ПРОГРАММНЫЙ КОМПЛЕКС ASTRAREGUL

РГДП.58.29.14.000-001-08 РП

Системы отчетности

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Редакция 4

СПИСОК ИЗМЕНЕНИЙ

Редакция	Список изменений
Редакция 4	- Обновлен раздел Настройка СУБД. - Добавлен раздел Пример настройки конфигурационного файла. - Добавлены разделы по установке и настройке Alpha.Reports на AstraLinux и РЕД ОС 7.3.
Редакция 3	- Актуализированы скриншоты под версию Alpha.Reports 1.1.6.125.
Редакция 2	- Обновлен раздел Установка компонентов. - Добавлен раздел Настройка СУБД. - Обновлен раздел Секретная строка.

ОГЛАВЛЕНИЕ

СПИСОК ИЗМЕНЕНИЙ 2
1. Системы отчетности 5
1.1. Alpha.Reports 6
1.1.1. Установка компонентов 7
1.1.1.1. Windows 8
1.1.1.1.1. Alpha.Reports.Base 9
1.1.1.1.2. Alpha.Reports.Server 13
1.1.1.1.3. PostgreSQL 17
1.1.1.2. AstraLinux 26
1.1.1.2.1. Alpha.Reports 27
1.1.1.2.2. PostgreSQL 29
1.1.1.3. РЕД ОС 7.3 30
1.1.1.3.1. Alpha.Reports 31
1.1.1.3.2. PostgreSQL 32
1.1.2. Настройка 34
1.1.2.1. Windows 35
1.1.2.1.1. Настройка СУБД 36
1.1.2.1.1.1. Примеры настройки конфигурационного файла 48
1.1.2.1.2. Настройка Alpha.Reports.Base 51
1.1.2.1.2.1. Пример файла конфигурации 54
1.1.2.1.3. Настройка Alpha.Reports.Server 56
1.1.2.1.3.1. Пример файла appsettings.json 58
1.1.2.1.4. Секретная строка 59
1.1.2.1.5. Источники данных 60
1.1.2.2. AstraLinux 62
1.1.2.2.1. Настройка СУБД 63
1.1.2.2.1.1. Примеры настройки конфигурационного файла 71
1.1.2.2.2. Настройка Alpha.Reports.Base 74
1.1.2.2.2.1. Пример файла конфигурации 76
1.1.2.2.3. Настройка Alpha.Reports.Server 78
1.1.2.2.3.1. Пример файла appsettings.json 80
1.1.2.2.4. Секретная строка 81

1.1.2.2.5. Источники данных 82
1.1.2.3. РЕД ОС 7.3 85
1.1.2.3.1. Настройка СУБД 86
1.1.2.3.1.1. Примеры настройки конфигурационного файла 95
1.1.2.3.2. Настройка Alpha.Reports.Base 98
1.1.2.3.2.1. Пример файла конфигурации 100
1.1.2.3.3. Настройка Alpha.Reports.Server 102
1.1.2.3.3.1. Пример файла appsettings.json 104
1.1.2.3.4. Секретная строка 105
1.1.2.3.5. Источники данных 106
1.1.3. Дизайнер отчетов 109
1.1.3.1. Создание отчета 113
1.1.4. Конфигуратор сервера отчетов 117

1. Системы отчетности

Система отчетности	Версия	Компания
Alpha.Reports	1.1.7.129	Атомик Софт

1.1. Alpha.Reports

Для формирования отчетов можно использовать сторонний программный компонент Alpha.Reports от компании "Атомик Софт".

Программный компонент Alpha.Reports лицензируется согласно действующей политике лицензирования компании "Атомик Софт".

