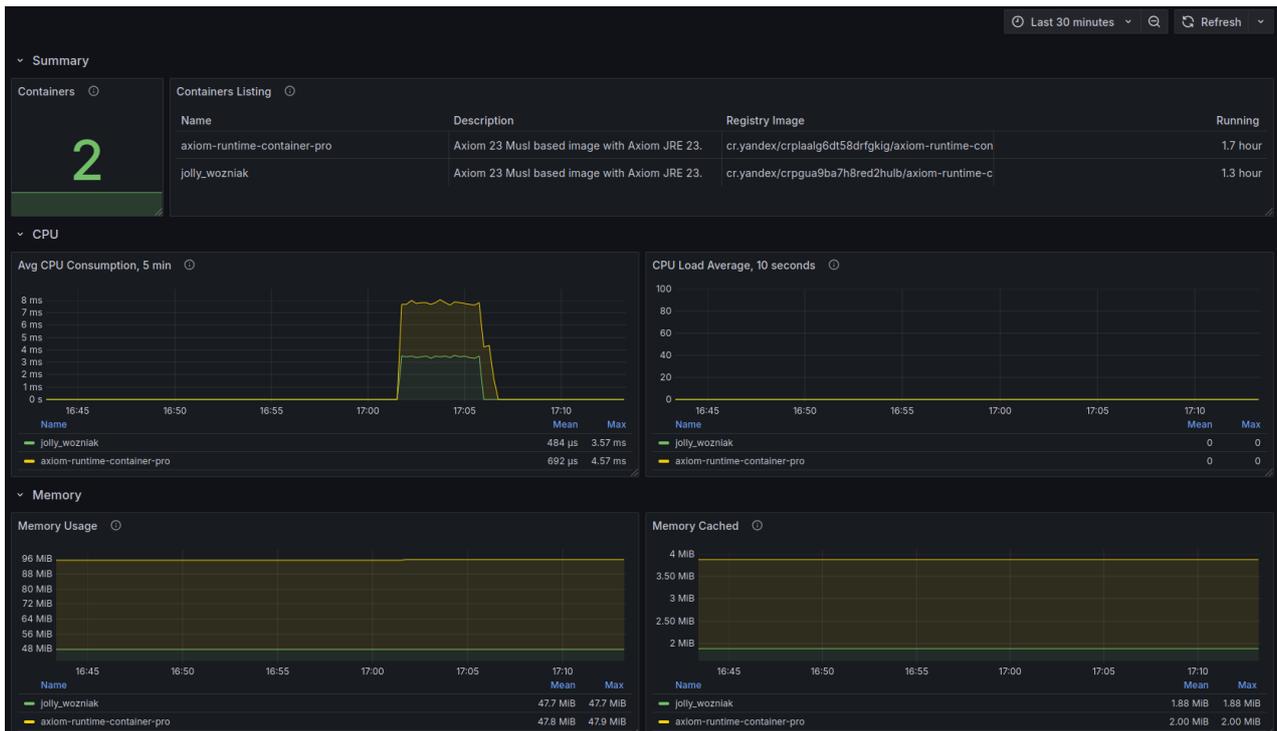


Примечание:

Официальная страница Grafana по импорту дашбордов: [Import dashboards](https://grafana.com/docs/grafana/latest/dashboards/import-dashboards/).

Использование

Откройте дашборд. В нем должны отображаться графики с данными: <http://localhost:3000/d/6cfc052e/axiom-runtime-container-pro>



Примечание:

Официальная документация Grafana по настройке дашбордов: [Dashboards](#).

6. Описание графиков

Summary

Поле **Containers** отображает количество запущенных экземпляров Axiom Runtime Container Pro. В поле используется метрика `container_last_seen`.

Таблица **Containers Listing** отображает каждый запущенный экземпляр с его именем, описанием, образом и временем работы. В таблице используется метрика `container_start_time_seconds`.

