

УТВЕРЖДЕН  
643.72410666.00067-07 98 01-ЛУ

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ  
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 28.  
Поддержка мониторинга СУБД

643.72410666.00067-07 98 01-28

Листов 57

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## АННОТАЦИЯ

В документе приведены сведения, необходимые для установки и эксплуатации компонентов, предназначенных для мониторинга СУБД:

- Компонент «node\_exporter». Версия компонента – 1.7.0;
- Компонент «postgres\_exporter». Версия компонента – 0.15.0;
- Компонент «sql\_exporter». Версия компонента – 0.13.0.
- Система «Prometheus». Версия системы – 3.5.0.
- Утилита «Alertmanager». Версия компонента – 0.27.0.

Настоящее руководство предназначено для администраторов СУБД.



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 5.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 6.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\6\bin»;
- ОС Linux – «/usr/jatoba-6/bin».



### Важная информация

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

Степени важности примечаний, применяемые в документе:



**Важная информация** – указания, требующие особого внимания



**Дополнительная информация** – указания, позволяющие упростить работу с изделием

## СОДЕРЖАНИЕ

1. Назначение компонентов.....	5
1.1. Условия применения.....	5
1.2. Ограничения по эксплуатации.....	5
2. Архитектура системы мониторинга.....	6
3. Установка и настройка целевых СУБД.....	8
3.1. Установка СУБД.....	8
3.2. Настройка конфигурационных файлов.....	8
3.3. Установка расширения «pg_stat_statements».....	8
4. Установка экспортера «jatoba*_node_exporter».....	11
5. Установка экспортера «jatoba*_postgres_exporter».....	15
5.1. Установка утилиты и службы «jatoba*_postgres_exporter».....	15
5.2. Создание пользователя СУБД «postgres_exporter».....	16
5.3. Настройка переменных окружения.....	16
5.4. Запуск утилиты «postgres_exporter».....	18
6. Установка экспортера «jatoba*_sql_exporter».....	21
6.1. Установка утилиты и службы «sql_exporter».....	21
6.2. Настройка переменных окружения.....	22
6.3. Создание пользователя СУБД «sql_exporter».....	22
6.4. Настройка параметров экспортера и подключения к БД в файле «sql_exporter.yml».....	23
6.5. Запуск утилиты «jatoba*_sql_exporter».....	24
7. Система «Prometheus».....	27
7.1. Установка системы «Prometheus».....	27
7.2. Конфигурация системы «Prometheus».....	28
7.2.1. Примеры блока «postgres-exporter».....	29
7.2.2. Примеры блока «sql-exporter».....	30
7.2.3. Примеры блока «node-exporter».....	31
7.3. Запуск системы «Prometheus».....	33
8. Утилита «Alertmanager».....	37
8.1. Установка утилиты и службы «alertmanager».....	37
8.2. Настройка параметров утилиты файле «alertmanager.yml».....	38
8.3. Запуск утилиты «alertmanager».....	39
9. Подключение к JDS.....	41
9.1. Настройка SSH-соединения.....	42
9.2. Конфигурирование JDS.....	43
9.3. Настройка связки системы «Prometheus» и утилиты «Alertmanager».....	45
10. Настройка экспортёров для компонента ja_Hipe_Cluster.....	47
10.1. Параметры стенда.....	47

10.2. Конфигурирование системы «Prometheus» .....	50
11. Обновление системы «Prometheus».....	53
Термины и определения .....	55
Перечень сокращений.....	56

## 1. НАЗНАЧЕНИЕ КОМПОНЕНТОВ

Компонент «node\_exporter» – программный инструмент, предназначенный для мониторинга и сбора метрик с различных компонентов в Linux-подобных ОС.

Компонент собирает и экспортирует различные метрики, такие как загрузка процессора, использование памяти, статистика сети, системные вызовы и т.д. Собранные данные могут быть отправлены на сервер «Prometheus» для визуализации и анализа.

Компонент «postgres\_exporter» – инструмент для сбора и экспорта метрик PostgreSQL, таких как статистика по базе данных, нагрузка на сервер, количество запросов и т.д. Он разработан для работы с PostgreSQL и предоставляет данные в формате, удобном для системы «Prometheus». С помощью «postgres\_exporter» можно отслеживать производительность PostgreSQL, выявлять проблемы и оптимизировать настройки базы данных.

Компонент «sql\_exporter» – инструмент для экспорта данных из SQL-запросов в формат, удобный для анализа и визуализации. Он позволяет получать информацию о структуре таблиц, данных, индексах, статистике и других параметрах базы данных. Полезен для анализа производительности системы, выявления проблем и оптимизации запросов.

### 1.1. Условия применения

Компоненты могут использоваться с СУБД «Jatoba» версий 5.x и выше, под управлением операционных систем GNU/Linux.

### 1.2. Ограничения по эксплуатации

Для подключения целевой СУБД к компоненту «Jatoba data safe» требуется указывать IP-адрес в строке подключения утилит к СУБД и не использовать параметр «localhost».

Символы «коммерческое эт» «@», «амперсанд» «&», «равно» «=», «вопросительный знак» «?» и «двоеточие» «:», не рекомендуется использовать в именах пользователей и в паролях, для исключения ошибки в строке подключения.

Эти символы используются для разделения параметров строки подключения.

Ограничений по совместимости с другими компонентами нет.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

## 2. АРХИТЕКТУРА СИСТЕМЫ МОНИТОРИНГА

Архитектура системы мониторинга основана на том, что:

- на серверах целевых СУБД устанавливается экспортер «node\_exporter» (см. р. 4);
- на целевых СУБД с предустановленным расширением «pg\_stat\_statements» устанавливаются утилиты сбора метрик, такие как:
  - экспортер «postgres\_exporter» (см. р. 5);
  - экспортер «sql\_exporter» (см. р. 6);
- система «Prometheus» собирает их в своём хранилище (см. р. 7);
- компонент «Jatoba data safe» использует данные хранилища «Prometheus» для отображения их в разделе «Мониторинг»;
- утилита «Alertmanager» обеспечивает контроль над пороговыми значениями и рассылку уведомлений (см. р. 8).

В зависимости от количества СУБД, подключенных к мониторингу и общей нагрузки, система «Prometheus» может быть установлена на отдельном сервере. В этом случае «JDS» будет получать данные по сети, что увеличит нагрузку на неё.

Целесообразнее компонент JDS и систему «Prometheus» устанавливать на одном сервере. Такая конфигурация сделает данный сервер полноценным сервером мониторинга и безопасности.

Для каждой наблюдаемой СУБД должны быть настроены все экспортёры.

В рассматриваемом примере на ОС Ubuntu 22.04 используются параметры сети и программного обеспечения, приведенные в таблице 2.1.

Таблица 2.1 – Конфигурация стенда

№	Имя сервера	IP-адрес	ПО	Port	Роль
1	u602doc-jds01	10.116.102.41/24			Сервер мониторинга
1.1			JDS		
1.2			Prometheus	9090	
			Alert manager	9093 22	
2	u602doc-pgp01	10.116.102.45/24			Целевая СУБД
2.1			pg_stat_statements		
2.2			node_exporter	9100	
2.3			postgres_exporter	9187	
2.4			sql_exporter	9399	

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

№	Имя сервера	IP-адрес	ПО	Port	Роль
3	u602doc-ldap01	10.116.102.47/24			Целевая СУБД
3.1			pg_stat_statements		
3.2			node_exporter	9100	
3.3			postgres_exporter	9187	
3.4			sql_exporter	9399	

Схема стенда представлена на рисунке 2.1.

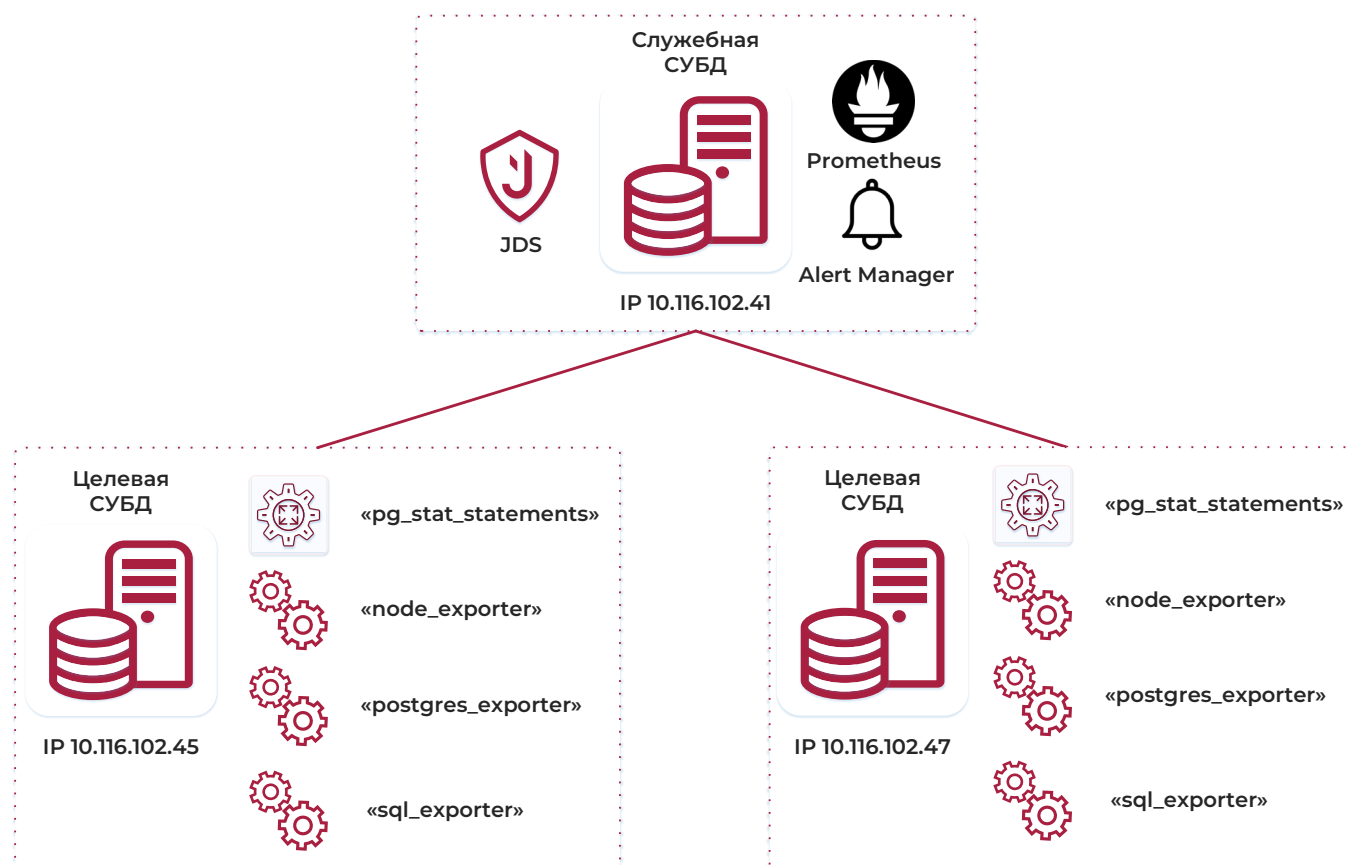


Рисунок 2.1 – Схема стенда

### 3. УСТАНОВКА И НАСТРОЙКА ЦЕЛЕВЫХ СУБД

#### 3.1. Установка СУБД

Установка СУБД «Jatoba» выполняется от имени пользователя, обладающего административными привилегиями в системе, в соответствии с документом «Защищенная система управления базами данных «Jatoba». Руководство по установке».

#### 3.2. Настройка конфигурационных файлов

Целевые СУБД должны быть настроены на приём подключений

В конфигурационном файле «postgresql.conf», в разделе «CONNECTIONS AND AUTHENTICATION» раскомментирован и установлен параметр:

```
listen_addresses = '*'
```

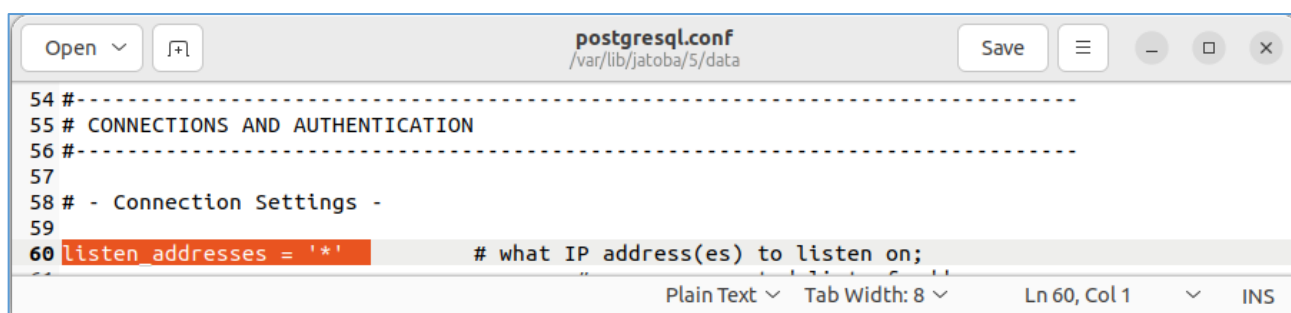


Рисунок 3.1 - Конфигурационный файл «postgresql.conf»

В конфигурационном файле «pg\_hba.conf» разрешены подключения к СУБД в параметре:

host	all	all	all	all	md5
------	-----	-----	-----	-----	-----

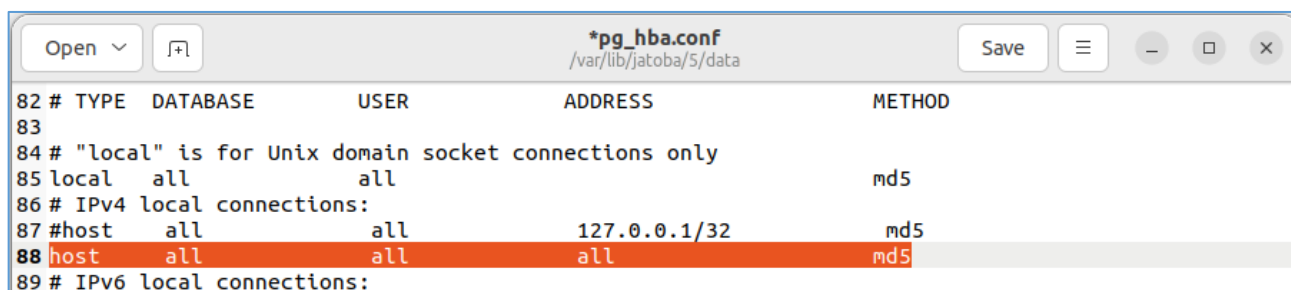


Рисунок 3.2 – Конфигурационный файл «pg\_hba.conf»

#### 3.3. Установка расширения «pg\_stat\_statements»

На каждой целевой СУБД должно быть установлено расширение «pg\_stat\_statements».



Для установки расширения «pg\_stat\_statements» потребуется:

- В конфигурационном файле «postgresql.conf», в разделе «Shared Library Preloading» для последующей загрузки расширения установить параметр:

```
shared_preload_libraries = 'pg_stat_statements'
```



Рисунок 3.3 – Строка загрузки расширения в конфигурационном файле СУБД «postgresql.conf»

- В разделе «STATISTICS» – «Monitoring» раскомментировать строку и добавить параметры:

```
compute_query_id = on
```

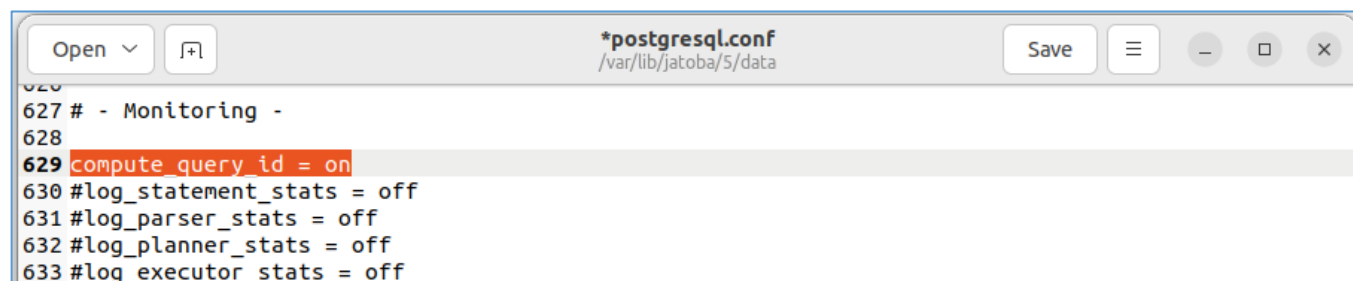


Рисунок 3.4 – Строка параметра «compute\_query\_id» в конфигурационном файле СУБД «postgresql.conf»

- В разделе «CUSTOMIZED OPTIONS» добавить параметры:

```
pg_stat_statements.max = 10000  
pg_stat_statements.track = all
```

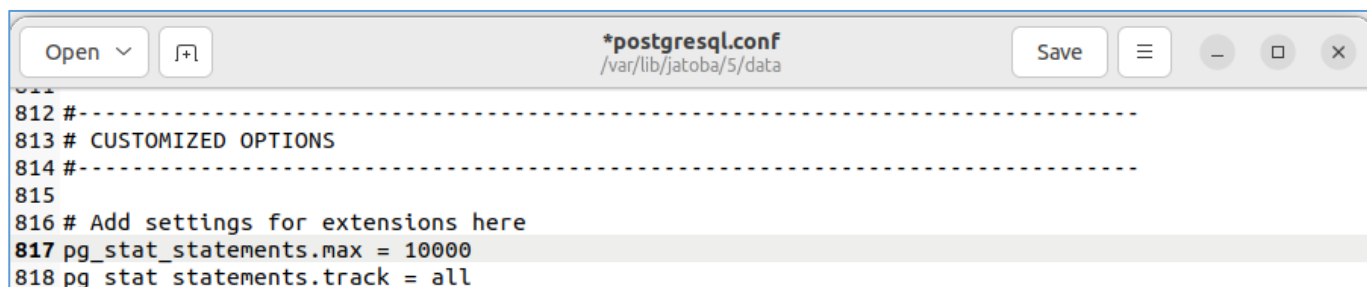


Рисунок 3.5 – Параметры статистики в конфигурационном файле СУБД «postgresql.conf»

Сохранить конфигурационный файл «postgresql.conf» и перезагрузить СУБД.