1.1.1. Установка компонентов

Windows
AstraLinux
РЕД ОС 7.3

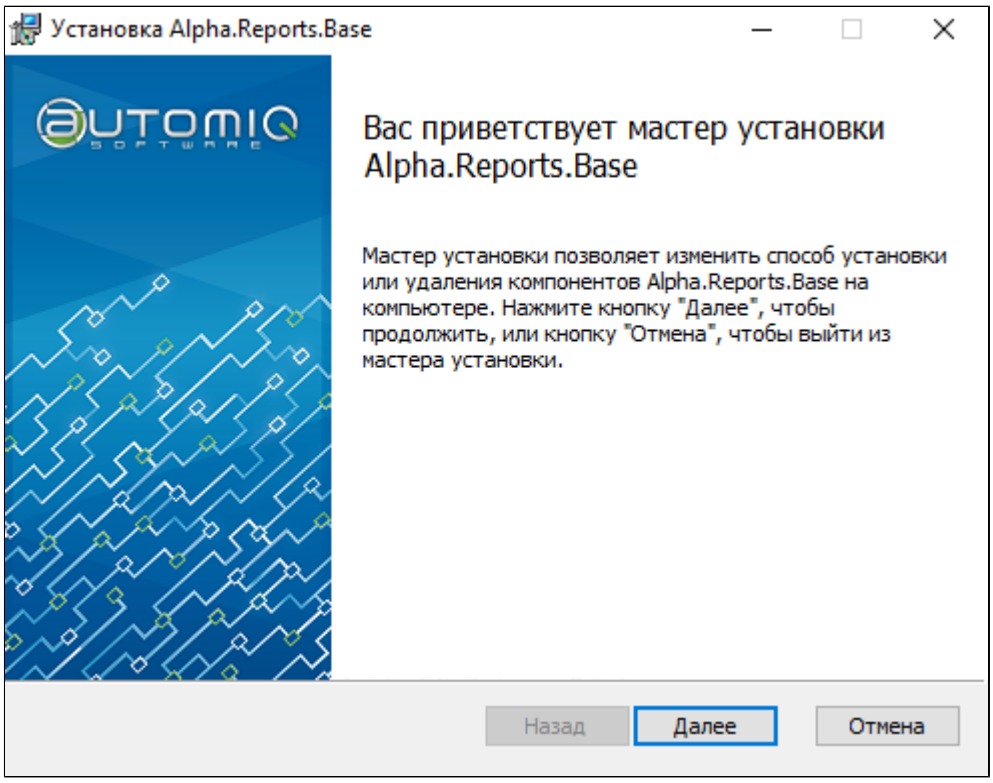
1.1.1.1. Windows

Alpha.Reports.Base
Alpha.Reports.Server
PostgreSQL

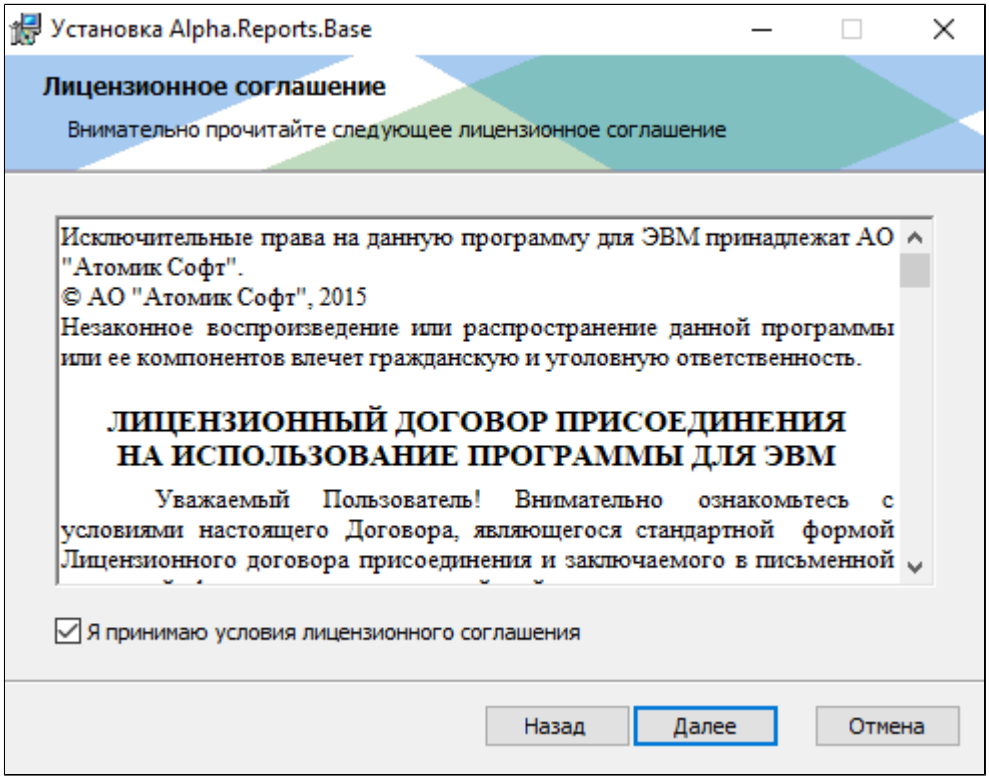
1.1.1.1.1. Alpha.Reports.Base

Чтобы установить программный компонент Alpha.Reports.Base, выполните следующие действия:

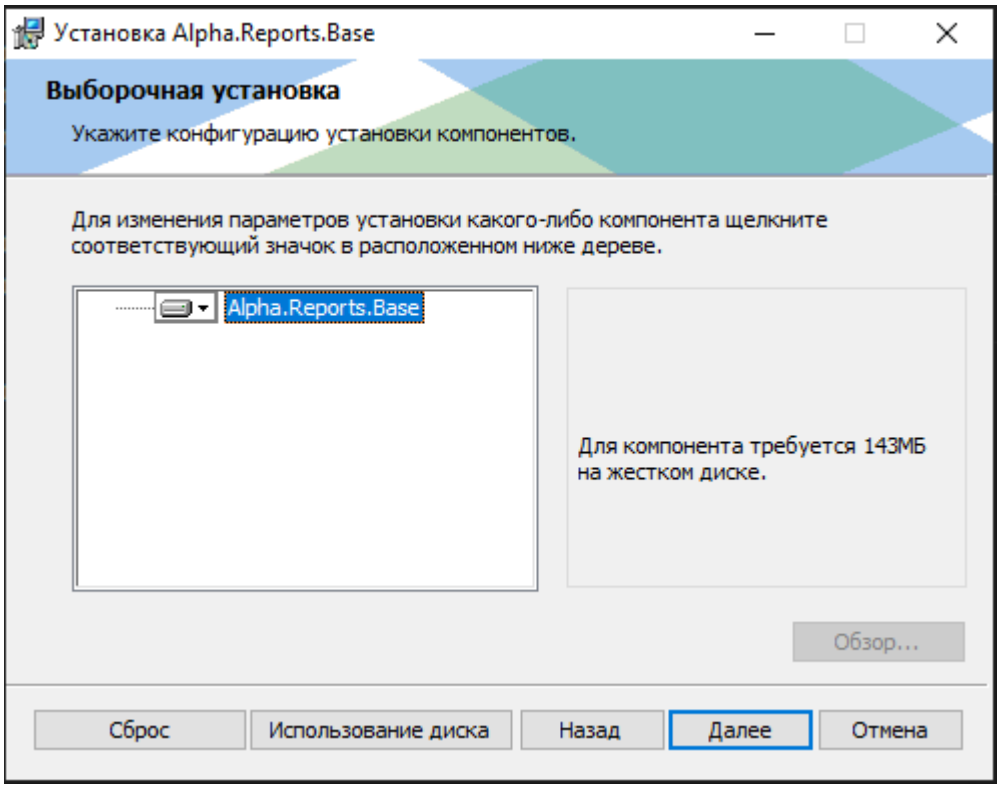
1. Запустите дистрибутив установки "Alpha.Reports.Base.Win.x64.msi". Откроется мастер установки. Нажмите кнопку "Далее".



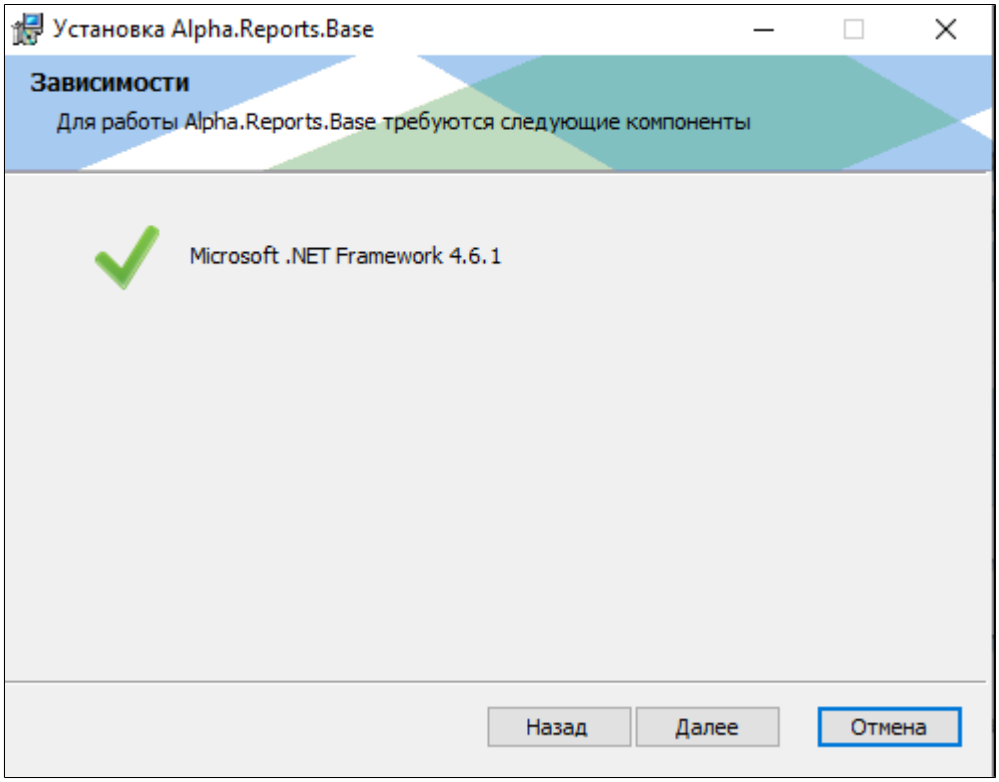
2. Внимательно ознакомьтесь с лицензионным соглашением и установите флаг "Я принимаю условия лицензионного соглашения".