CPU

График **Avg CPU Consumption, 5 min** показывает среднесекундную частоту увеличения потребления CPU на интервале 5 минут. График полезен для мониторинга нагрузки на CPU, создаваемой контейнером, и диагностики производительности. В графике используется метрика `container_cpu_usage_seconds_total`.

График **CPU Load Average** показывает значение load average, создаваемое контейнером на интервале 10 секунд. График полезен для мониторинга нагрузки на CPU, создаваемой контейнером. В графике используется метрика `container_cpu_load_average_10s`.

Memory

График **Memory Usage** показывает суммарное количество памяти, используемое контейнером. График полезен для диагностики проблем с памятью и производительностью. В графике используется метрика `container_memory_usage_bytes`.

График **Memory Cached** показывает кеш памяти (количество памяти для кеширования часто используемых данных). Полезен для диагностики проблем с памятью и определения лимитов использования памяти (например, флаг `--memory`). В графике используется метрика `container_memory_cache`.

График **Swap Usage** показывает количество памяти из области swap, использованной контейнером. Полезен для диагностики проблем с памятью и определения лимитов использования памяти (например, с такими флагами, как `--memory-swap` и `--memory-swappiness`). В графике используется метрика `container_memory_swap`.

График **Allocation Failures** показывает среднесекундную скорость увеличения количества отказов выделения памяти за последние 5 минут. График полезен для диагностики проблем с

памятью. В графике используется метрика `container_memory_failures_total`.

Network

График **Received Network Traffic** показывает суммарное количество сетевого трафика, полученного контейнером. График полезен для анализа сетевой нагрузки и выявления проблем с сетью. В графике используется метрика `container_network_receive_bytes_total`.

График **Sent Network Traffic** показывает суммарное количество сетевого трафика, отправленного контейнером. График полезен для анализа сетевой нагрузки и выявления проблем с сетью. В графике используется метрика `container_network_transmit_bytes_total`.

График **Network Errors** показывает суммарное количество сетевых ошибок при отправке и получении данных. График полезен для выявления проблем с сетью (например, проблемы с сетевой политикой или таблицами маршрутизации). В графике используются метрики `container_network_receive_errors_total` и `container_network_transmit_errors_total`.

График **Packets Dropped** показывает суммарное количество отброшенных сетевых пакетов при отправке и получении данных. График полезен для анализа сетевого поведения и выявления проблем с сетью (таких, как перегрузка сети или проблем, связанных с фаерволом и конфигурацией сети). В графике используются метрики `container_network_transmit_packets_dropped_total` и `container_network_receive_packets_dropped_total`.

I/O

График **File System Usage** показывает суммарный дисковый объем данных, занятый контейнером (не считая слоев самого образа). График полезен для отслеживания использования памяти и диагностики проблем с ней. В графике используется метрика `container_fs_usage_bytes`.

График **Avg Read / Write Frequency, 5m** показывает среднесекундную скорость увеличения количества операций чтения/записи за последние 5 минут. График полезен для анализа использования диска и диагностики проблем избыточного использования диска. В графике используются метрики `container_fs_reads_total` и `container_fs_writes_total`.

График **Inodes Number** показывает число суммарное число inodes в файловой системе контейнера, а также число свободных inodes. График можно использовать для диагностики проблем, связанных с созданием новых файлов и директорий. В графике используются метрики `container_fs_inodes_free` и `container_fs_inodes_total`.