Расширение «pg\_stat\_statements» устанавливается при помощи SQL-команды:

```
CREATE EXTENSION pg_stat_statements;
```

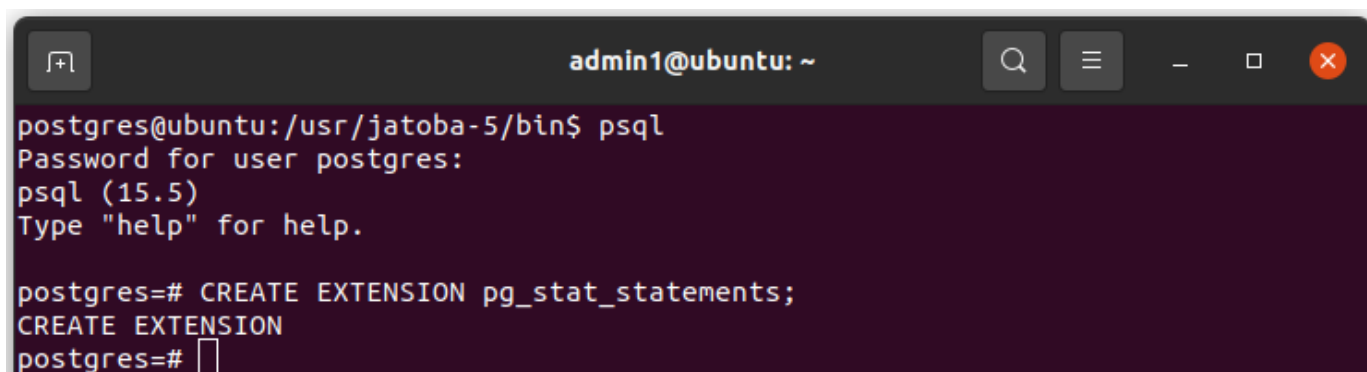


Рисунок 3.6 – Создание расширения

#### 4. УСТАНОВКА ЭКСПОРТЕРА «JATOBА\*\_NODE\_EXPORTER»

Экспортер «jatoba\*\_node\_exporter» должен быть установлен на всех целевых СУБД.

Экспортер позволяет снимать различные метрики с Linux-подобных операционных систем. Это агент, который передает серверу «Prometheus» аппаратные и программные показатели работы GNU/Linux.

Установка пакета выполняется в соответствии с Руководством по установке, из локального репозитория командой:

```
apt-get install jatoba<ver>-node-exporter
```

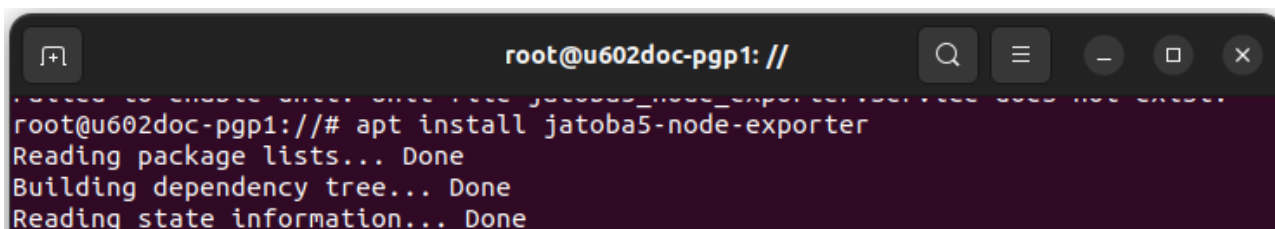


Рисунок 4.1 – Установка пакета «jatoba\*\_node\_exporter»

В результате установки пакета будет создан пользователь ОС «node\_exporter\_usr», от которого будет производиться запуск утилиты.

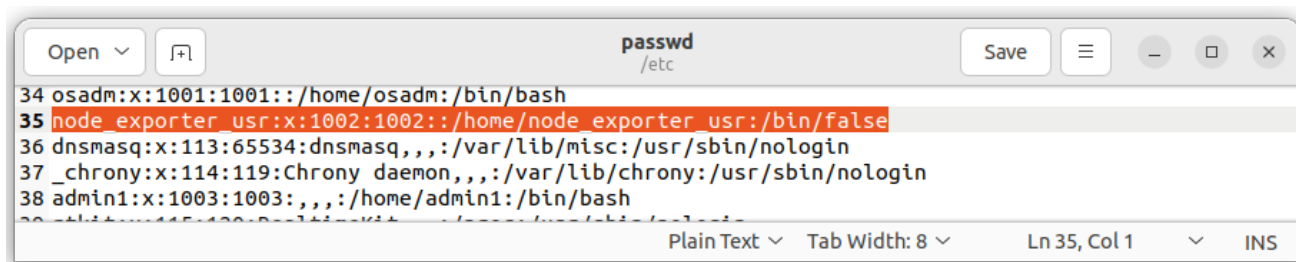


Рисунок 4.2 -пользователя «node\_exporter\_usr»

У данного пользователя нет интерактивной оболочки для входа.

Автоматически будет создан файл конфигурации сервиса по адресу:

```
/usr/lib/systemd/system/jatoba5_node_exporter.service
```

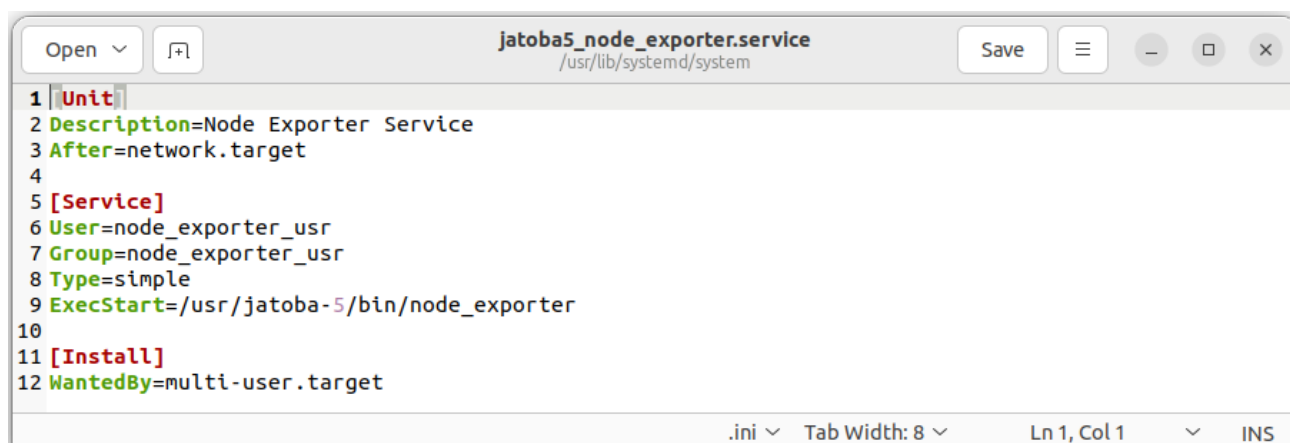


Рисунок 4.3 – Содержание конфигурационного файла

Далее требуется запустить службу экспортера, включить ее в автозапуск и проверить статус работы:

```

# systemctl enable jatoba<ver>_node_exporter
# systemctl start jatoba<ver>_node_exporter
# systemctl status jatoba<ver>_node_exporter
  
```

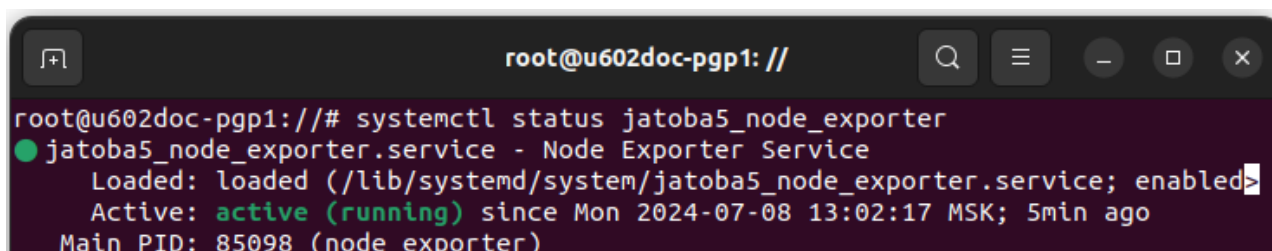


Рисунок 4.4 - Проверка сервиса «jatoba\*\_node\_exporter»

Чтобы проверить статус работы экспортера нужно в браузере открыть веб-интерфейс экспортера:

```

# localhost:9100
# http://0.0.0.0:9100
  
```

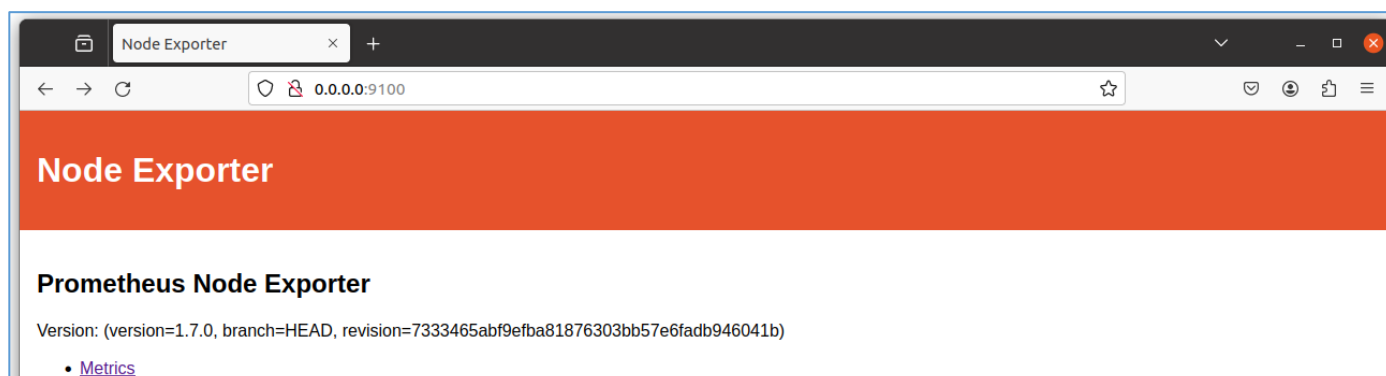


Рисунок 4.5 – Веб-интерфейс утилиты «node\_exporter»

В рассматриваемом примере на целевой СУБД:

- u602doc-pgrp01 IP - 10.116.102.45 веб-интерфейс утилиты «node\_exporter» проверяется по URL:

```
# http://10.116.102.45:9100
```

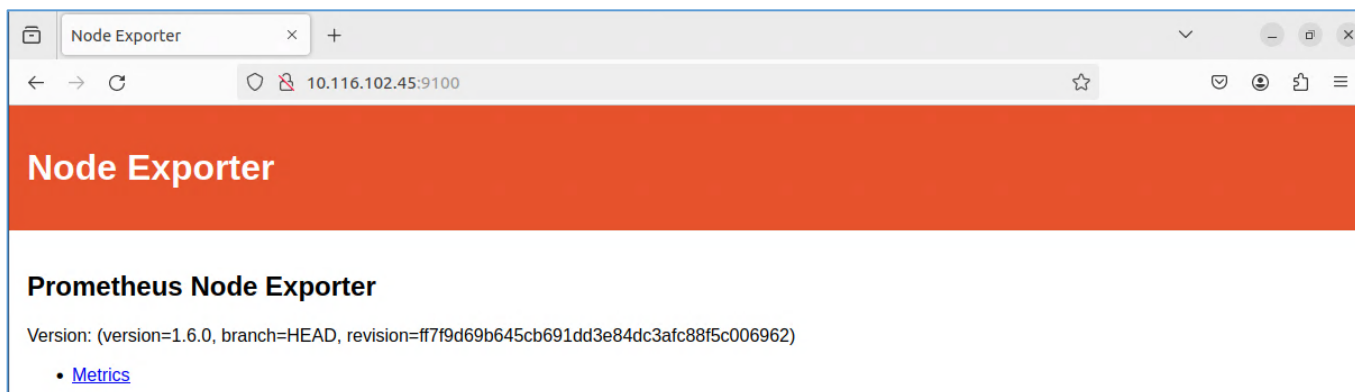


Рисунок 4.6 – Веб-интерфейс утилиты «node\_exporter» на целевой СУБД u602doc-pgrp01 IP - 10.116.102.45

- u602doc-ldap01 IP-10.116.102.47 веб-интерфейс утилиты «node\_exporter» проверяется по URL:

```
# http://10.116.102.47:9100
```



Рисунок 4.7 – Веб-интерфейс утилиты «node\_exporter» на целевой СУБД u602doc-ldap01 IP-10.116.102.47

По умолчанию экспортер использует все доступные коллекторы метрик.

Состав снимаемых метрик отображается на странице:

```
localhost:9100/metrics
```

При необходимости может быть изменен состав используемых коллекторов с помощью опций командной строки:

```
./jatoba<ver>_node_exporter --[no-]collector.netdev --[no-]  
collector.netstat
```

Если необходимо изменить значения адреса веб-интерфейса (:9100), node\_exporter запускается с опцией --web.listen-address:

```
./jatoba<ver>_node_exporter --web.listen-address=:9101
```



Изменение состава метрик либо адреса веб-интерфейса целесообразнее сохранить в файле сервиса «node\_exporter.service». Иначе при перезагрузке ОС настройки компонента вернутся к изначальным, хранящимся в файле сервиса

Ручной запуск утилиты производится командой:

```
./jatoba<ver>_node_exporter
```

Никакой конфигурации экспортера не требуется.

## 5. УСТАНОВКА ЭКСПОРТЕРА «JATOBA\*\_POSTGRES\_EXPORTER»

Экспортер «jatoba\*postgres\_exporter» должен быть установлен на всех целевых СУБД, и в той же БД в которой установлено расширение pg\_stat\_statements.

С помощью данного экспортера снимаются метрики с сервера PostgreSQL (Jatoba). Это агент, написанный на языке Golang, подключающийся к заданному источнику данных (БД) и по запросу сервера «Prometheus» возвращающий ему значения метрик. Состав метрик заранее предопределен и их значения вычисляются с помощью фиксированных SQL-запросов.

### 5.1. Установка утилиты и службы «jatoba\*\_postgres\_exporter»

Установка пакета выполняется в соответствии с Руководством по установке, из локального репозитория командой:

```
apt-get install jatoba<ver>-postgres-exporter
```

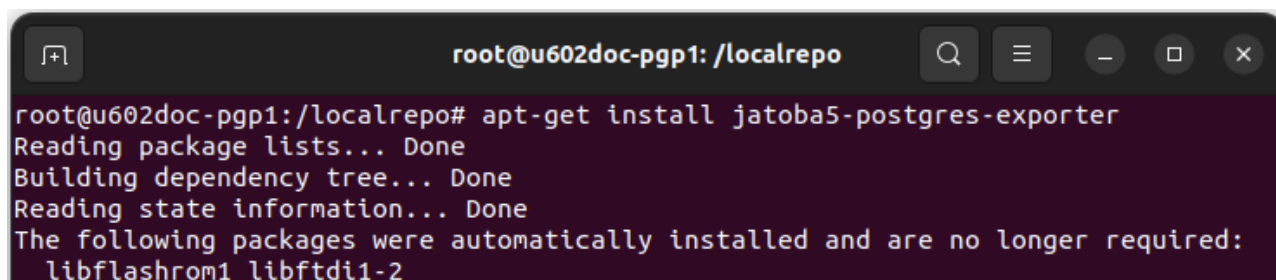


Рисунок 5.1 – Установка пакета jatoba\*-postgres-exporter

В результате установки пакета будет создан:

- файл запуска по адресу:

```
/usr/jatoba-<ver>/bin/postgres_exporter
```

- конфигурационный файл по адресу:

```
/usr/jatoba-<ver>/monitoring/default/postgres_exporter
```

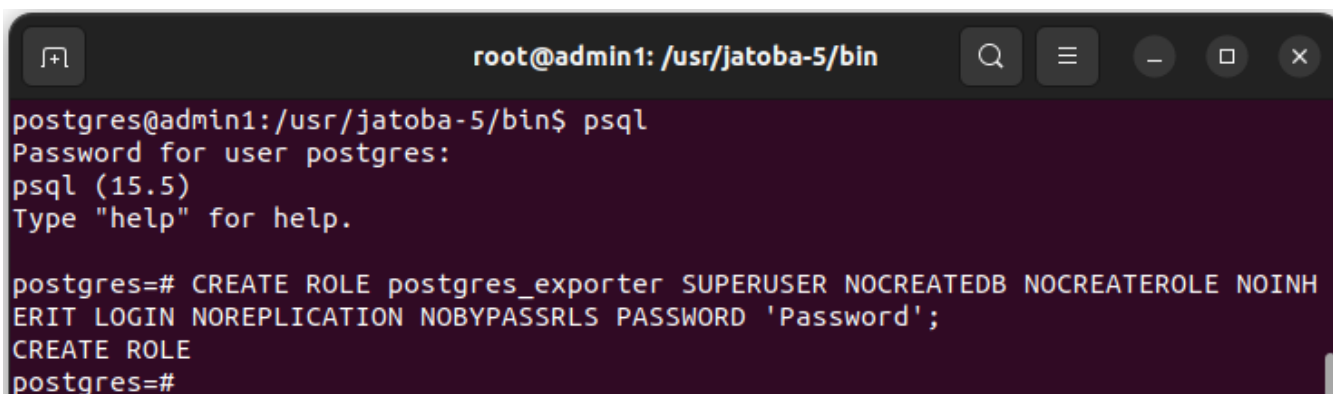
- пользователь ОС «postgres\_exporter\_usr», от которого будет производиться запуск сервиса.

У данного пользователя нет интерактивной оболочки для входа.