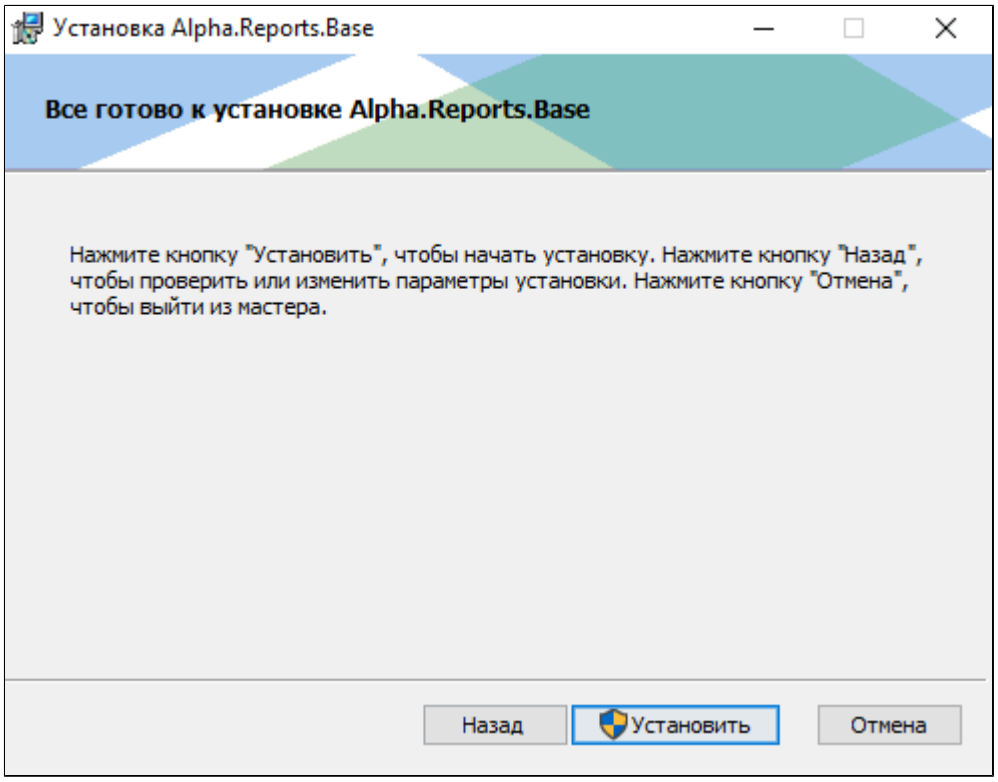
3. Выберите программные компоненты, которые необходимо установить и нажмите кнопку "Далее".



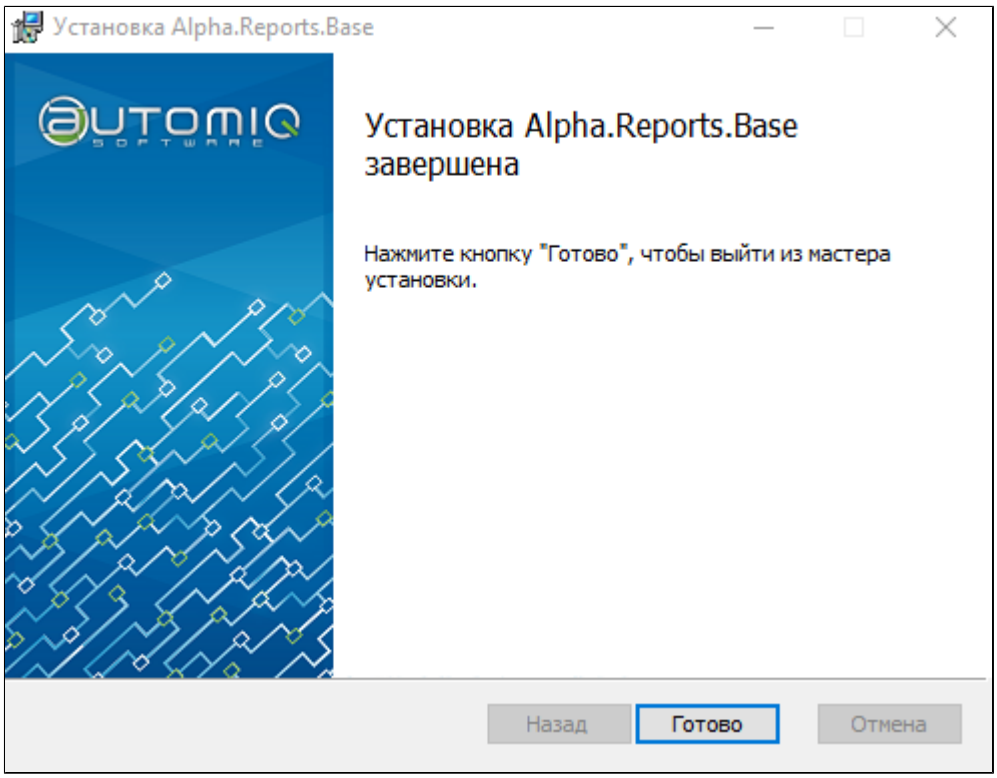
4. Мастер установки уведомит о наличии требуемых дополнительных компонентов. Для продолжения установки Alpha.Reports.Base установите все необходимые дополнительные компоненты и нажмите кнопку "Далее".



5. Подтвердите установку, нажав кнопку "Установить".



6. Дождитесь окончания установки компонента и нажмите кнопку "Готово", чтобы выйти из мастера установки.