7. Добавление метрик

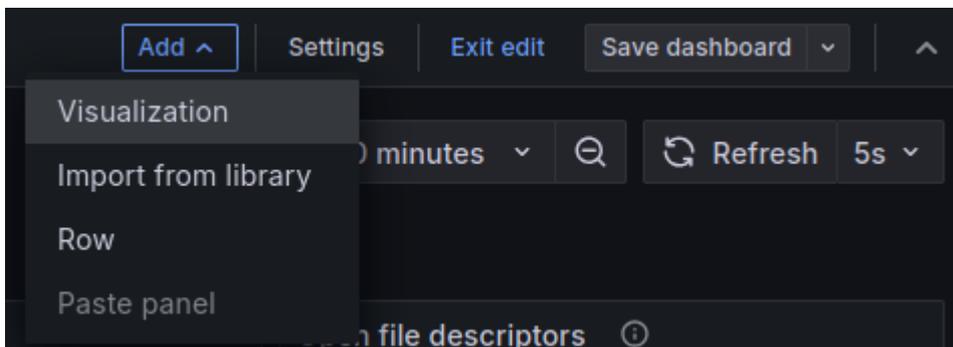
Полный список метрик, доступных для использования в Grafana, представлен на странице cAdvisor: <http://localhost:8080/metrics>

Метрики представлены в следующем формате:

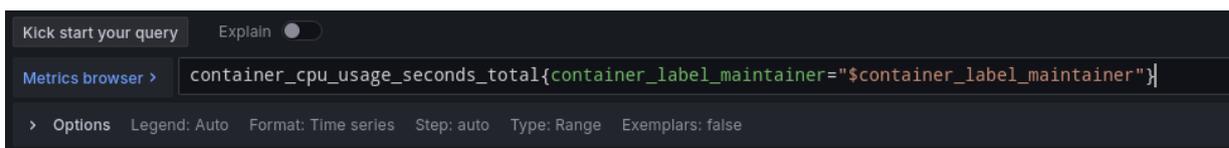
```
# HELP container_network_receive_bytes_total Cumulative count of bytes received
# TYPE container_network_receive_bytes_total counter
container_network_receive_bytes_total{container_label_maintainer="",container_label_org_opencontainers_image_authors="",container_label_org_opencontainers_image_description="",id="/",image="",interface="enp0s31f6",name=""} 0
1735289569756
container_network_receive_bytes_total{container_label_maintainer="",container_label_org_opencontainers_image_authors="",container_label_org_opencontainers_image_description="",id="/",image="",interface="tun0",name=""} 7.3779021e+07
1735289569756
```

Для добавления графика:

1. Откройте дашборд и нажмите в правом верхнем углу **Add > Visualisation**:



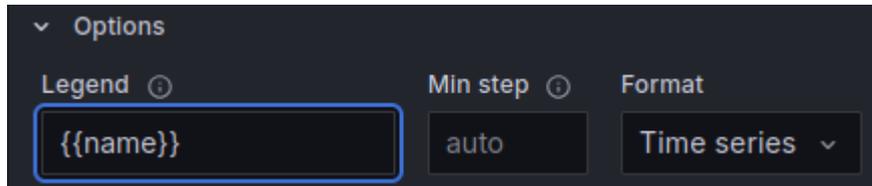
2. Добавьте метрику в поле **Enter a PromQL query**. Например, для добавления метрики `container_cpu_usage_seconds_total` укажите:



Обратите внимание, что значение `{container_label_maintainer=\"$container_label_maintainer\"}`, добавленное к

названию метрики в поле **Enter a PromQL query**, позволяет отобразить на графике Axiom Runtime Container Pro, исключая любые другие Docker контейнеры, запущенные в системе.

3. Для улучшения отображения параметров при наведении на график, выберите `Options > Legend > Custom` и укажите параметр в двойных фигурных скобках, например `{{name}}`:



Для поиска доступных параметров найдите метрику на [странице cAdvisor](#) или укажите её в качестве запроса в [Prometheus](#), когда cAdvisor и Prometheus запущены.



Совет:

Дополнительную информацию о метриках можно найти на страницах документации cAdvisor: [cAdvisor docs](#).

Для полной информации о панелях и визуализациях Grafana, см. [официальную документацию](#).



Инструкция по настройке мониторинга Axiom Runtime Container Pro через Grafana