## 5.2. Создание пользователя СУБД «postgres\_exporter»

Для соединения утилиты с СУБД создать пользователя СУБД «postgres\_exporter» SQL-командой:

```
CREATE ROLE postgres_exporter SUPERUSER NOCREATEDB NOCREATEROLE  
NOINHERIT LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'Password';
```



The screenshot shows a terminal window with the title bar 'root@admin1: /usr/jatoba-5/bin'. The user 'postgres' is at the prompt 'postgres@admin1:/usr/jatoba-5/bin\$'. They enter 'psql', which prompts for a password. After entering the password, the 'psql (15.5)' prompt appears. The user then enters the SQL command 'CREATE ROLE postgres\_exporter SUPERUSER NOCREATEDB NOCREATEROLE NOINHERIT LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'Password';'. The command is executed successfully, and the prompt returns to 'postgres=#'.

Рисунок 5.2 – Создание роли «postgres\_exporter»



В рассматриваемом примере пользователь СУБД «postgres\_exporter» является привилегированным пользователем

## 5.3. Настройка переменных окружения

Дальнейшая настройка утилиты требует внесения параметров подключения в файле переменных окружения «postgres\_exporter», командой:

```
gedit /usr/jatoba-5/monitoring/default/postgres_exporter
```

Необходимо настроить имя пользователя, пароль и параметры SSL-подключения в файле переменных окружения «postgres\_exporter».

Строка подключения выполнена в формате схемы URL. Основная форма URI подключения имеет синтаксис:

```
postgresql://[пользователь@][сервер] [/база_данных] [?:указание_па  
раметра]  
где пользователь:  
имя_пользователя[:пароль]  
и сервер:  
[узел] [:порт] [, ...]  
и указание_параметра:
```



имя=значение [&...]

В качестве обозначения схемы URI может использоваться postgresql:// или postgres://. Остальные части URI являются необязательными. В следующих примерах показан допустимый синтаксис URI:

```
postgresql://  
postgresql://localhost  
postgresql://localhost:5433  
postgresql://localhost/mydb  
postgresql://user@localhost  
postgresql://user:secret@localhost  
postgresql://other@localhost/otherdb?connect_timeout=10&application_name=myapp  
postgresql://host1:123,host2:456/somedb?target_session_attrs=any&application_name=myapp
```

В рассматриваемом примере на целевой СУБД:

– u602doc-pgr01 IP - 10.116.102.45 строка подключения утилиты к СУБД имеет следующий вид:

```
DATA_SOURCE_NAME="postgresql://postgres_exporter:Password@10.116.102.45:5432/postgres?sslmode=disable"
```



Обратите внимание, что необходимо прописывать общий, а не локальный адрес сетевого интерфейса

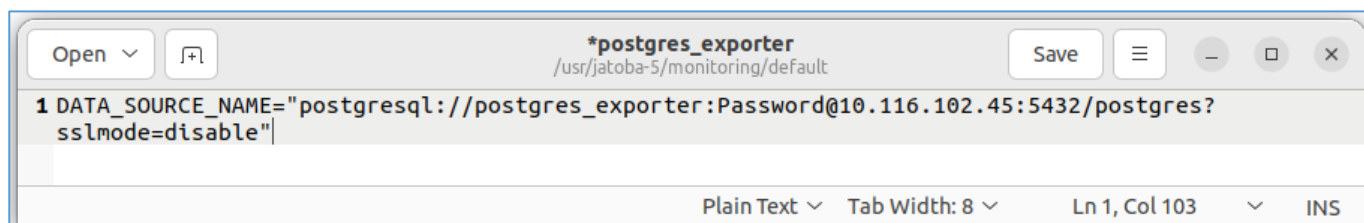


Рисунок 5.3 – Содержание файла «postgres\_exporter.default» на целевой СУБД u602doc-pgr01 IP - 10.116.102.45

– u602doc-ldar01 IP-10.116.102.47 строка подключения утилиты к СУБД имеет следующий вид:

```
DATA_SOURCE_NAME="postgresql://postgres_exporter:Password@10.116.102.47:5432/postgres?sslmode=disable"
```

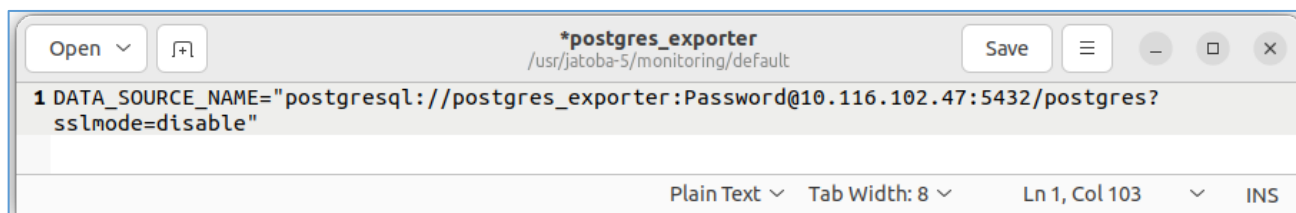


Рисунок 5.4– Содержание файла «postgres\_exporter.default» на целевой СУБД u602doc-ldap01 IP-10.116.102.47

#### 5.4. Запуск утилиты «postgres\_exporter»

Обновить конфигурацию system командой:

```
# sudo systemctl daemon-reload
```

Запустить службу экспортера, включить ее автозапуск и проверить статус работы:

```
# systemctl start jatoba<ver>_postgres_exporter
# systemctl enable jatoba<ver>_postgres_exporter
# systemctl status jatoba<ver>_postgres_exporter
```

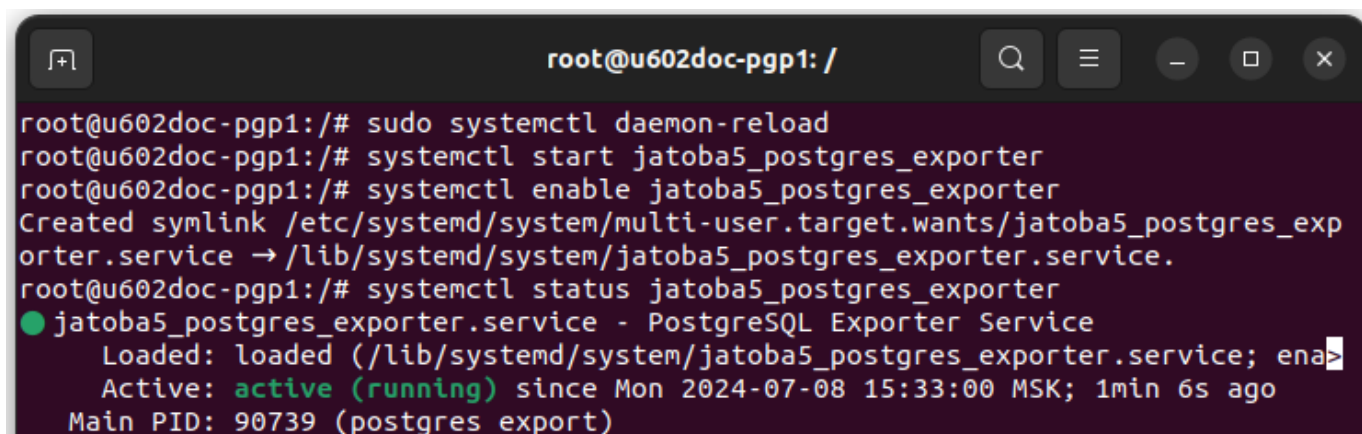


Рисунок 5.5 – Запуск и вывод статуса службы «postgres\_exporter»

Чтобы проверить статус работы экспортера нужно в браузере открыть веб-интерфейс экспортера:

```
# localhost:9187
# http://0.0.0.0:9187
```

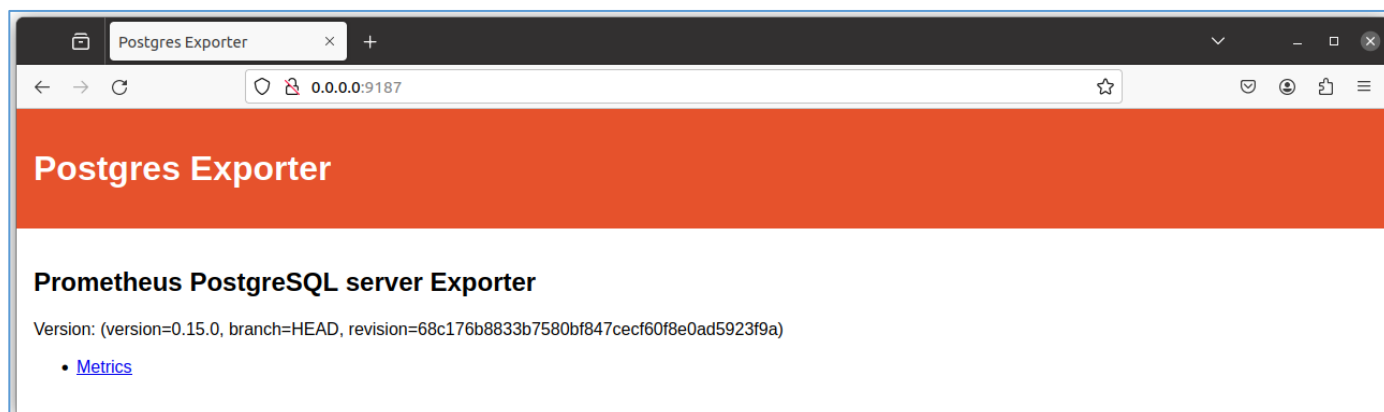


Рисунок 5.6 – Веб-интерфейс «postgres\_exporter»

В рассматриваемом примере на целевой СУБД:

– u602doc-pgp01 IP - 10.116.102.45 веб-интерфейс утилиты «node\_exporter» проверяется по URL:

```
# http://10.116.102.45:9187
```

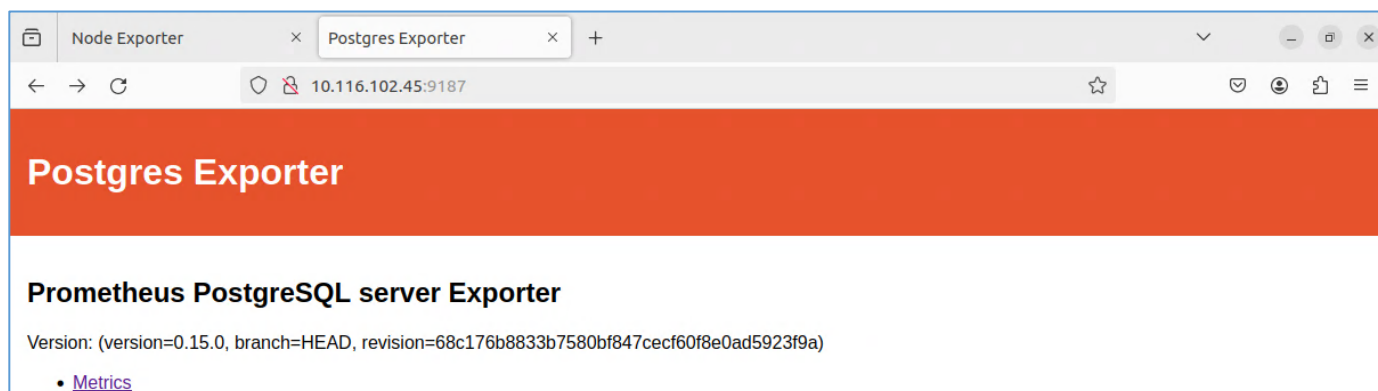


Рисунок 5.7 – Веб-интерфейс «postgres\_exporter» на целевой СУБД u602doc-pgp01 IP - 10.116.102.45

– u602doc-ldap01 IP-10.116.102.47 веб-интерфейс утилиты «node\_exporter» проверяется по URL:

```
# http://10.116.102.47:9187
```

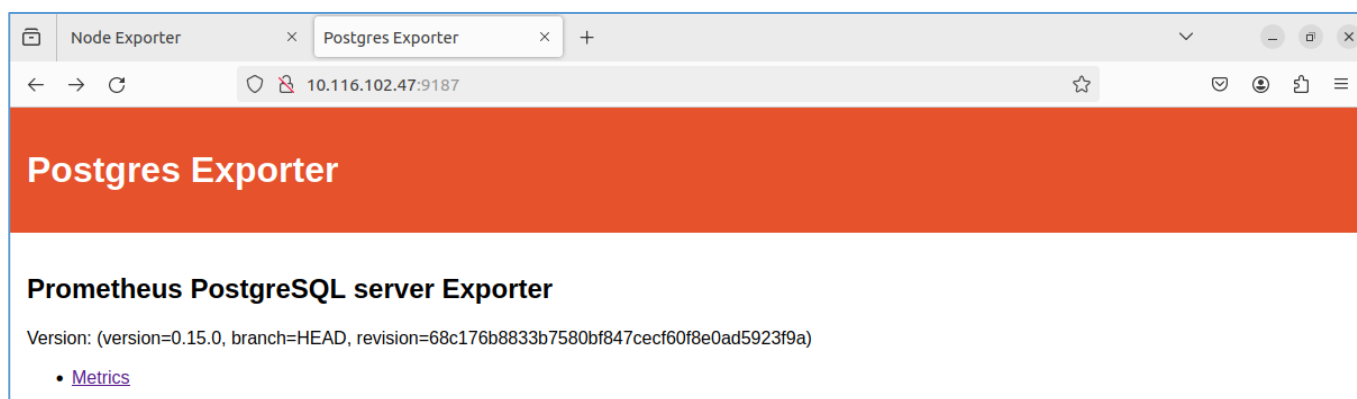


Рисунок 5.8 – Веб-интерфейс «postgres\_exporter» на целевой СУБД u602doc-ldap01 IP-10.116.102.47

При успешном подключении к БД на странице localhost:9187/metrics будет показан список значений метрик с префиксом «pg\_» в имени.

Если необходимо изменить значение адреса веб-интерфейса (по умолчанию :9187), postgres\_exporter запускается с опцией --web.listen-address, например:

```
export DATA_SOURCE_NAME=postgresql://postgres:secret@127.0.0.1  
./jatoba*_postgres_exporter --web.listen-address=:9188
```



Изменение адреса веб-интерфейса целесообразнее сохранить в файле сервиса «postgres\_exporter». Иначе при перезагрузке ОС настройки компонента вернуться к изначальным, хранящимся в файле сервиса.

Полный список опций командной строки postgres\_exporter можно вывести, если запустить его с опцией --help.

## 6. УСТАНОВКА ЭКСПОРТЕРА «JATOBA\*\_SQL\_EXPORTER»

Экспортер «jatoba\*\_SQL\_exporter» должен быть установлен на всех целевых СУБД и в той же БД в которой установлено расширение pg\_stat\_statements.

Данный экспортер можно использовать для расширения состава метрик, снимаемых с сервера PostgreSQL стандартным экспортером «jatoba\*\_postgres\_exporter» (см. п. 5).

Это агент, также написанный на языке Golang, который подключается к заданному источнику данных (БД) и забирает с него метрики по pull-запросу сервера «Prometheus».

Состав собираемых метрик и SQL-запросов, которые их возвращают, полностью конфигурируемы пользователем. Используемые SQL-запросы группируются в так называемые коллекторы, состав которых легко может быть расширен. Также в коллекторе для каждого возвращаемого запросом поля задается мэппинг на соответствующую метрику.

### 6.1. Установка утилиты и службы «sql\_exporter»

Установка пакета выполняется в соответствии с Руководством по установке, из локального репозитория командой:

```
apt-get install jatoba<ver>-sql-exporter
```

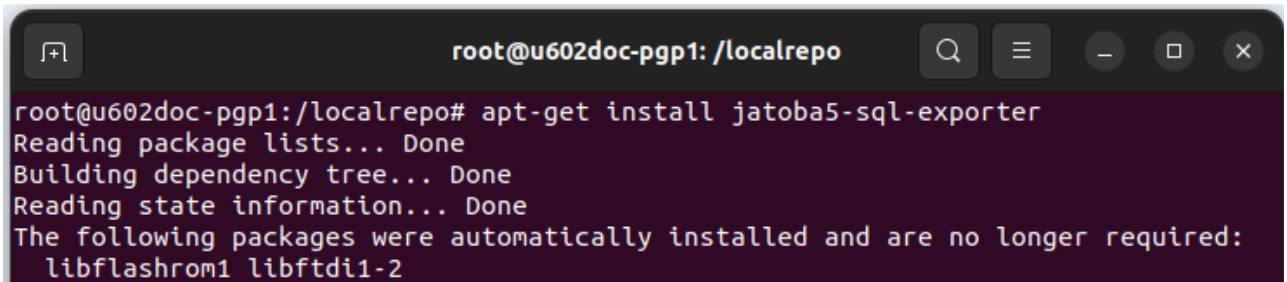


Рисунок 6.1 – Установка пакета «jatoba\*-sql-exporter»

В результате установки пакета будет создан:

- файл запуска по адресу:

```
/usr/jatoba-<ver>/bin/sql_exporter
```

- конфигурационный файл по адресу:

```
/usr/jatoba-<ver>/monitoring/default/sql_exporter.yml
```

- конфигурационный файл переменных окружения по адресу

```
/usr/jatoba-5/monitoring/default/sql_exporter
```

– пользователь ОС «sql\_exporter\_usr», от которого будет производиться запуск сервиса.

У данного пользователя нет интерактивной оболочки для входа.

## 6.2. Настройка переменных окружения

Проверить параметры экспортера в файле переменных окружения «sql\_exporter», выполнив команду редактирования:

```
gedit /usr/jatoba-5/monitoring/default/sql_exporter
```

Основным из параметров является путь к конфигурационному файлу «sql\_exporter.yml» в строке параметра CONF\_FILE.

```
CONF_FILE=/etc/sql_exporter/sql_exporter.yml
```

Настройка и расположение файла «sql\_exporter.yml» приведены в п. 6.4 настоящего документа.