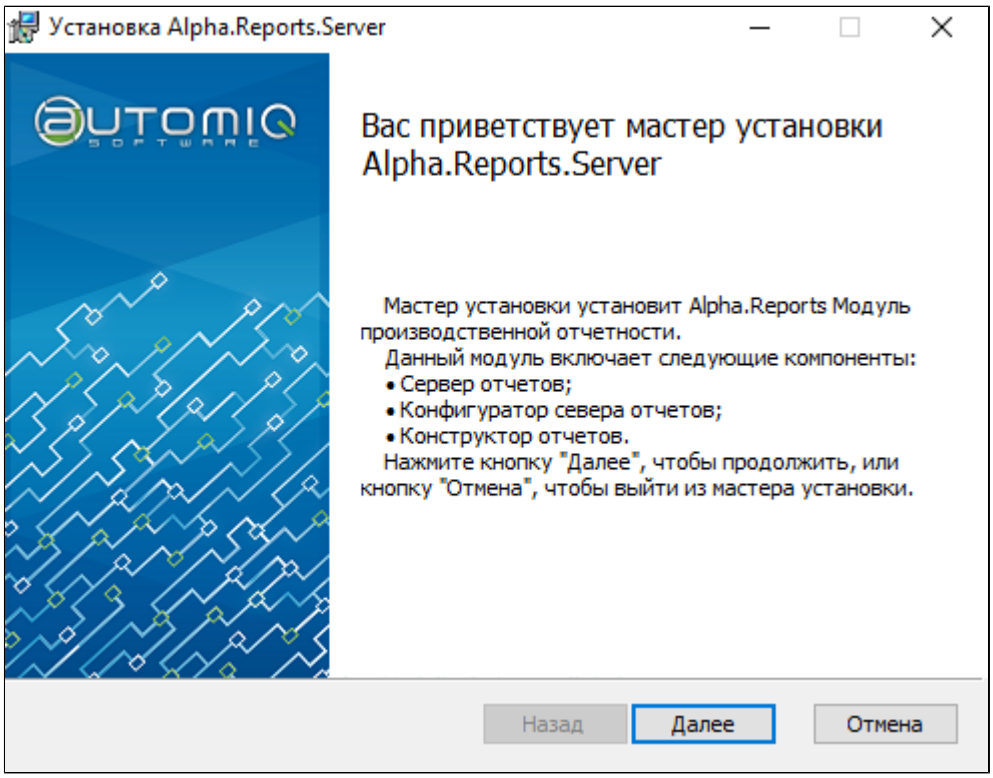
Каталог установки:

C:\Program Files\Automiq\Alpha.Reports

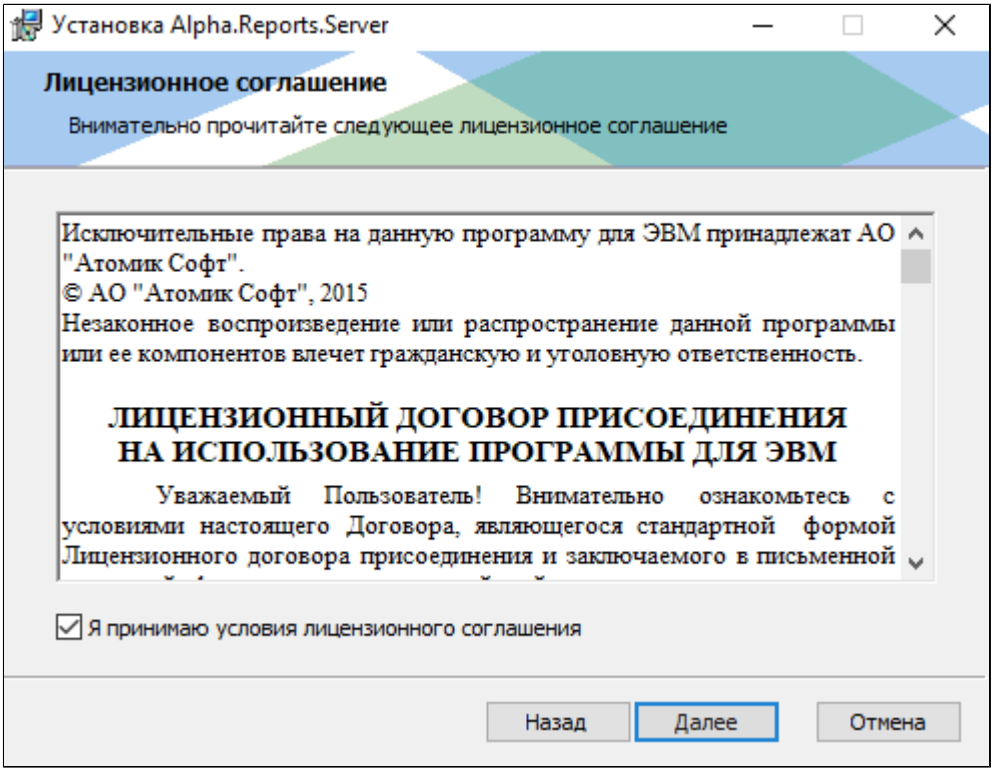
1.1.1.1.2. Alpha.Reports.Server

Чтобы установить программный компонент Alpha.Reports.Server, выполните следующие действия:

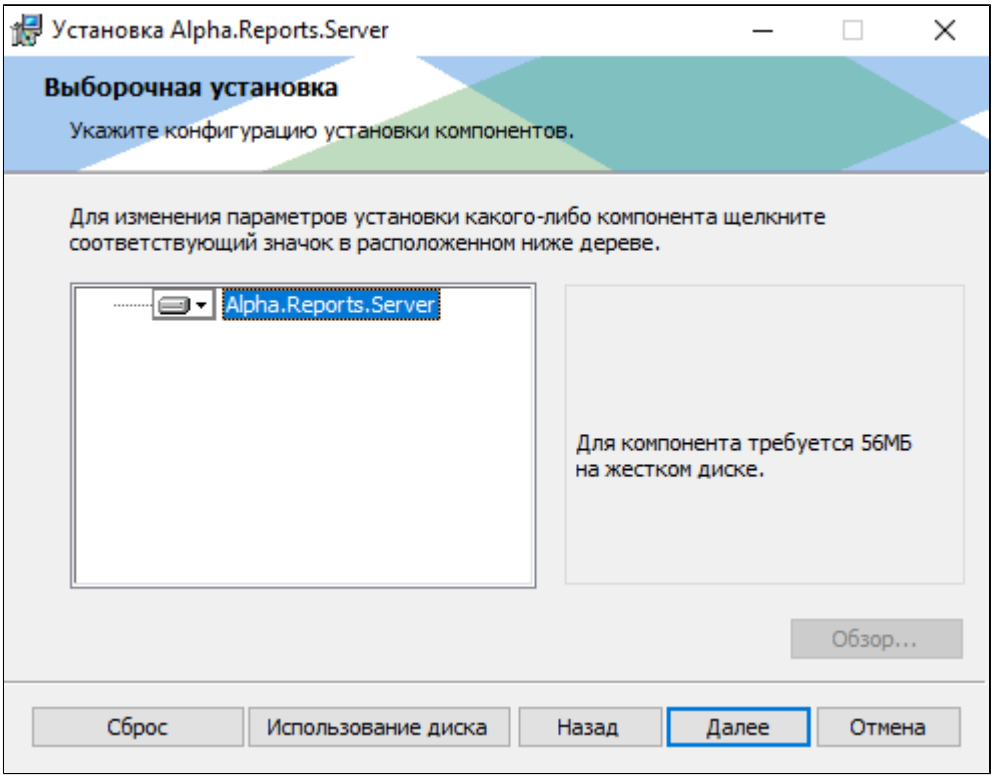
1. Запустите дистрибутив установки "Alpha.Reports.Server.Win.x64.msi". Откроется мастер установки. Нажмите кнопку "Далее".



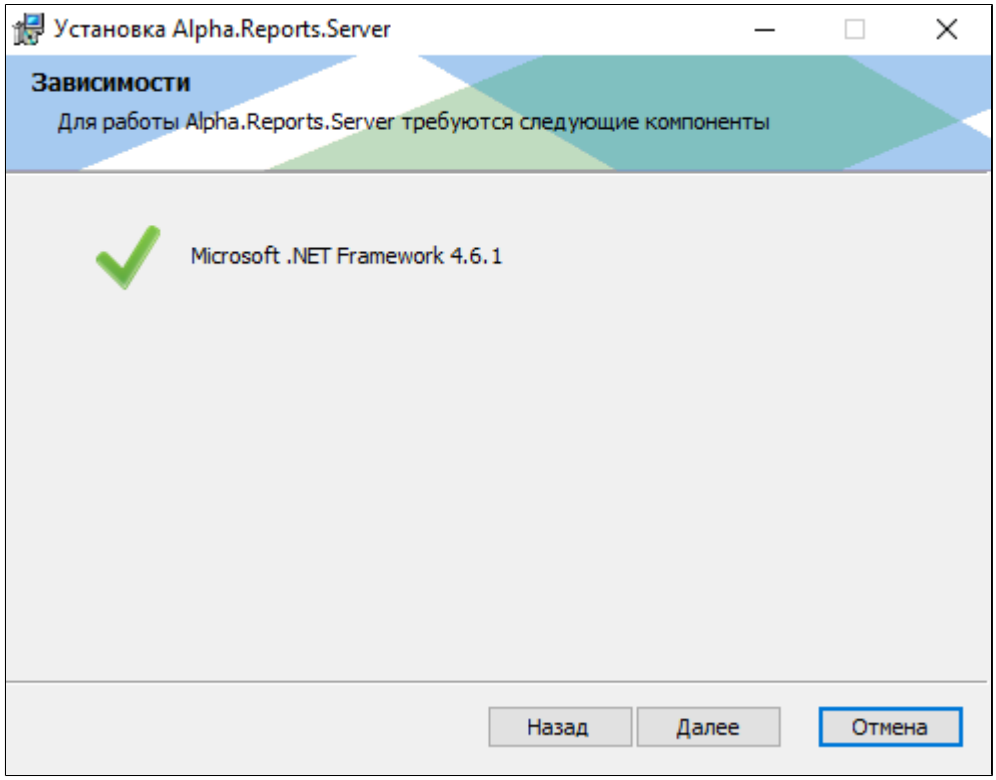
2. Внимательно ознакомьтесь с лицензионным соглашением и установите флаг "Я принимаю условия лицензионного соглашения".



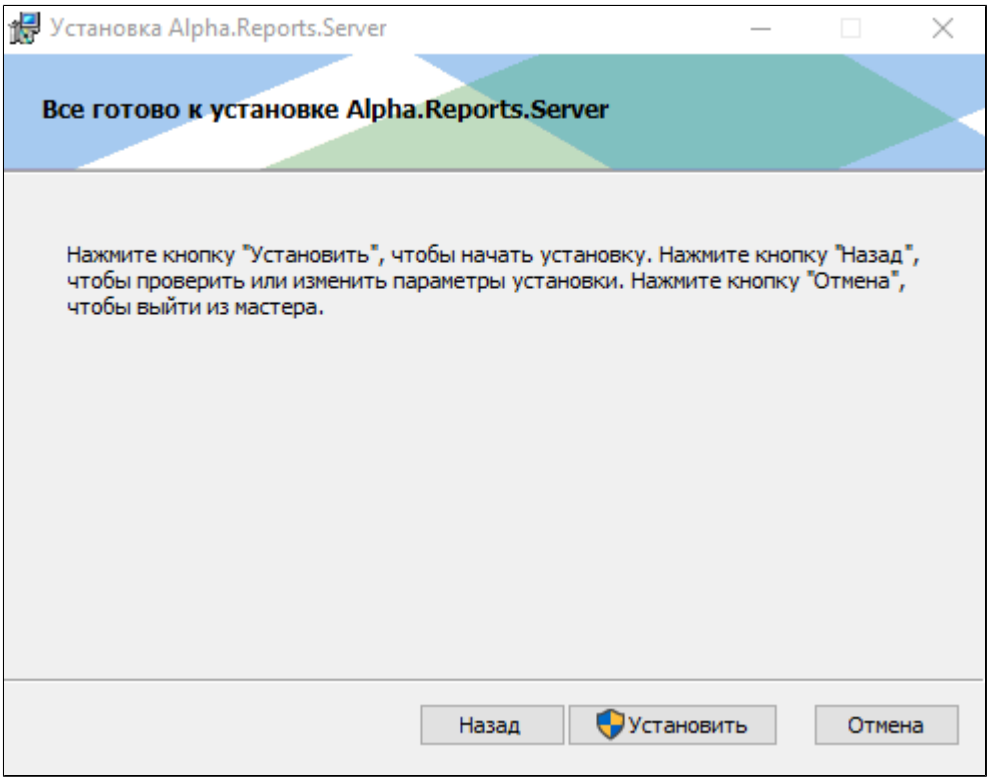
3. Выберите программные компоненты, которые необходимо установить и нажмите кнопку "Далее".