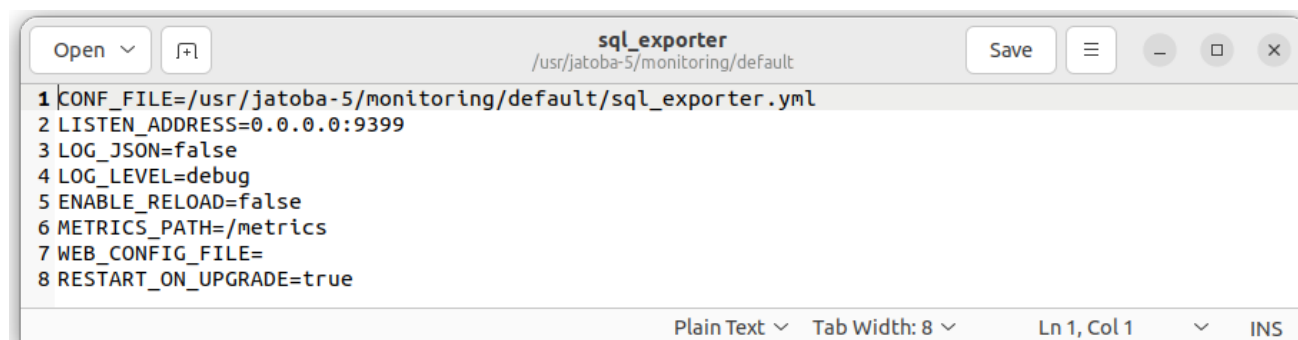
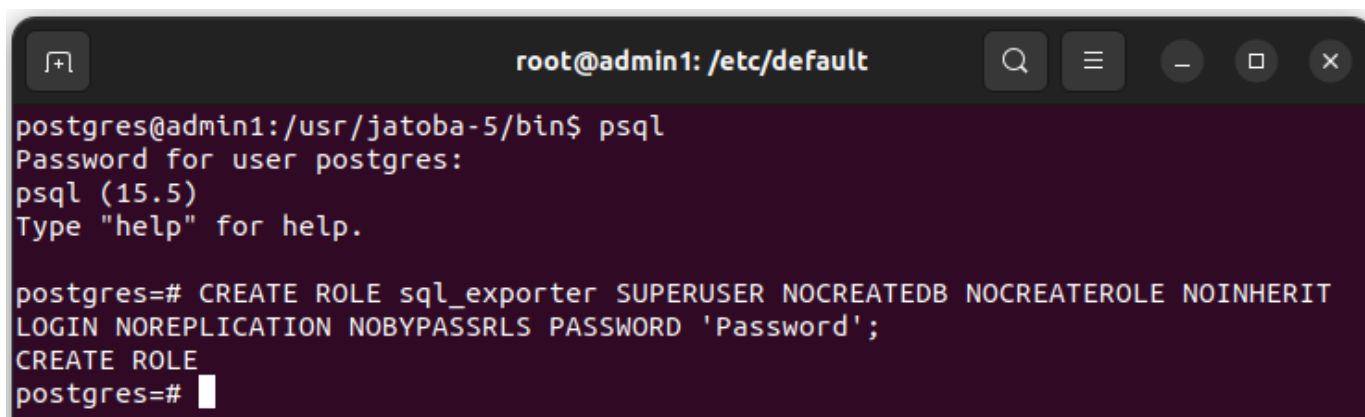


Рисунок 6.2 – Содержание файла переменных окружения «sql\_exporter»

## 6.3. Создание пользователя СУБД «sql\_exporter»

Для соединения утилиты с СУБД необходимо создать пользователя СУБД «sql\_exporter» SQL-командой:

```
CREATE ROLE sql_exporter SUPERUSER NOCREATEDB NOCREATEROLE  
NOINHERIT LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'Password';
```



```
root@admin1: /etc/default
postgres@admin1:/usr/jatoba-5/bin$ psql
Password for user postgres:
psql (15.5)
Type "help" for help.

postgres=# CREATE ROLE sql_exporter SUPERUSER NOCREATEDB NOCREATEROLE NOINHERIT
LOGIN NOREPLICATION NOBYPASSRLS PASSWORD 'Password';
CREATE ROLE
postgres=#
```

Рисунок 6.3 – Создание роли «sql\_exporter»



В рассматриваемом примере пользователь СУБД «sql\_exporter» является привилегированным пользователем

#### 6.4. Настройка параметров экспортера и подключения к БД в файле «sql\_exporter.yml»

Основным параметром для настройки параметров экспортера и подключения к БД в файле «sql\_exporter.yml» является параметр «data\_source\_name».

Требуется открыть файл для редактирования командами:

```
gedit /usr/jatoba-5/monitoring/default/sql_exporter.yml
```

Строка подключения выполнена в формате схемы URL. Синтаксис строки описан в п. 5.3 настоящего документа.

В рассматриваемом примере на целевой СУБД:

– u602doc-pgr01 IP - 10.116.102.45 строка подключения утилиты к СУБД имеет следующий вид:

```
data_source_name:
'postgresql://sql_exporter:Password@10.116.102.45:5432/postgres
?sslmode=disable'
```



Обратите внимание, что необходимо прописывать общий, а не локальный адрес сетевого интерфейса

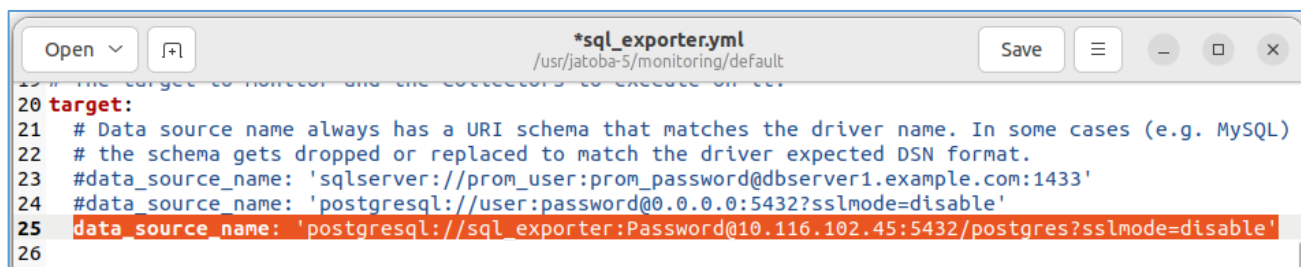


Рисунок 6.4 – Содержание файла «sql\_exporter.yml», строка «data\_source\_name» на целевой СУБД u602doc-pgp01 IP - 10.116.102.45

– u602doc-ldap01 IP-10.116.102.47 строка подключения утилиты к СУБД имеет следующий вид:

```
data_source_name:
'postgresql://sql_exporter:Password@10.116.102.47:5432/postgres
?sslmode=disable'
```

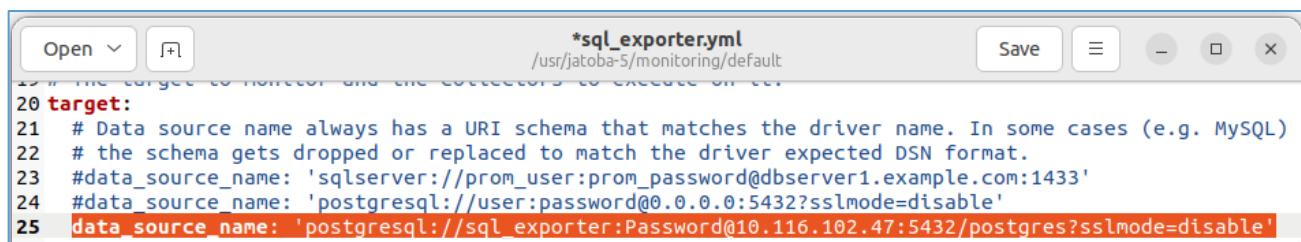


Рисунок 6.5 – Содержание файла «sql\_exporter.yml», строка «data\_source\_name» на целевой СУБД u602doc-ldap01 IP-10.116.102.47

Сохранить внесенные изменения.

В дистрибутиве содержится файл с подготовленными метриками для мониторинга СУБД «Jatoba» «postgres.collector.yml», который по умолчанию использует «jatoba\*\_SQL\_exporter».

### 6.5. Запуск утилиты «jatoba\*\_sql\_exporter»

Обновить конфигурацию systemd:

```
# systemctl daemon-reload
```

Запустить службу экспортера, включить ее в автозапуск и проверить статус работы:

```
# systemctl start jatoba5_sql_exporter
# systemctl enable jatoba5_sql_exporter
# systemctl status jatoba5_sql_exporter
```



```
root@u602doc-pgp1: /
root@u602doc-pgp1:/# systemctl daemon-reload
root@u602doc-pgp1:/# systemctl start jatoba5_sql_exporter
root@u602doc-pgp1:/# systemctl enable jatoba5_sql_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/jatoba5_sql_exporter.service → /lib/systemd/system/jatoba5_sql_exporter.service.
root@u602doc-pgp1:/# systemctl status jatoba5_sql_exporter
● jatoba5_sql_exporter.service - SQL Exporter for Prometheus
   Loaded: loaded (/lib/systemd/system/jatoba5_sql_exporter.service; enabled;
   Active: active (running) since Tue 2024-07-09 08:38:59 MSK; 17s ago
```

Рисунок 6.6 – Установка и запуск службы «sql\_exporter»

Чтобы проверить статус работы экспортера, нужно в браузере открыть веб-интерфейс экспортера:

```
# localhost:9399
# http://0.0.0.0:9399
```

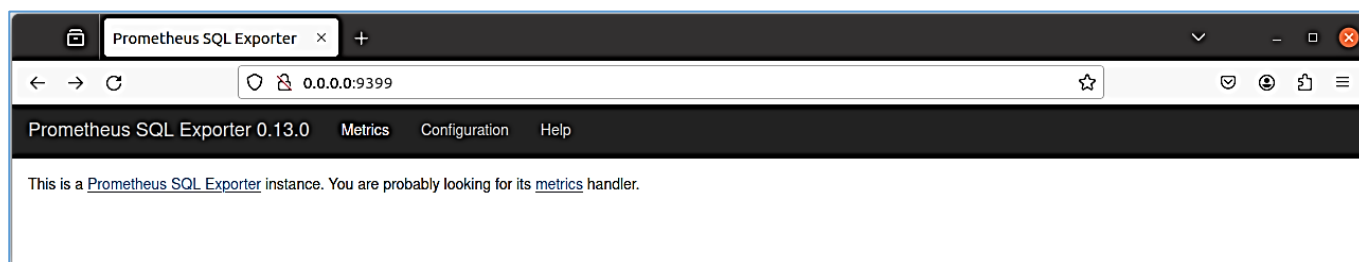


Рисунок 6.7 – Веб-интерфейс «sql\_exporter»

В рассматриваемом примере на целевой СУБД:

– u602doc-pgp01 IP - 10.116.102.45 веб-интерфейс утилиты «sql\_exporter» проверяется по URL:

```
# http://10.116.102.45:9399
```

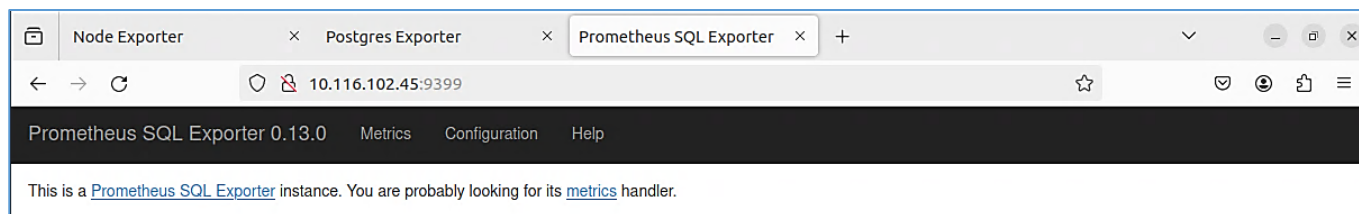


Рисунок 6.8 – Веб-интерфейс «sql\_exporter» на целевой СУБД u602doc-pgp01 IP - 10.116.102.45

– u602doc-ldap01 IP-10.116.102.47 веб-интерфейс утилиты «node\_exporter» проверяется по URL:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
# http://10.116.102.47:9399
```

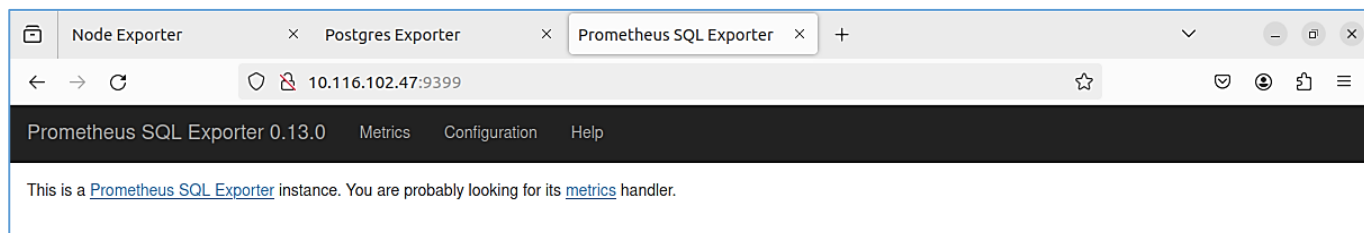


Рисунок 6.9 – Веб-интерфейс «sql\_exporter» на целевой СУБД u602doc-ldap01 IP-10.116.102.47

При успешном подключении к БД и отсутствии ошибок в конфигурации на странице localhost:9399/metrics будет показан список значений снятых метрик.

Если необходимо изменить значения адреса веб-интерфейса (:9399), jatoba\*\_sql\_exporter запускается с опцией -web.listen-address, например:

```
./jatoba5_sql_exporter -web.listen-address :9398
```



Изменение адреса веб-интерфейса целесообразнее сохранить в файле сервиса «sql\_exporter». Иначе при перезагрузке ОС настройки компонента вернуться к изначальным, хранящимся в файле сервиса.

Полный список опций командной строки sql\_exporter можно вывести, если запустить его с опцией -help.

## 7. СИСТЕМА «PROMETHEUS»

«Prometheus» – система мониторинга различных программных систем и сервисов. «Prometheus» собирает и сохраняет метрики в виде временных рядов данных. Информация о каждой метрике хранится вместе с отметкой времени, когда она была записана, и опциональным набором меток (labels), представляющих пары «ключ: значение». Сами метрики являются числовыми измерениями, которые по типу могут быть монотонно возрастающими значениями счетчиков (counter) или произвольно изменяющимися значениями датчиков (gauge).

Основными компонентами системы «Prometheus» являются:

- Сервер «Prometheus», который собирает и сохраняет метрики в своей встроенной базе данных TSDB;
- Экспортеры данных, которые по запросу сервера снимают метрики с заданных сервисов (targets) и возвращают их серверу;
- Web UI, с помощью которого можно исследовать собранные метрики с помощью языка запросов promQL.

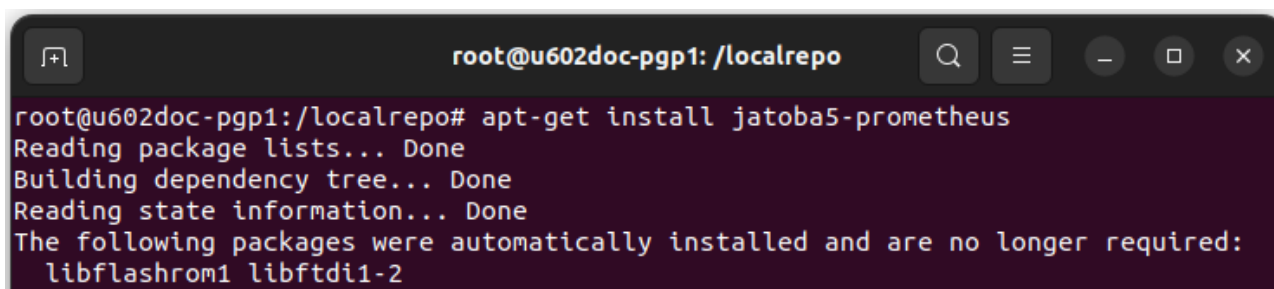
Документация по использованию системы находится на официальном сайте разработчика по адресу: <https://prometheus.io>.

### 7.1. Установка системы «Prometheus»

Документация по использованию системы находится на официальном сайте разработчика по адресу: <https://prometheus.io>.

Установка пакета выполняется в соответствии с Руководством по установке, из локального репозитория командой:

```
apt-get install jatoba<ver>-prometheus
```



```
root@u602doc-pgp1: /localrepo
root@u602doc-pgp1:/localrepo# apt-get install jatoba5-prometheus
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
```

Рисунок 7.1 – Установка пакета «jatoba<ver>-prometheus»

В результате установки пакета будет создан:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

- файл переменных окружения сервиса по адресу:

```
usr/jatoba-5/monitoring/default/prometheus
```

- файл сервиса по адресу

```
usr/lib/systemd/system/jatoba5_prometheus.service
```

- файл конфигурации, адаптированный под использование с СУБД «Jatoba» по адресу:

```
usr/jatoba-5/monitoring/default/prometheus.yml
```

- база данных по адресу:

```
/opt/prometheus
```

- служебные директории веб-консоли по адресу:

```
usr/jatoba-5/monitoring/prometheus
```

- пользователь ОС «prometheus», от которого будет производиться запуск сервиса.

У данного пользователя нет интерактивной оболочки для входа и нет домашней директории.

## 7.2. Конфигурация системы «Prometheus»

Необходимо задать конфигурацию сервера в формате YAML выполнив команду редактирования:

```
gedit /usr/jatoba-5/monitoring/default/prometheus.yml
```

В конфигурации важными параметрами являются:

- частота опроса метрик (scrape\_interval);
- время ожидания ответа (scrape\_timeout);
- HTTP, IP адреса (targets).

Для параметра «targets» возможно указать одну или несколько целей, для получения метрик с экспортера, при этом параметр будет иметь синтаксис, с одной целью:

```
- targets: ['X.X.X.X:port']
```

и с несколькими целями

```
- targets: ['X.X.X.X:port', 'X.X.X.X:port']
```

В рассматриваемом примере, в конфигурационном файле `prometheus.yml` устанавливаются IP-адреса серверов, находящихся под наблюдением.



Обратите внимание, что необходимо прописывать общий, а не локальный адрес сетевого интерфейса

Ниже приведены примеры таких «job-name».

### 7.2.1. Примеры блока «postgres-exporter»

Пример стандартного «postgres-exporter» экспортера с двумя целями:

```
# стандартный экспортер данных для PostgreSQL
- job_name: "postgres-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.102.45:9187', '10.116.102.47:9187']
      labels:
        alias: postgres
```

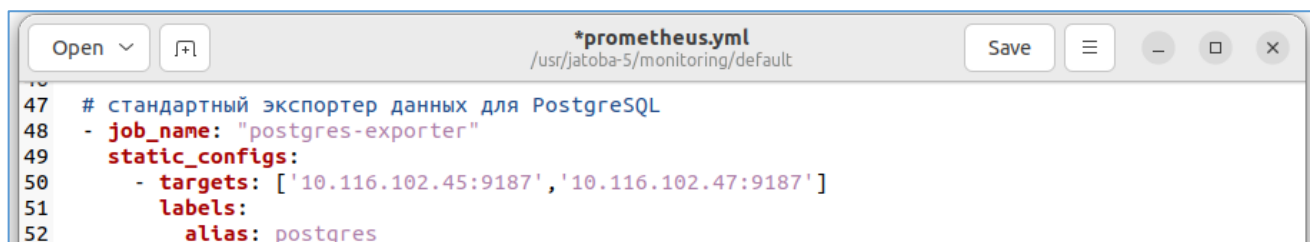


Рисунок 7.2 - Стандартный экспортер данных для PostgreSQL

Пример стандартного «postgres-exporter» экспортера для двух «job-name» с одной и двумя целями:

```
# Экспортер данных для PostgreSQL сервера srv1
- job_name: "srv1-postgres-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.103.45:9187']
      labels:
```

```
alias: postgres

# Экспортер данных для PostgreSQL сервера srv2 и srv3
- job_name: "srv2-srv3-postgres-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.103.46:9187', '10.116.103.47:9187']
      labels:
        alias: postgres
```



Рисунок 7.3 - «postgres-exporter» экспортера для двух «job-name» с одной и двумя целями

### 7.2.2. Примеры блока «sql-exporter»

Пример экспортера данных для SQL с двумя целями:

```
# экспортер данных для SQL
- job_name: "sql-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.102.45:9399', '10.116.102.47:9399']
      labels:
        alias: postgres
```

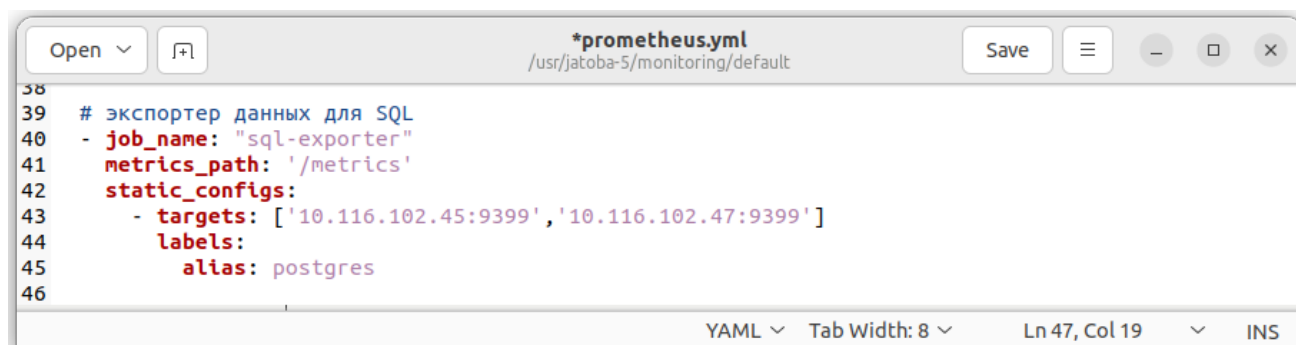


Рисунок 7.4 - «sql-exporter» с двумя целями

Пример экспортера SQL для двух «job-name» с одной и двумя целями:

```
# Экспортер данных для SQL сервера srv1
- job_name: "srv1-sql-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.103.45:9399']
      labels:
        alias: postgres

# Экспортер данных для SQL сервера srv2 и srv3
- job_name: "srv2-srv3-sql-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.103.46:9399', '10.116.103.47:9399']
      labels:
        alias: postgres
```



Рисунок 7.5 – «sql-exporter» для двух «job-name» с одной и двумя целями

### 7.2.3. Примеры блока «node-exporter»

Пример экспортера данных "node-exporter" для GNU/ Linux с двумя целями:

```
# экспортер данных для Linux
- job_name: "node-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.102.45:9100', '10.116.102.47:9100']
      labels:
        alias: os
```

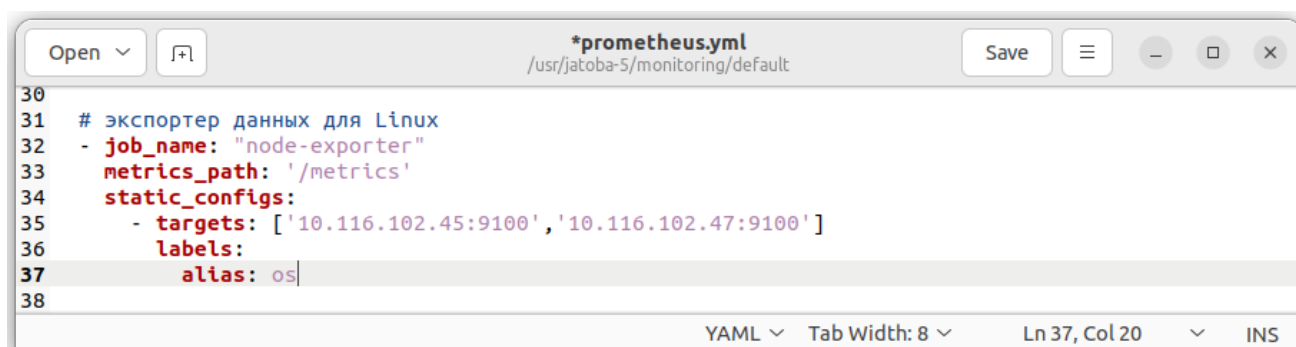


Рисунок 7.6 – «node-exporter» для GNU/ Linux с двумя целями

Пример экспортера данных «node-exporter» для GNU/Linux для двух «job-name» с одной и двумя целями:

```
# Экспортер данных для Linux сервера srv1
- job_name: "srv1-node-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.103.46:9100']
      labels:
        alias: os

# Экспортер данных для Linux сервера srv2 и srv3
- job_name: "srv2-srv3-node-exporter"
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.116.103.46:9100', '10.116.103.47:9100']
      labels:
        alias: os
```

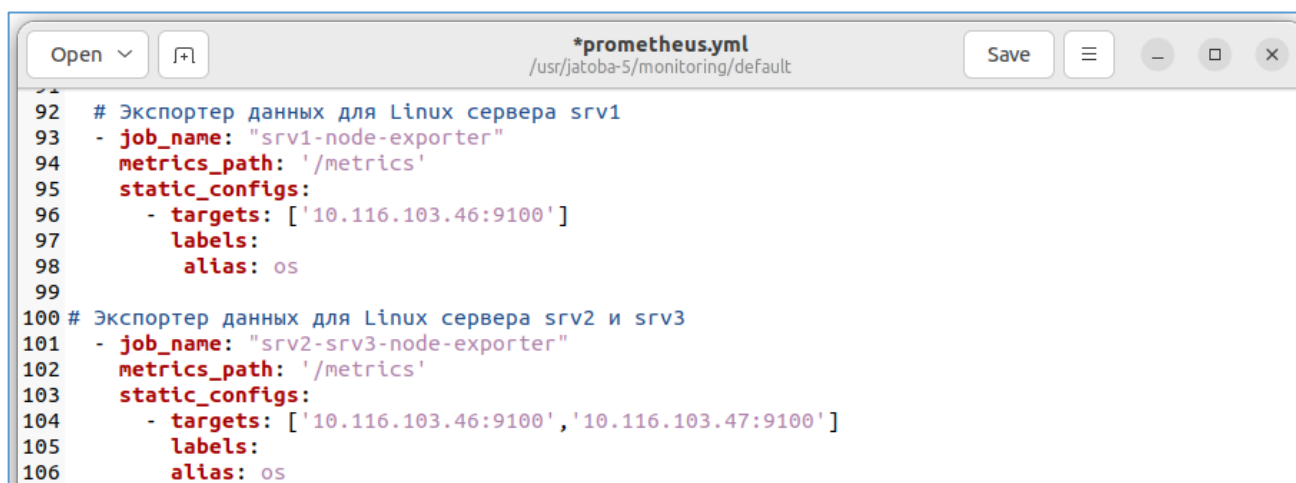


Рисунок 7.7 – «node-exporter» для GNU/Linux для двух «job-name» с одной и двумя целями



В конфигурационном файле prometheus.yml, в строке «targets», допустимо указывать любое количество адресов экспортеров, относящихся к одной или разным СУБД.

Обработка критических событий и вычисление rules не заданы, хотя соответствующие блоки присутствуют в конфигурации.

Полное описание параметров конфигурации сервера приведено в документации <https://prometheus.io/docs/prometheus/latest/configuration/configuration/>

### 7.3. Запуск системы «Prometheus»

Перед запуском сервиса требуется удостовериться в корректности содержания файла сервиса.

Просмотр файла осуществляется командой в терминале ОС:

```
gedit usr/lib/systemd/system/jatoba5_prometheus.service
```

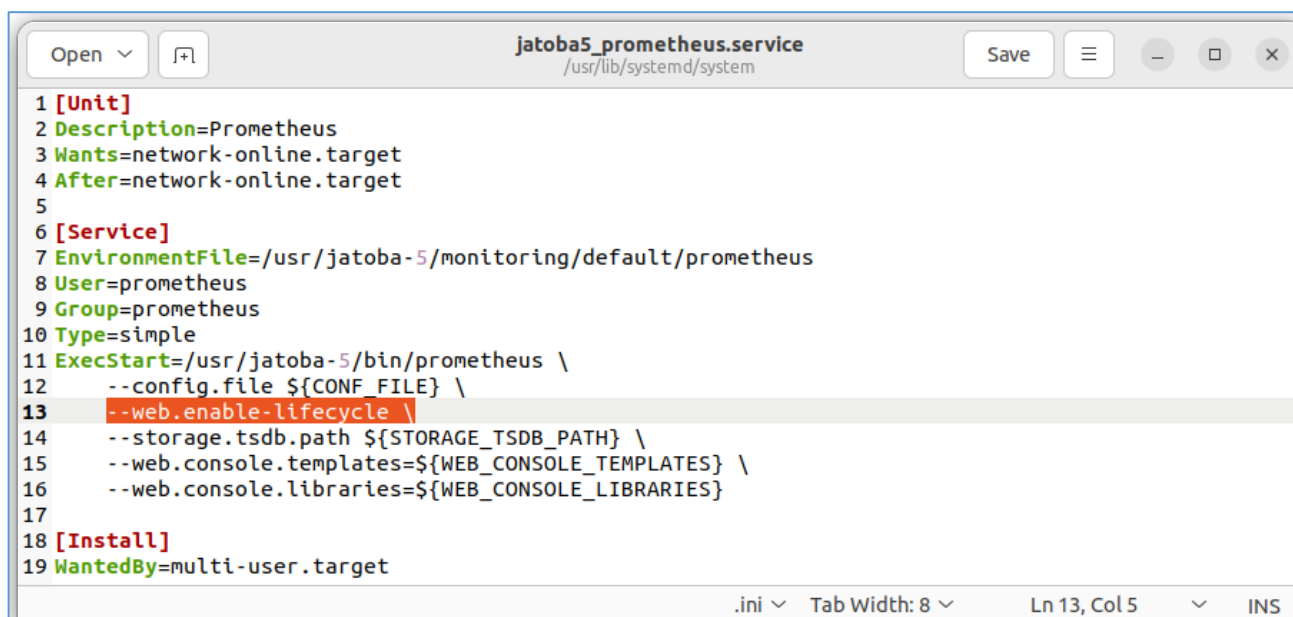


Рисунок 7.8 – Содержание файла сервиса «jatoba-5\_prometheus.service»

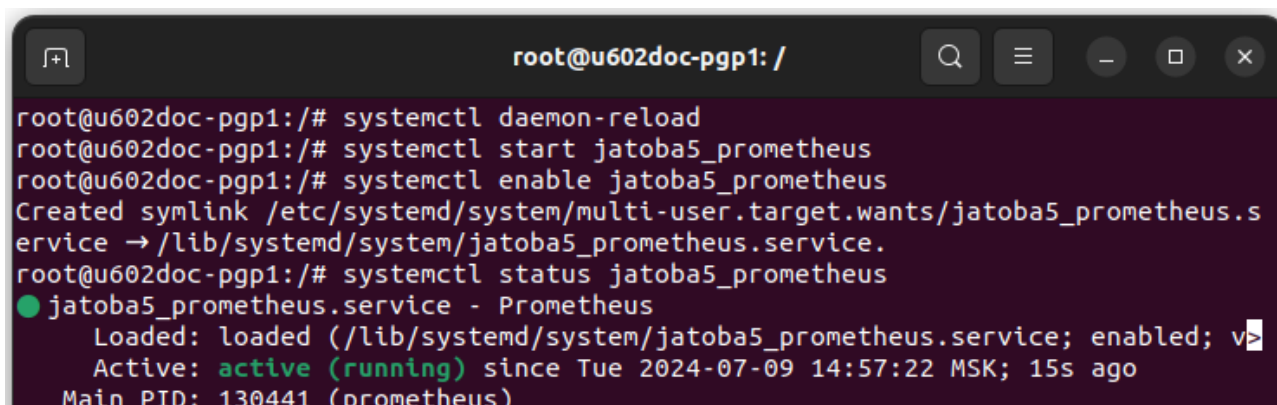
Обновить конфигурацию systemd:

```
# systemctl daemon-reload
```

Запустить службу система, включить ее в автозапуск и проверить статус работы:

```
# systemctl start jatoba5_prometheus
# systemctl enable jatoba5_prometheus
```

```
# systemctl status jatoba5_prometheus
```



```
root@u602doc-pgp1: /
root@u602doc-pgp1:/# systemctl daemon-reload
root@u602doc-pgp1:/# systemctl start jatoba5_prometheus
root@u602doc-pgp1:/# systemctl enable jatoba5_prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/jatoba5_prometheus.s
ervice → /lib/systemd/system/jatoba5_prometheus.service.
root@u602doc-pgp1:/# systemctl status jatoba5_prometheus
● jatoba5_prometheus.service - Prometheus
   Loaded: loaded (/lib/systemd/system/jatoba5_prometheus.service; enabled; v>
   Active: active (running) since Tue 2024-07-09 14:57:22 MSK; 15s ago
   Main PID: 130441 (prometheus)
```

Рисунок 7.9 Установка и запуск службы системы «Prometheus»

Статус запущенного сервера «Prometheus» можно проверить с помощью web UI, открыв в браузере страницу с адресом <http://localhost:9090>.

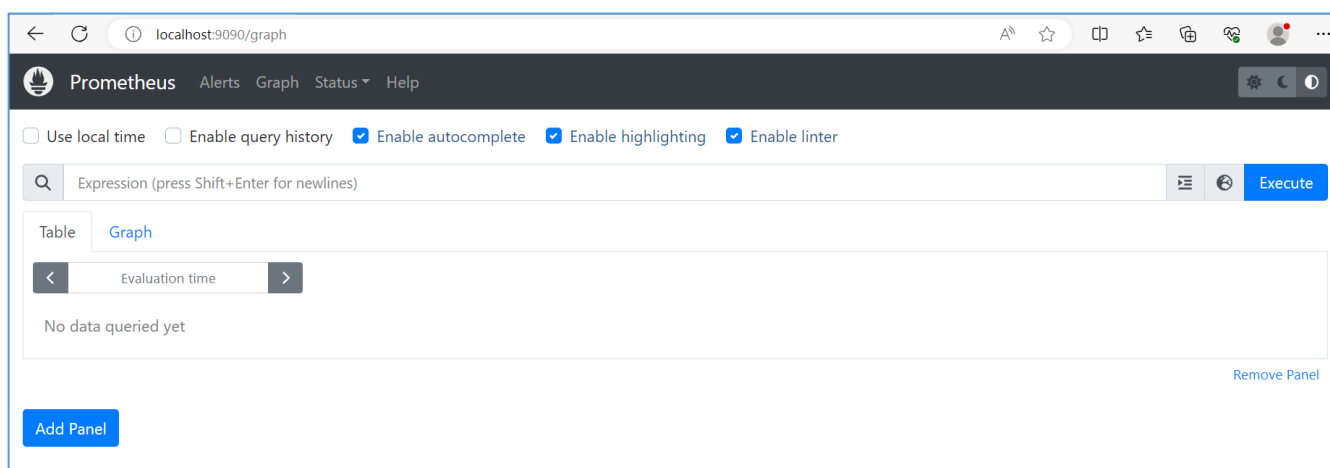


Рисунок 7.10 – Веб интерфейс системы «Prometheus»

На вкладке «Status» можно посмотреть текущую конфигурацию и опции запуска сервера, статус встроенной базы данных tsdb и заданные цели (targets).

В окне «Expression» можно ввести название метрики или выражение на языке promQL и, нажатием на кнопку «Execute», отобразить результаты в виде таблицы или графика.

Руководство по языку запросов promQL располагается по адресу: <https://prometheus.io/docs/prometheus/latest/querying/basics/>

Полный список собираемых метрик можно открыть при нажатии на кнопку «Open Metrics Explorer» слева от кнопки «Execute» или отобразить в окне «Expressions» при вводе первых символов наименования метрики, если включена опция автодополнения.

Список значений собираемых метрик для каждой цели можно отобразить на странице веб-интерфейса соответствующего экспортера данных, например, localhost:9090/metrics.

The screenshot shows the Prometheus web interface at 0.0.0.0:9090/targets. It displays a list of targets under the 'Targets' section. The interface includes a search bar, status filters (All, Unhealthy, Collapse All), and a legend for target states (Unknown, Unhealthy, Healthy). Three targets are listed, each with a table of details.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<b>node-exporter (1/1 up)</b> <a href="#">show less</a>					
http://localhost:9100/metrics	UP	alias="os" instance="localhost:9100" job="node-exporter"	26.282s ago	12.156ms	
<b>postgres-exporter (1/1 up)</b> <a href="#">show less</a>					
http://localhost:9187/metrics	UP	alias="postgres" instance="localhost:9187" job="postgres-exporter"	6.890s ago	1.9s	
<b>sql-exporter (1/1 up)</b> <a href="#">show less</a>					
http://localhost:9399/metrics	UP	alias="postgres" instance="localhost:9399" job="sql-exporter"	1.496s ago	7.645ms	

Рисунок 7.11 – Список значений собираемых метрик

В рассматриваемом примере подключение к системе «Prometheus» используется адрес:

http://10.116.102.41:9090/

Перейдя в меню «Target» отразятся статистические данные наблюдаемых СУБД.

Prometheus Time Series
+
10.116.102.41:9090/targets?search=
Prometheus
Alerts
Graph
Status
Help

## Targets

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

### node-exporter (2/2 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://10.116.102.45:9100/metrics">http://10.116.102.45:9100/metrics</a>	UP	alias="os" instance="10.116.102.45:9100" job="node-exporter"	5.404s ago	11.536ms	
<a href="http://10.116.102.47:9100/metrics">http://10.116.102.47:9100/metrics</a>	UP	alias="os" instance="10.116.102.47:9100" job="node-exporter"	18.15s ago	10.563ms	

### postgres-exporter (2/2 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://10.116.102.45:9187/metrics">http://10.116.102.45:9187/metrics</a>	UP	alias="postgres" instance="10.116.102.45:9187" job="postgres-exporter"	18.958s ago	1.2s	
<a href="http://10.116.102.47:9187/metrics">http://10.116.102.47:9187/metrics</a>	UP	alias="postgres" instance="10.116.102.47:9187" job="postgres-exporter"	6.189s ago	30.167ms	

Рисунок 7.12 – Страница целей системы «Prometheus»

## 8. УТИЛИТА «ALERTMANAGER»

Alertmanager — это инструмент для управления и обработки оповещений в системе мониторинга Prometheus. Он выполняет следующие функции:

- группировка оповещений: группирует похожие оповещения для снижения шума и предотвращения дублирования.
- удаление дубликатов: гарантирует отправку уникальных оповещений без повторений.
- маршрутизация и приглушение оповещений: позволяет определять правила и конфигурации для маршрутизации оповещений нужным получателям на основе их важности или других критериев. Также можно временно приглушить оповещения во время обслуживания или определённых периодов.
- уведомление о тревоге: интегрируется с различными каналами связи, такими как электронная почта, Slack, PagerDuty и другие, позволяя отправлять уведомления о тревогах нужным людям или командам.

### 8.1. Установка утилиты и службы «alertmanager»

Установка пакета выполняется в соответствии с Руководством по установке, из локального репозитория командой:

```
apt-get install jatoba<ver>-alertmanager
```

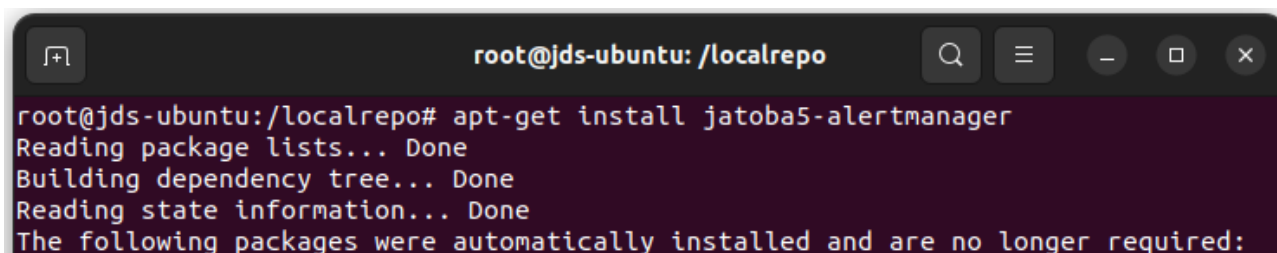


Рисунок 8.1 – Установка пакета «jatoba\*-alertmanager»

В результате установки пакета будет создан:

- файл запуска по адресу:

```
/usr/jatoba-<ver>/bin/alertmanager
```

- конфигурационный файл по адресу:

```
/usr/jatoba-<ver>/monitoring/default/alertmanager.yml
```

- служба по адресу:

```
/usr/lib/systemd/jatoba<ver>_alertmanager.service
```

- пользователь ОС «alertmanager», от которого будет производиться запуск сервиса.

У данного пользователя нет интерактивной оболочки для входа и нет домашней директории.

## 8.2. Настройка параметров утилиты файле «alertmanager.yml»

После установки пакета в конфигурационном файле будут установлены параметры по умолчанию.

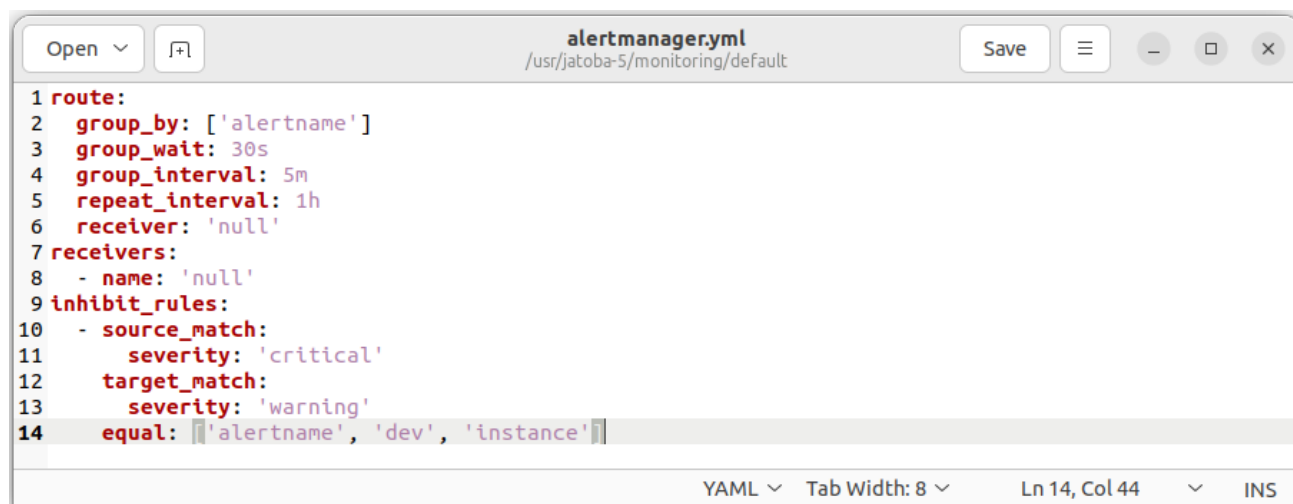


Рисунок 8.2 – Параметры по умолчанию

Редактирование конфигурационного файла выполняется командой:

```
gedit /usr/jatoba-<ver>/monitoring/default/alertmanager.yml
```

В узле «global» необходимо указать данные для подключения к почтовому серверу. Целесообразно использовать специальную, неперсонофицированную, техническую учетную запись почты, от имени которой будет рассылаться предупреждения.

В узле route указываются настройки агрегирования предупреждений.

В узле «receivers» в узле «email\_general» указываются настройки получателя. Используется общий узел для всех получателей, email получателя подставляется из метки «emailto» с помощью шаблона.

```
global:
  smtp_smarthost: mail.domain.ru:587
  smtp_from: domain_name@domain.ru
  smtp_auth_user_name: user_name@domain.ru
  smtp_auth_password: password
  smtp_require_tls: true
route:
  receiver: email_general
  group_by: [emailto]
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 3h
receivers:
  - name: email_general
    email_configs:
      - send_resolved: true
        to: '{{ .CommonLabels.emailto }}'
```

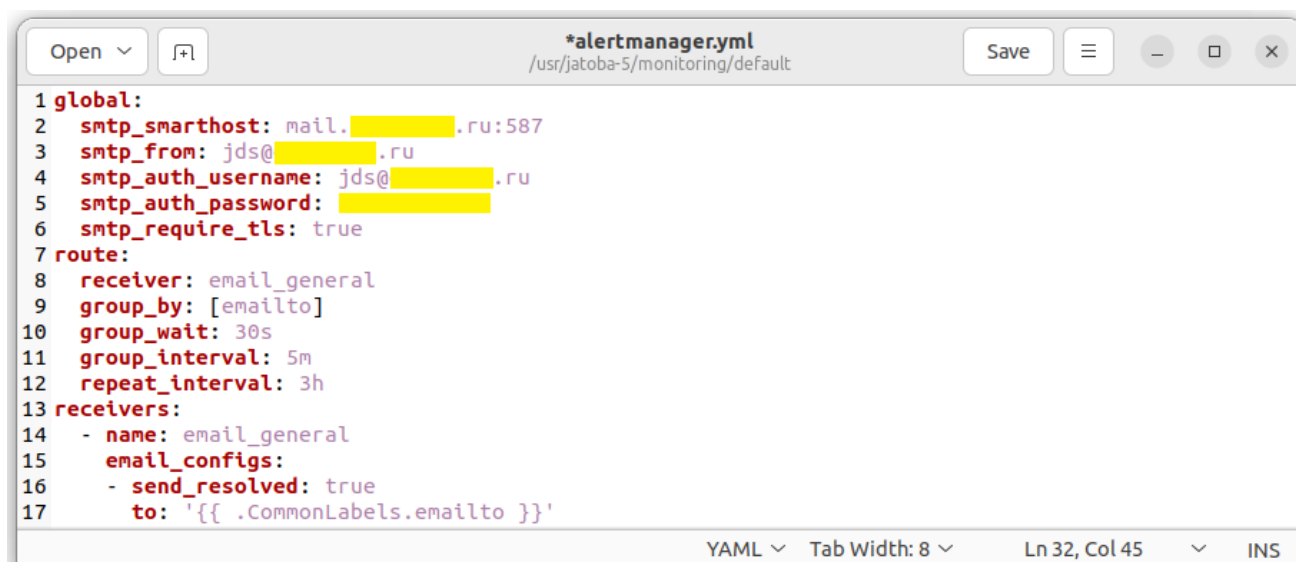


Рисунок 8.3 - Конфигурационного файла «alertmanager.yml»

### 8.3. Запуск утилиты «alertmanager»

Обновить конфигурацию system командой:

```
# sudo systemctl daemon-reload
```

Запустить службу утилиты, включить ее автозапуск и проверить статус работы:

```
# systemctl start jatoba<ver>_alertmanager
# systemctl enable jatoba<ver>_alertmanager
# systemctl status jatoba<ver>_alertmanager
```

```
root@jds-ubuntu: /
root@jds-ubuntu:/# sudo systemctl daemon-reload
root@jds-ubuntu:/# systemctl start jatoba5_alertmanager
root@jds-ubuntu:/# systemctl enable jatoba5_alertmanager
Created symlink /etc/systemd/system/multi-user.target.wants/jatoba5_alertmanager.service → /lib/systemd/system/jatoba5_alertmanager.service.
root@jds-ubuntu:/# systemctl status jatoba5_alertmanager
● jatoba5_alertmanager.service - Alertmanager
   Loaded: loaded (/lib/systemd/system/jatoba5_alertmanager.service; enabled;
   Active: active (running) since Wed 2024-07-10 12:33:43 UTC; 27s ago
```

Рисунок 8.4 - Запуск и вывод статуса службы «jatoba\*\_alertmanager»

Чтобы проверить статус работы утилиты, нужно в браузере открыть веб-интерфейс утилиты «Alertmanager»:

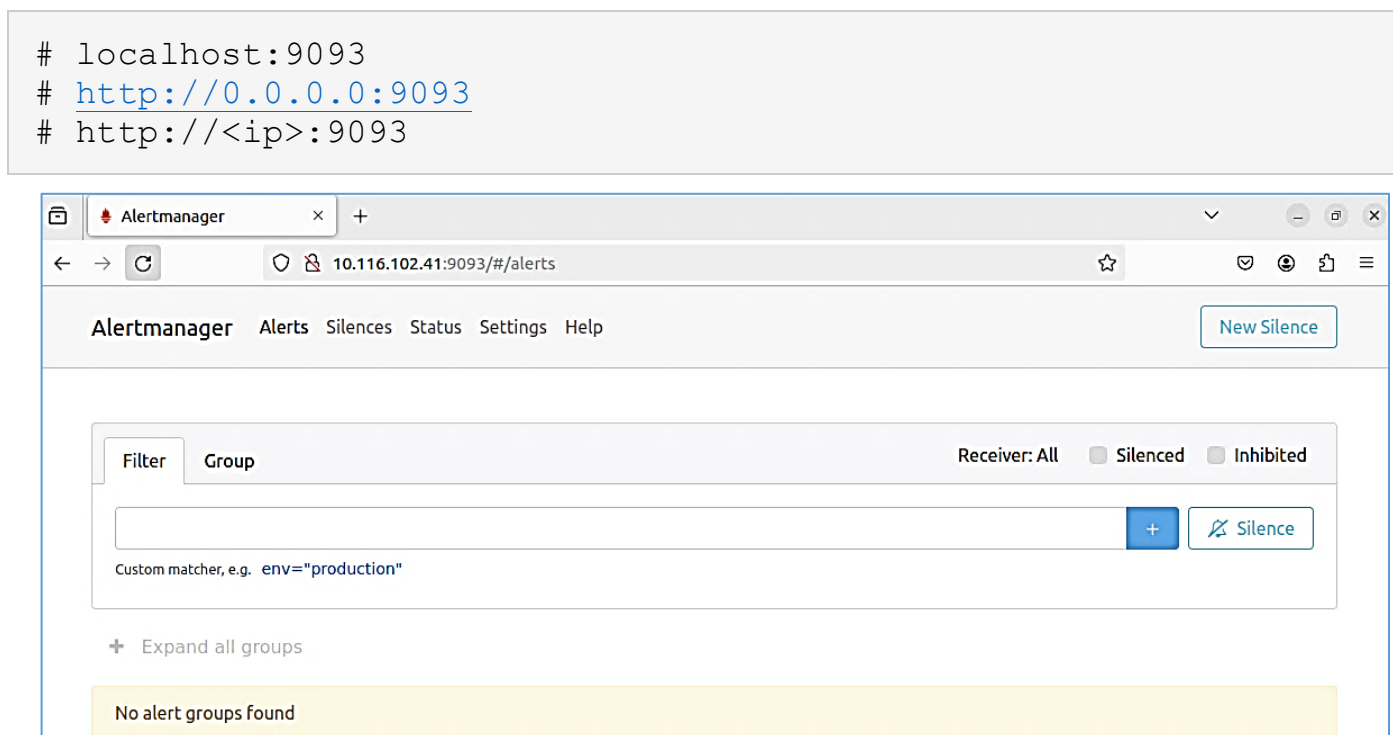


Рисунок 8.5 – Веб-интерфейс утилиты «alertmanager»

На данном шаге конфигурирование утилиты не закончено. Проверена, только работоспособность. Интеграция с другими компонентами описана в разделе 9 «Подключение к JDS».



## 9. ПОДКЛЮЧЕНИЕ К JDS

Подключение хранилища системы «Prometheus» к компоненту «Jatoba data safe» для отображения в разделе «Мониторинг» описано в документе «Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe», в пункте «Источники данных».

Для настройки «Уведомлений» о контролируемых значениях СУБД требуется сконфигурировать 3 компонента, такие как система «Prometheus», утилита «Alertmanager» и компоненты пользовательского веб-интерфейса для администраторов «Jatoba data safe» (JDS).

Последовательность действий будет следующая.

Настраивается SSH-соединение на хосте и/или с хостом с установленной системой «Prometheus» (см. п. 9.1).

В разделе «Настройки» компонента JDS, во вкладке «Источник данных» созданное подключение к системе «Prometheus» дополняется параметрами «Настройки конфигурации предупреждений» (уведомлений).

В этой настройке указывается, IP адрес системы «Prometheus», порт подключения, пользователь и путь к файлу с правилами уведомлений. В последствии это имя файла будет использовано в конфигурировании системы «Prometheus».

Файл с правилами уведомлений предварительно не создаётся и появляется по вышеуказанному пути. Поэтому для его создания требуется создать уведомление в разделе «Мониторинг» в любом из дашбордов.

На хосте с системой «Prometheus» в конфигурационном файле

```
usr/jatoba-5/monitoring/default/prometheus.yml
```

связать систему «Prometheus» и утилиту «Alertmanager».

### 9.1. Настройка SSH-соединения

Настройка SSH-соединения производится в обязательном порядке для любой архитектуры компонентов. В том числе, если утилита «Alertmanager», система «Prometheus» и JDS установлены на одном хосте.

Необходимо настроить SSH-соединение с хоста компонента JDS на сервер с развернутой системой «Prometheus». Соединение будет использоваться компонентом JDS для копирования конфигурационного файла с правилами предупреждений.

В настройках SSH-сервера должны быть разрешены локальные подключения и подключения от имени и с правами пользователя «root».

Следует выполнить следующие действия:

- создать папку пользователя, под которым работает JDS:

```
sudo -s  
mkdir /home/jds  
chown jds /home/jds  
exit
```

- сгенерировать ключи под пользователем JDS, скопировать на хост с системой «Prometheus»:

```
sudo -u jds /usr/bin/bash  
ssh-keygen  
(задать пустой пароль)  
ssh-copy-id root@IP  
(yes)
```

- проверить соединение (должно соединиться без запроса пароля):

```
ssh root@IP  
exit  
exit
```

## 9.2. Конфигурирование JDS

### Вкладка «Источник данных»

На хосте с установленным компонентом JDS перейти в раздел «Настройки». Созданной подключение к системе «Prometheus» изменить, дополнив параметрами «Настройки конфигурации предупреждений» (уведомлений).

В этой настройке указывается:

- IP адрес системы «Prometheus»;
- порт подключения – 22, соответствующий SSH-подключению;
- пользователь – root;
- путь к файлу с правилами уведомлений:

```
/usr/jatoba-5/monitoring/default/alertrules.yml
```

В последствии это имя файла будет использовано в конфигурировании системы «Prometheus».

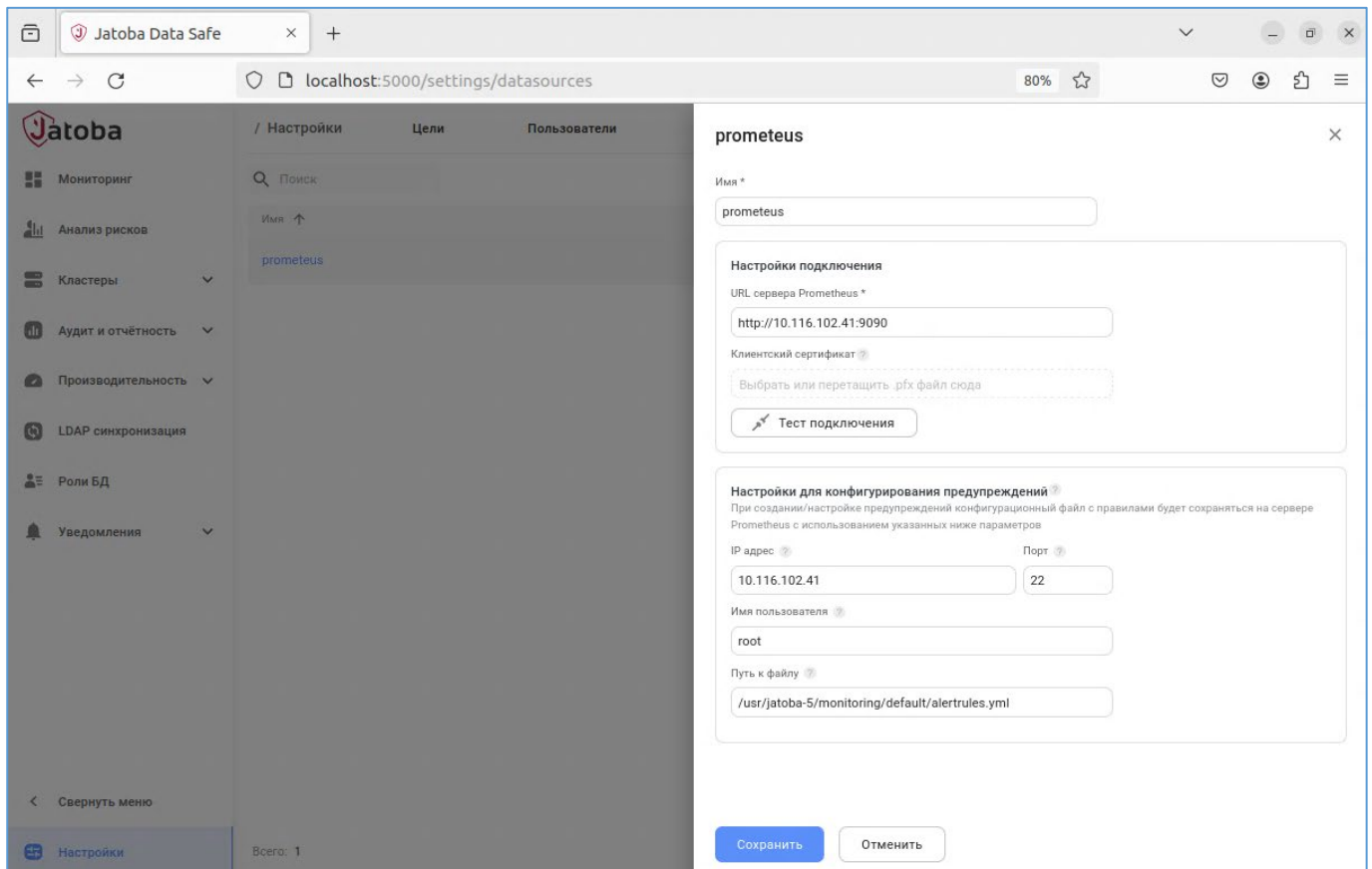


Рисунок 9.1 - «Настройки конфигурации предупреждений»

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

## Раздел «Мониторинг»

В разделе «Мониторинг» создав уведомление в любом из дашбордов с динамическими данными будет сформирован файл с правилами уведомлений по пути указанному в настройках «Источника данных» в сформированном подключении к системе «Prometheus»:

```
/usr/jatoba-5/monitoring/default/alertrules.yml
```

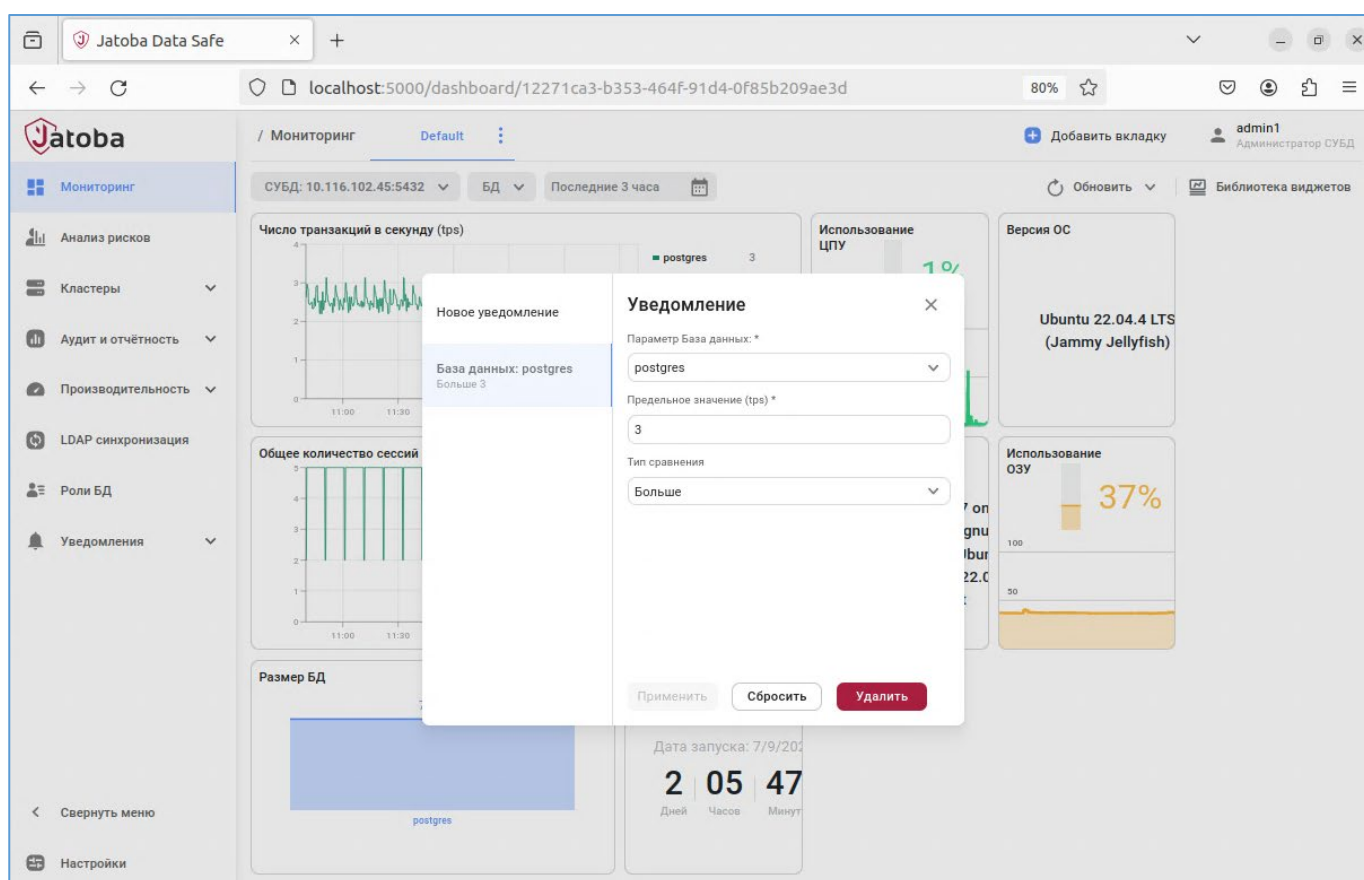


Рисунок 9.2 – Создание уведомления в компоненте JDS

### 9.3. Настройка связки системы «Prometheus» и утилиты «Alertmanager»

Имея данные конфигурации и конфигурационный файл уведомлений можно приступить к связке системы «Prometheus» и утилиты «Alertmanager», для чего надо выполнить команду редактирования конфигурационного файла системы «Prometheus»:

```
gedit usr/jatoba-5/monitoring/default/prometheus.yml
```

Соответствующий раздел «Alertmanager configuration» находится в начале файла и параметры надо внести именно в него. Вставка параметров в конец файла может привести к ошибке.

В узле «targets» указывается хост или хосты с установленной утилитой «Alertmanager»

В узле rule\_files необходимо указать имя конфигурационного файла уведомлений

```
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - IP**.*.*.*.*:9093
rule_files:
  - "alertrules.yml"
```

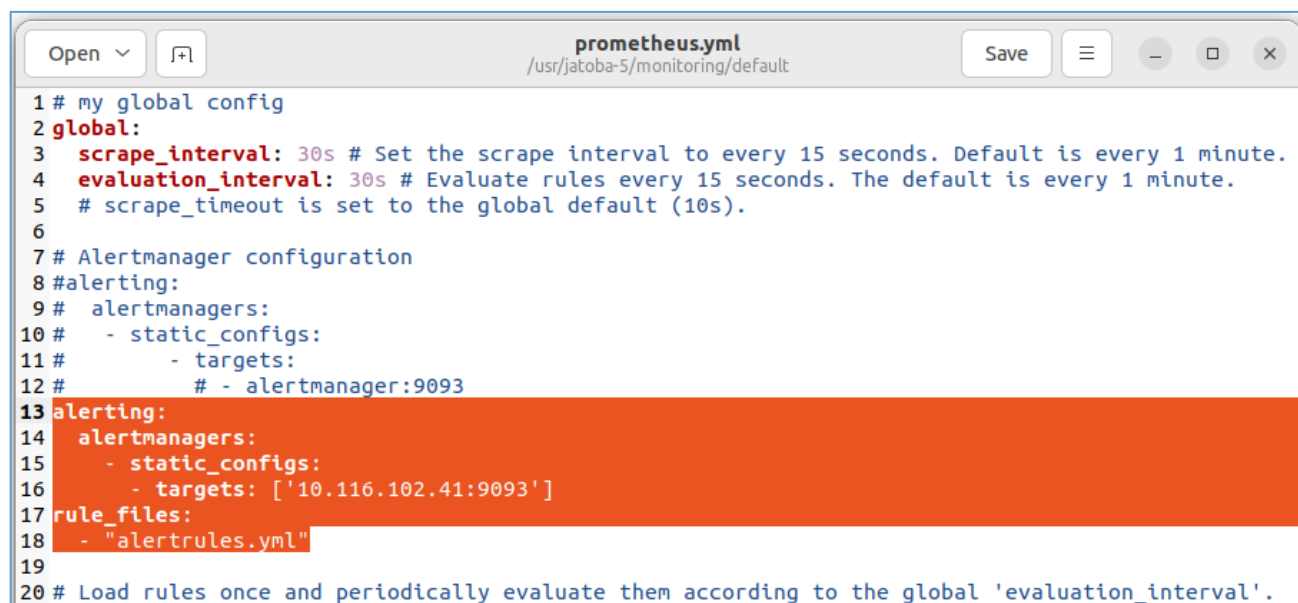


Рисунок 9.3 - Раздел «Alertmanager configuration»

Проверить корректность введенных параметров возможно командами:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
# cd /usr/jatoba-5/bin#  
# ./promtool check config /usr/jatoba-  
5/monitoring/default/prometheus.yml
```

Если параметры верны, перезапустить службу:

```
# systemctl restart jatoba5_prometheus
```

На данном шаге конфигурирование раздела «Мониторинг» компонента JDS закончено.

## 10. НАСТРОЙКА ЭКСПОРТЁРОВ ДЛЯ КОМПОНЕНТА JA\_HIPE\_CLUSTER

Настройка системы мониторинга подразумевает конфигурирование целевых СУБД и кластера на основе компонента «ja\_Hipe\_Cluster» в экосистеме СУБД «Jatoba». Используется выделенный сервер мониторинга «Jatoba Data Safe» собирающий данные для виджетов с сервера «Prometheus». Сервера «Prometheus» аккумулирует данные о СУБД и ОС с кластера и СУБД.

### 10.1. Параметры стенда

Параметры стенда, приведенные в таблице 10.1.

Таблица 10.1 – Конфигурация стенда

№	Имя сервера	IP-адрес	ПО	Port	Роль
1	u602doc-jds01	10.116.102.41/24	JDS		Сервер мониторинга
2	u602doc-pgp01	10.116.102.45/24			Сервер Prometheus
			Prometheus	9090	
			Alert manager	9093, 22	
			pg_stat_statements		
			node_exporter	9100	
			postgres_exporter	9187	
			sql_exporter	9399	
3	u602doc-hipe01	10.116.102.61/24			Coordinator
			citus		
			pg_stat_statements		
			node_exporter	9100	
			postgres_exporter	9187	
			sql_exporter	9399	
4	u602doc-hipe02	10.116.102.62/24			Worker (Node1)
			citus		
			pg_stat_statements		
			node_exporter	9100	
			postgres_exporter	9187	
			sql_exporter	9399	
5	u602doc-hipe03	10.116.102.63/24			Worker (Node2)
			citus		
			pg_stat_statements		
			node_exporter	9100	
			postgres_exporter	9187	
			sql_exporter	9399	
6	u602doc-ldap01	10.116.102.47/24			Целевая СУБД
			pg_stat_statements		
			node_exporter	9100	
			postgres_exporter	9187	
			sql_exporter	9399	

Схема стенда представлена на рисунке 2.1.

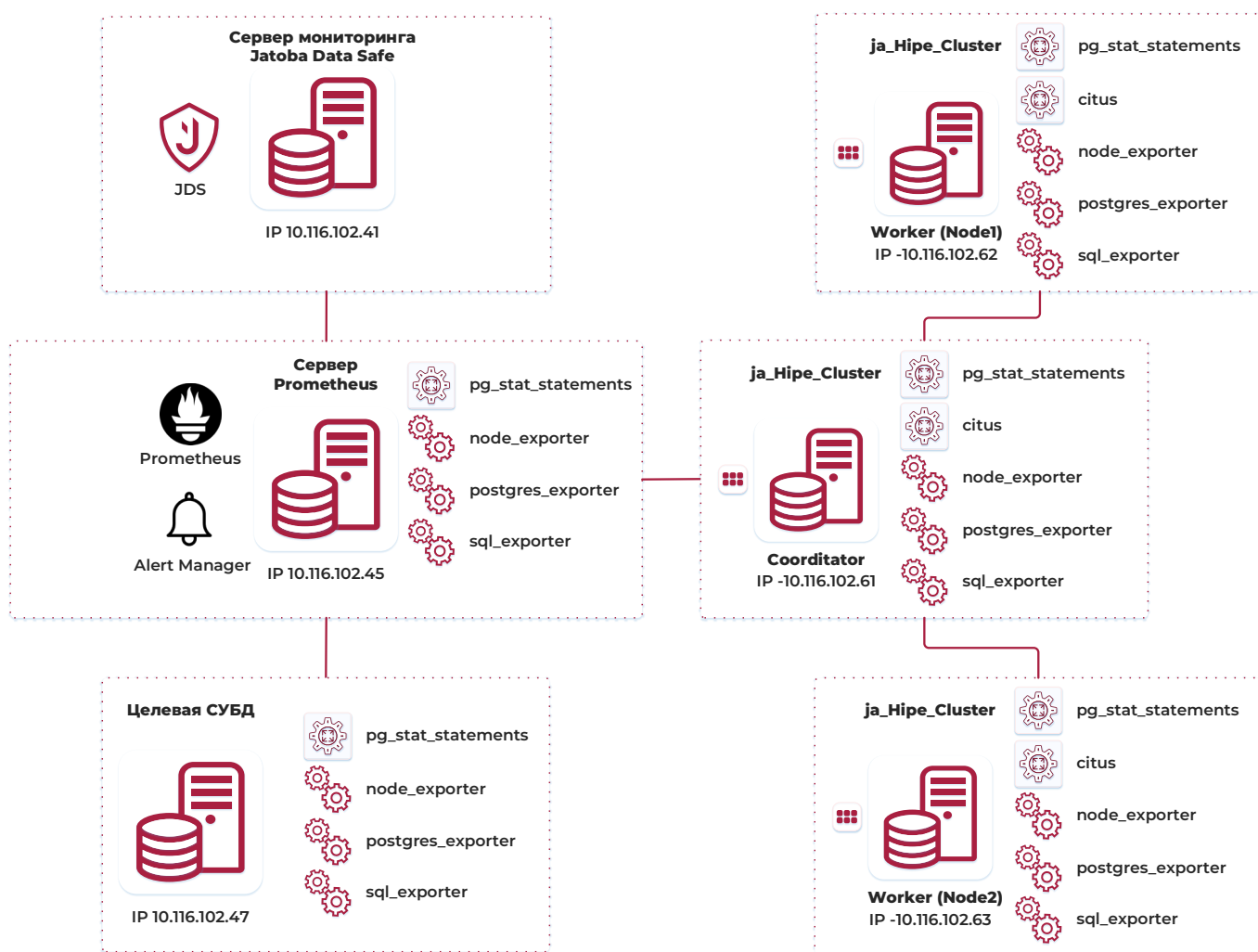


Рисунок 10.1 – Схема стенда мониторинга с кластером «ja\_Hipe\_Cluster»

Настройка экспортеров на целевой СУБД описан выше в настоящем документе и их параметры остаются неизменными в конфигурационном файле `/usr/jatoba-  
<ver>/monitoring/default/prometheus.yml`.

Конфигурация экспортеров кластера добавляется отдельно.

Настройка экспортеров `ja_Hipe_Cluster` для принципиально не отличается от вышеописанного процесса, но имеет ряд особенностей.

Настройка СУБД в кластере для мониторинга выполняется согласно раздела 3 «Установка и настройка целевых СУБД».

В результате:



- должна быть настроена аутентификация;
- в конфигурационный файл «postgresql.conf» добавлены параметры:

```
shared_preload_libraries = 'citus, pg_stat_statements'  
pg_stat_statements'compute_query_id = on  
pg_stat_statements.max = 10000  
pg_stat_statements.track = all
```

- должны быть установлены, как минимум, расширения pg\_stat\_statements, citus и citus\_columnar.