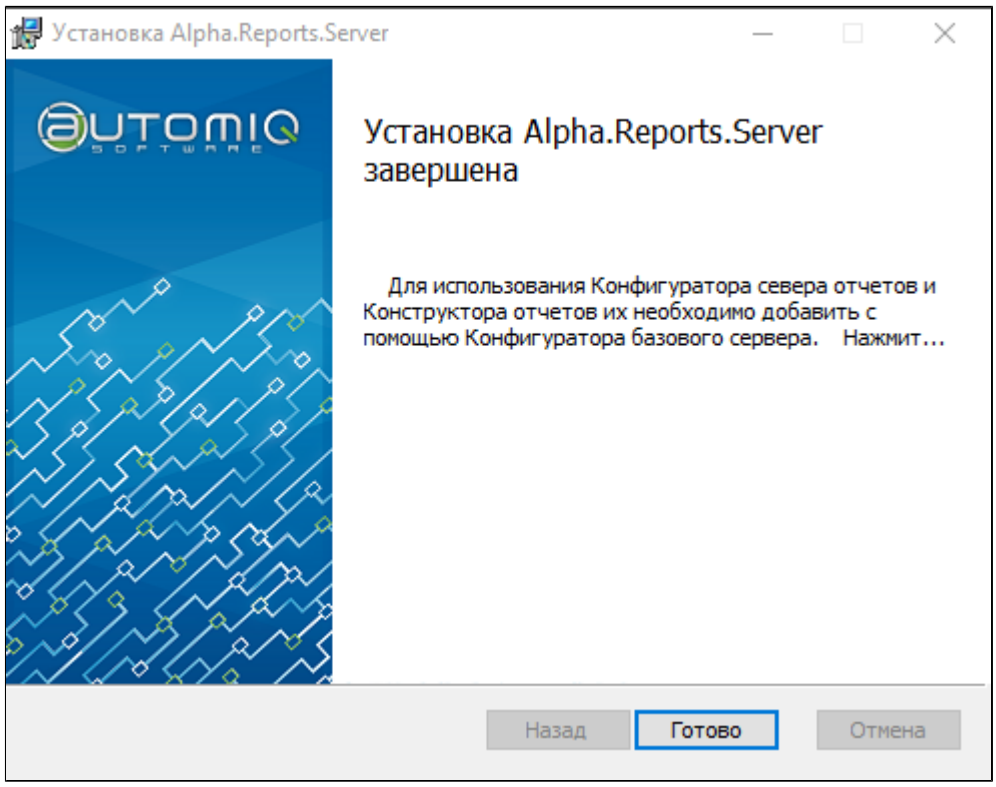
4. Мастер установки уведомит о наличии требуемых дополнительных компонентов. Для продолжения установки Alpha.Reports.Base установите все необходимые дополнительные компоненты и нажмите кнопку "Далее".



5. Подтвердите установку, нажав кнопку "Установить".



6. Дождитесь окончания установки компонента и нажмите кнопку "Готово", чтобы выйти из мастера установки.



Каталог установки:

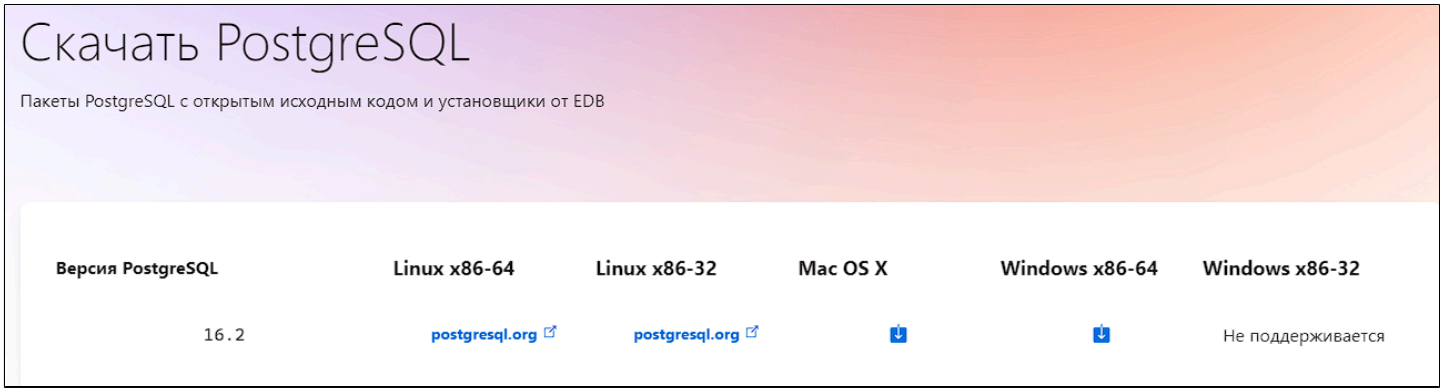
C:\Program Files\Automiq\Alpha.Reports

1.1.1.1.3. PostgreSQL

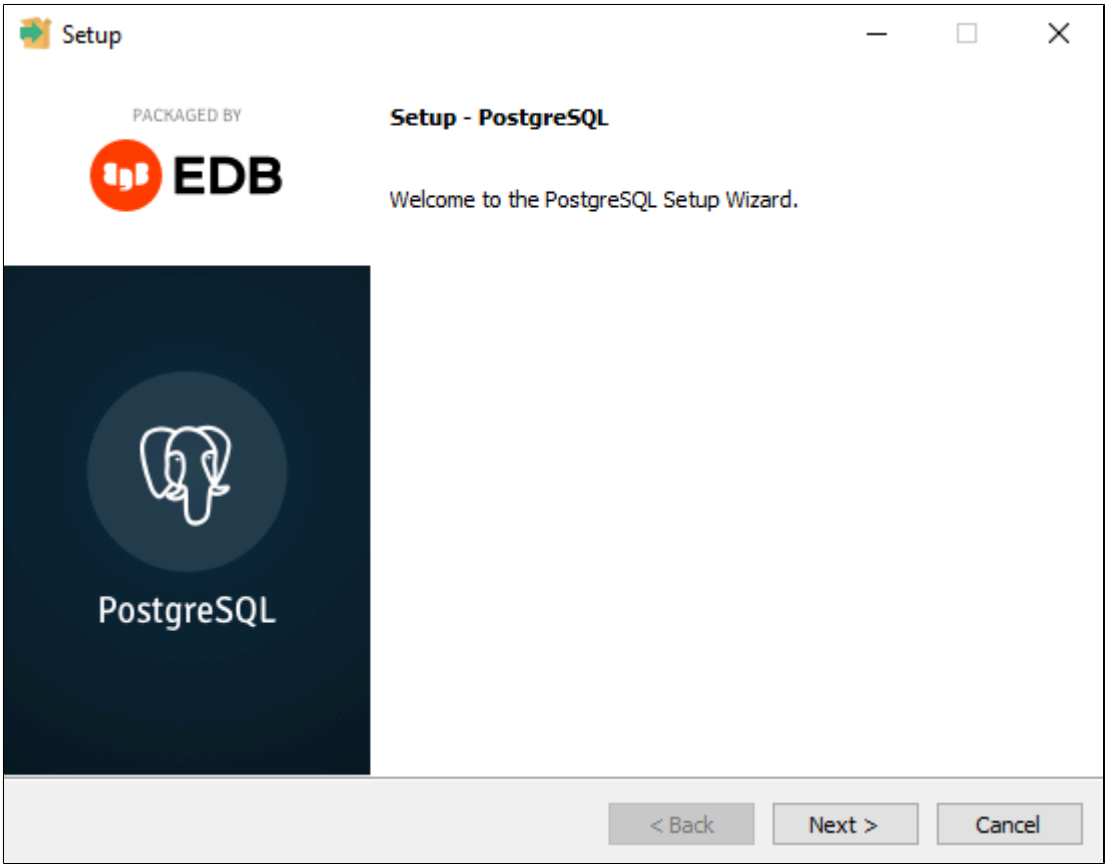
> Установку PostgreSQL производить с остановленными службами Alpha.Reports.Base и Alpha.Reports.Server.