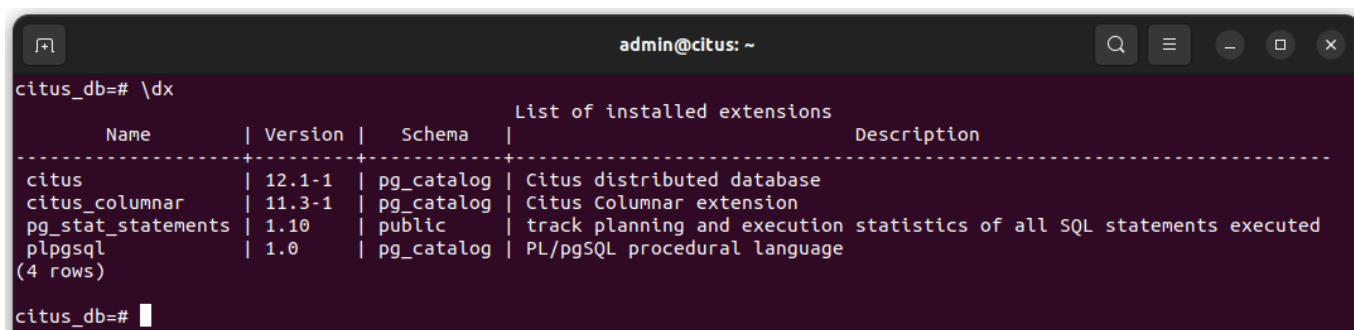


Рисунок 10.2 – Вывод установленных расширений

БД установки расширения pg\_stat\_statements не имеет принципиального значения.

Установка экспортёров выполняется согласно разделов, настоящего документа:

- 4 «Установка экспортёра «jatoba\*\_node\_exporter»;
- 5 «Установка экспортёра «jatoba\*\_postgres\_exporter»;
- 6 «Установка экспортёра «jatoba\*\_sql\_exporter».

В конфигурационных файлах «jatoba\*\_postgres\_exporter» и «jatoba\*\_sql\_exporter» в строке «DATA\_SOURCE\_NAME» должен быть указан IP-адрес узла.

## 10.2. Конфигурирование системы «Prometheus»

На сервере «Prometheus» (u602doc-pgp01, IP-10.116.102.45/24) в конфигурационный файл /usr/jatoba-<ver>/monitoring/default/prometheus.yml добавляется раздел с экспортёрами, установленными на узлах кластера.

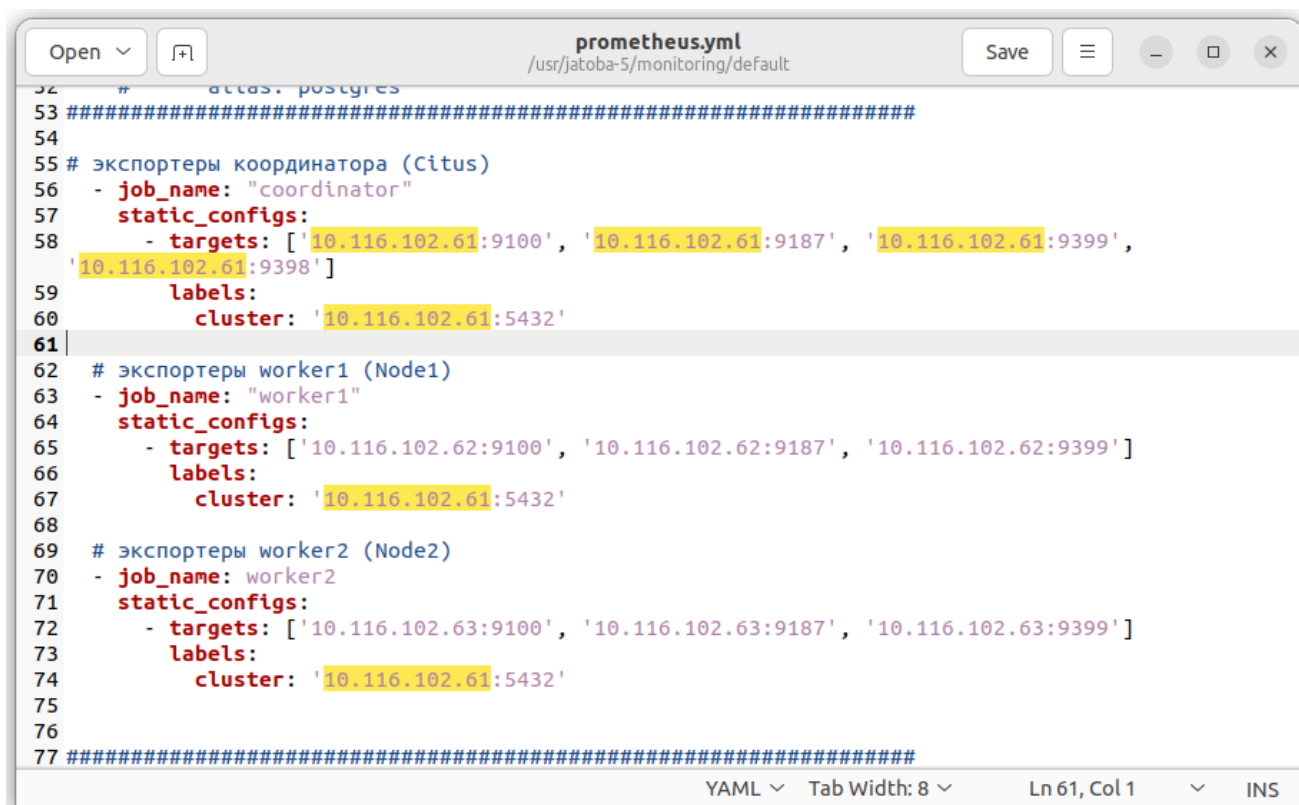
```
#####  
# экспортеры координатора (Citrus)  
- job_name: "coordinator"  
  static_configs:  
    - targets: ['10.116.102.61:9100', '10.116.102.61:9187',  
'10.116.102.61:9399', '10.116.102.61:9398']  
    labels:  
      cluster: 'ja_hipe_cluster'  
  
# экспортеры worker1 (Node1)  
- job_name: "worker1"  
  static_configs:  
    - targets: ['10.116.102.62:9100', '10.116.102.62:9187',  
'10.116.102.62:9399']  
    labels:  
      cluster: 'ja_hipe_cluster'  
  
# экспортеры worker2 (Node2)  
- job_name: worker2  
  static_configs:  
    - targets: ['10.116.102.63:9100', '10.116.102.63:9187',  
'10.116.102.63:9399']  
    labels:  
      cluster: 'ja_hipe_cluster'  
#####
```

В строке «job\_name» целесообразнее указать роль узла в кластере.

В строке «targets» перечисляются IP-адрес и порт каждого из установленных экспортёров, установленных на узле.

Строка «labels: cluster:» является меткой для формирования выпадающего списка «Кластеры» по которой формируется кластерная панель виджетов.

Строка является текстовой. Допускается указание IP-адреса узла, как показано на рисунке 10.3.




```

52 # atlas.postgres
53 #####
54
55 # экспортеры координатора (Citrus)
56 - job_name: "coordinator"
57   static_configs:
58     - targets: ['10.116.102.61:9100', '10.116.102.61:9187', '10.116.102.61:9399',
59 '10.116.102.61:9398']
60     labels:
61       cluster: '10.116.102.61:5432'
62
63 # экспортеры worker1 (Node1)
64 - job_name: "worker1"
65   static_configs:
66     - targets: ['10.116.102.62:9100', '10.116.102.62:9187', '10.116.102.62:9399']
67     labels:
68       cluster: '10.116.102.61:5432'
69
70 # экспортеры worker2 (Node2)
71 - job_name: worker2
72   static_configs:
73     - targets: ['10.116.102.63:9100', '10.116.102.63:9187', '10.116.102.63:9399']
74     labels:
75       cluster: '10.116.102.61:5432'
76
77 #####
  
```

Рисунок 10.3 – Использование IP-адреса узла в строке «labels: cluster:»

Целесообразнее указывать идентификационный признак подключенного к мониторингу кластера, как показано на рисунке 10.4.



```

54 #####
55
56 # экспортеры координатора (Citrus)
57 - job_name: "coordinator"
58   static_configs:
59     - targets: ['10.116.102.61:9100', '10.116.102.61:9187', '10.116.102.61:9399',
60 '10.116.102.61:9398']
61     labels:
62       cluster: 'ja_hipe_cluster'
63
64 # экспортеры worker1 (Node1)
65 - job_name: "worker1"
66   static_configs:
67     - targets: ['10.116.102.62:9100', '10.116.102.62:9187', '10.116.102.62:9399']
68     labels:
69       cluster: 'ja_hipe_cluster'
70
71 # экспортеры worker2 (Node2)
72 - job_name: worker2
73   static_configs:
74     - targets: ['10.116.102.63:9100', '10.116.102.63:9187', '10.116.102.63:9399']
75     labels:
76       cluster: 'ja_hipe_cluster'
77
78 #####
  
```

Рисунок 10.4 - Использование имени кластера в строке «labels: cluster:»

В результате на панели виджетов для кластера, в выпадающем списке отразится имя кластера.

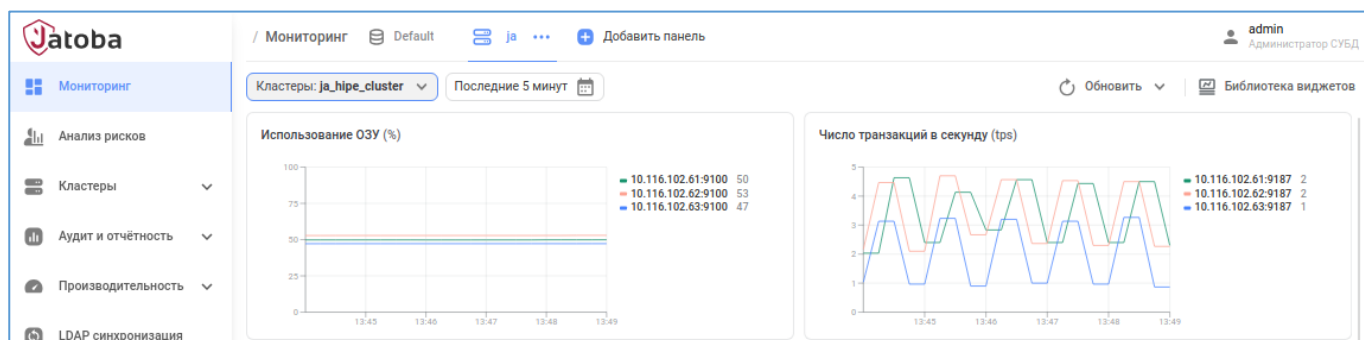


Рисунок 10.5 – Имя кластера на панели виджетов

На виджетах отразятся графы каждого узла кластера.

## 11. ОБНОВЛЕНИЕ СИСТЕМЫ «PROMETHEUS»

Обновление версии системы «Prometheus» требует сохранения конфигурационных файлов системы и выполняется в следующем порядке:

- 1) Сохранить резервную копию каталога конфигураций /usr/jatoba-<ver>/monitoring/default
- 2) Скопировать новый пакет в отдельную директорию. Например, в директорию /home/admin/Downloads.
- 3) Удалить существующий пакет

```
apt remove jatoba5-prometheus -y
```

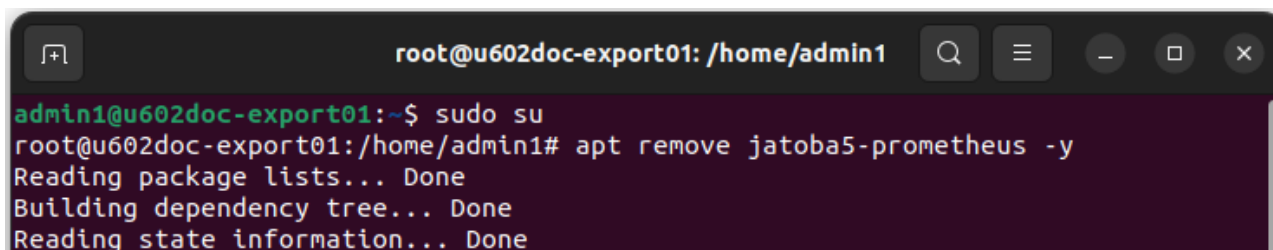


Рисунок 11.1 – Удаление существующего пакета

- 4) Установить новый пакет от имени и с правами привилегированного пользователя ОС:

– Находясь в каталоге с пакетом:

```
apt install ./jatoba6-prometheus_* -y
```

– Из корневого каталога командой:

```
apt install /home/admin/Downloads/jatoba5-prometheus* -y
```

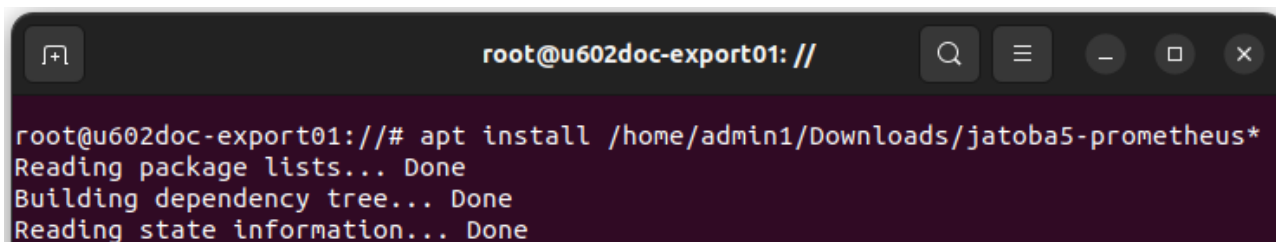


Рисунок 11.2 – Установка нового пакета

- 5) Восстановить резервные копии конфигураций в каталог /usr/jatoba-<ver>/monitoring/default

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

6) Проверить владельца и права на каталог приложения командой:

```
ls -la /opt/prometheus/
```

7) В случае отклонения настроек назначить корректные права командой:

```
chown prometheus:prometheus -R /opt/prometheus/
```

8) Обновить конфигурацию systemd:

```
# systemctl daemon-reload
```

9) Запустить службу система, включить ее в автозапуск и проверить статус работы:

```
# systemctl start jatoba<ver>_prometheus  
# systemctl enable jatoba<ver>_prometheus  
# systemctl status jatoba<ver>_prometheus
```

На данном шаге обновление системы «Prometheus» закончено.

В момент обновления компонента данные с экспортёров будут потеряны, т.к. экспортёры не хранят данные, а передают их постоянно.

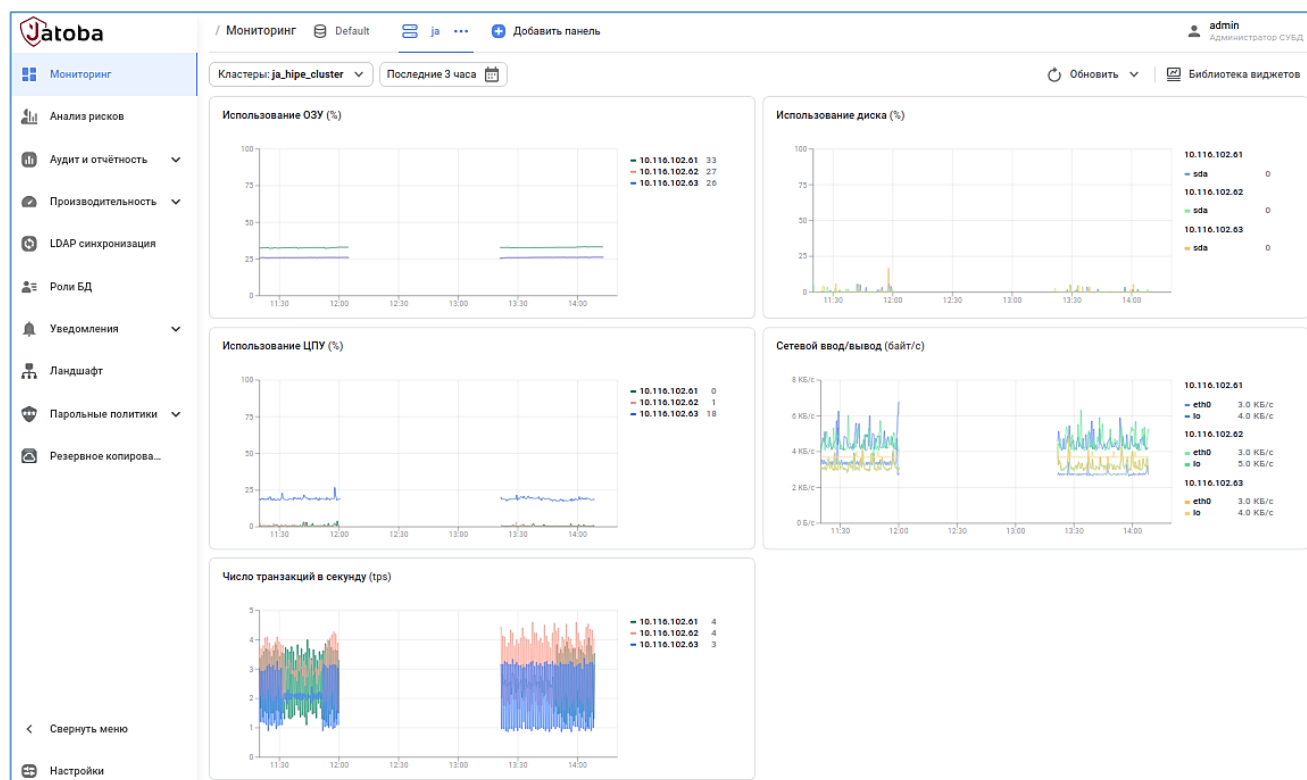


Рисунок 11.3 – Промежуток в данных собранных с экспортёров

## ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

@ - ГОСТ 34.302.2-91 (ИСО 8859/2-87) «Наборы 8 битных однобайтовых кодированных графических символов. латинский алфавит № 2» определяет символ как, «коммерческое эт» (англ. «commercial at»).

& - ГОСТ 34.302.2-91 (ИСО 8859/2-87) «Наборы 8 битных однобайтовых кодированных графических символов. латинский алфавит № 2» определяет символ как, «коммерческое и» (амперсанд) (англ. ampersand)

= - ГОСТ 34.302.2-91 (ИСО 8859/2-87) «Наборы 8 битных однобайтовых кодированных графических символов. латинский алфавит № 2» определяет символ как, «равно» (англ. equals SIGN).

? - ГОСТ 34.302.2-91 (ИСО 8859/2-87) «Наборы 8 битных однобайтовых кодированных графических символов. латинский алфавит № 2» определяет символ как, «вопросительный знак» (англ. question mark).

: - ГОСТ 34.302.2-91 (ИСО 8859/2-87) «Наборы 8 битных однобайтовых кодированных графических символов. латинский алфавит № 2» определяет символ как, «двоеточие» (англ. colon).

## **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ**

SQL	–	Structured Query Language
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных



[illegible]

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------