Чтобы установить PostgreSQL, выполните следующие действия:

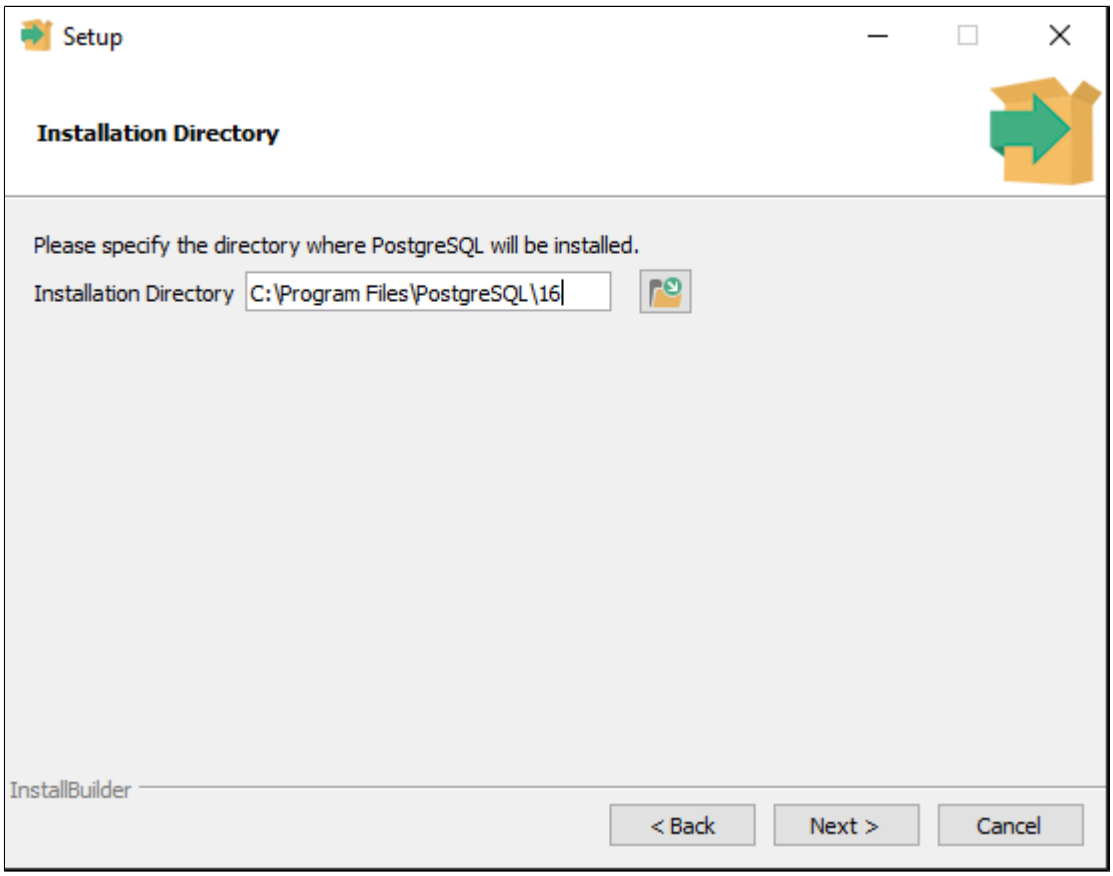
1. Скачайте PostgreSQL версии 11 или выше с официального сайта производителя:



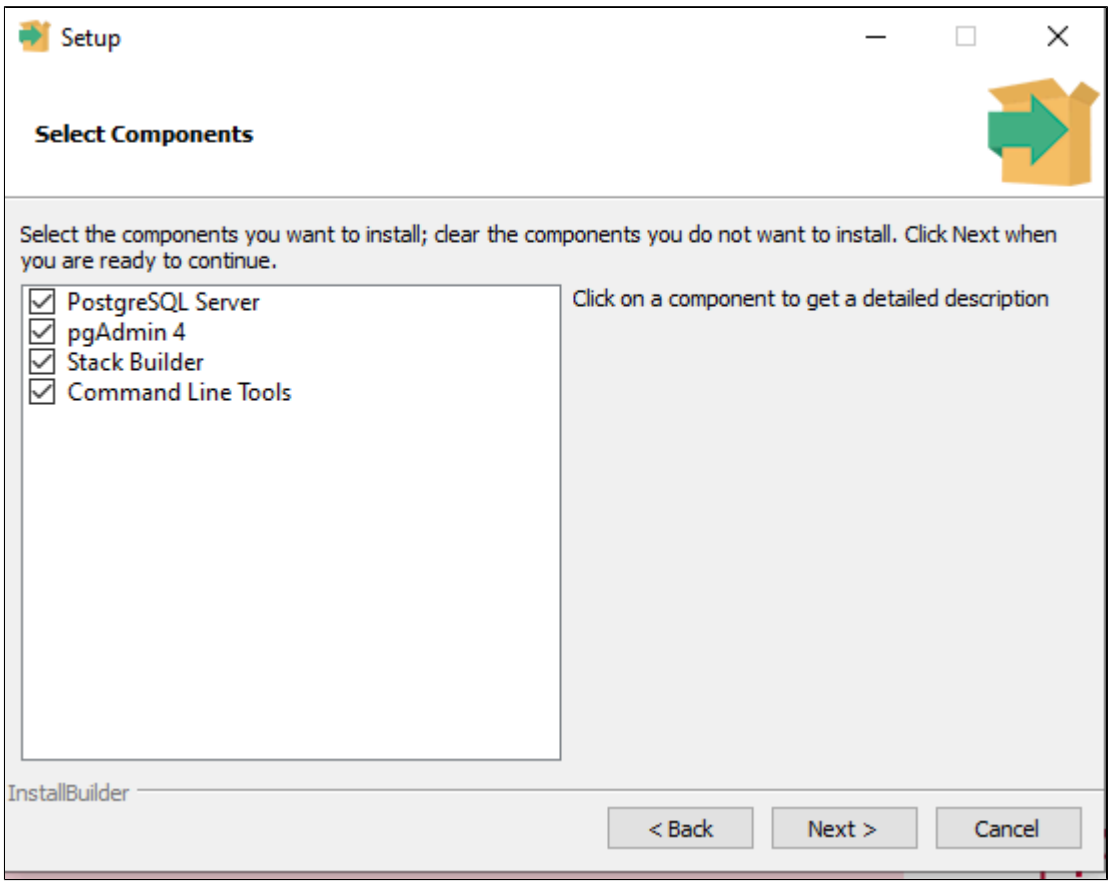
2. Запустите дистрибутив установки postgresql и нажмите кнопку "Next".



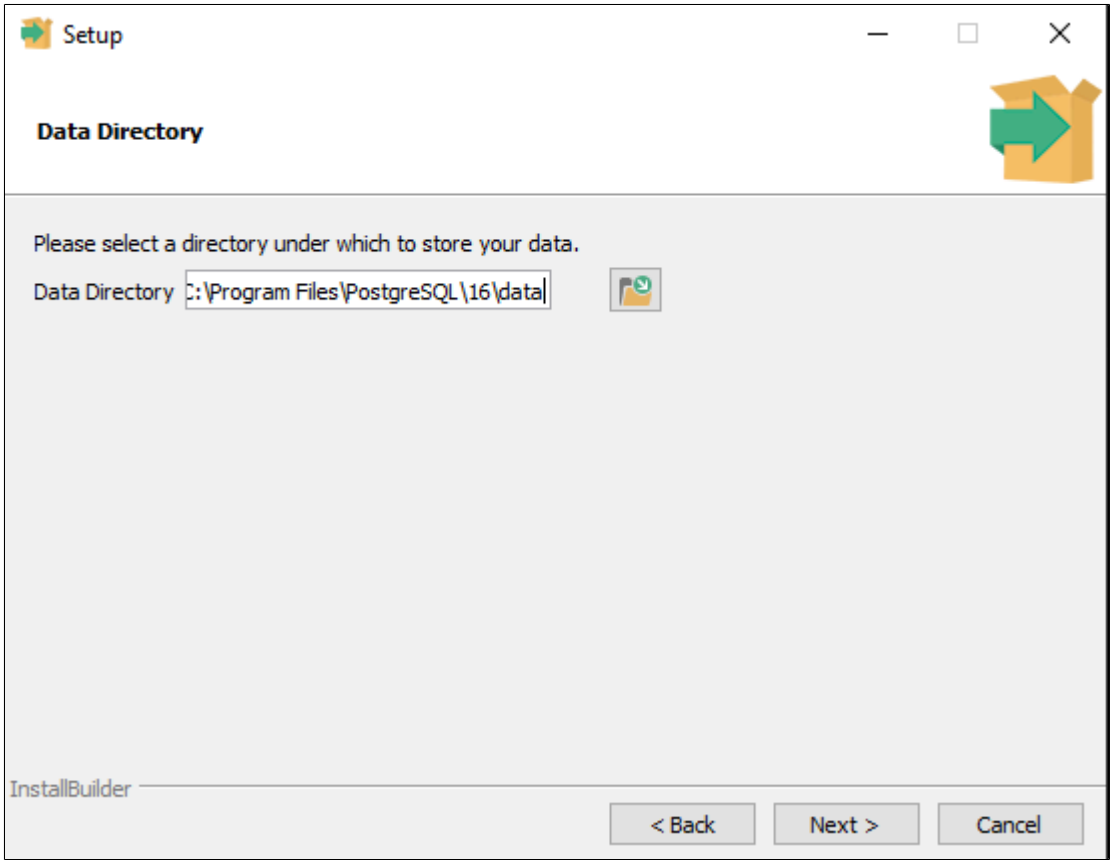
3. Выберите каталог установки и нажмите кнопку "Next".



4. На вкладке выбора компонентов отметьте установку "PgAdmin" и нажмите кнопку "Next".

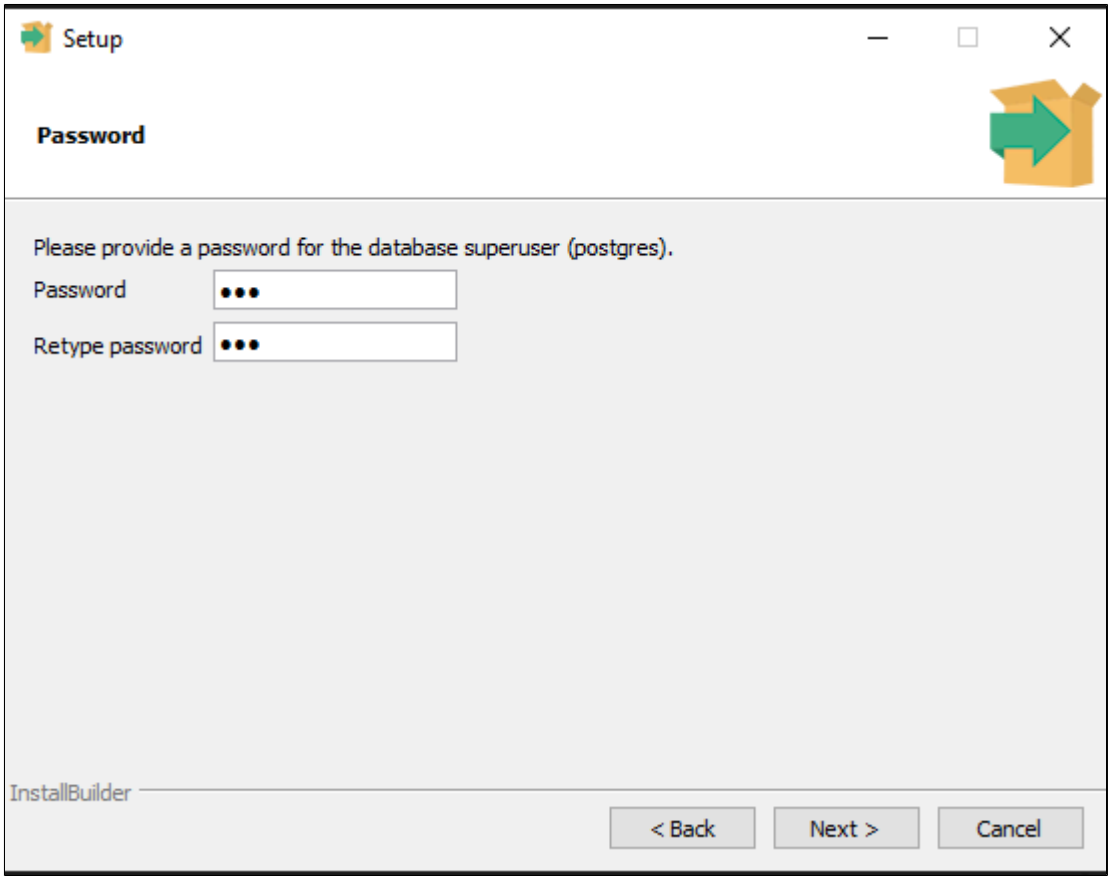


5. Выберите каталог для хранения данных и нажмите кнопку "Next".

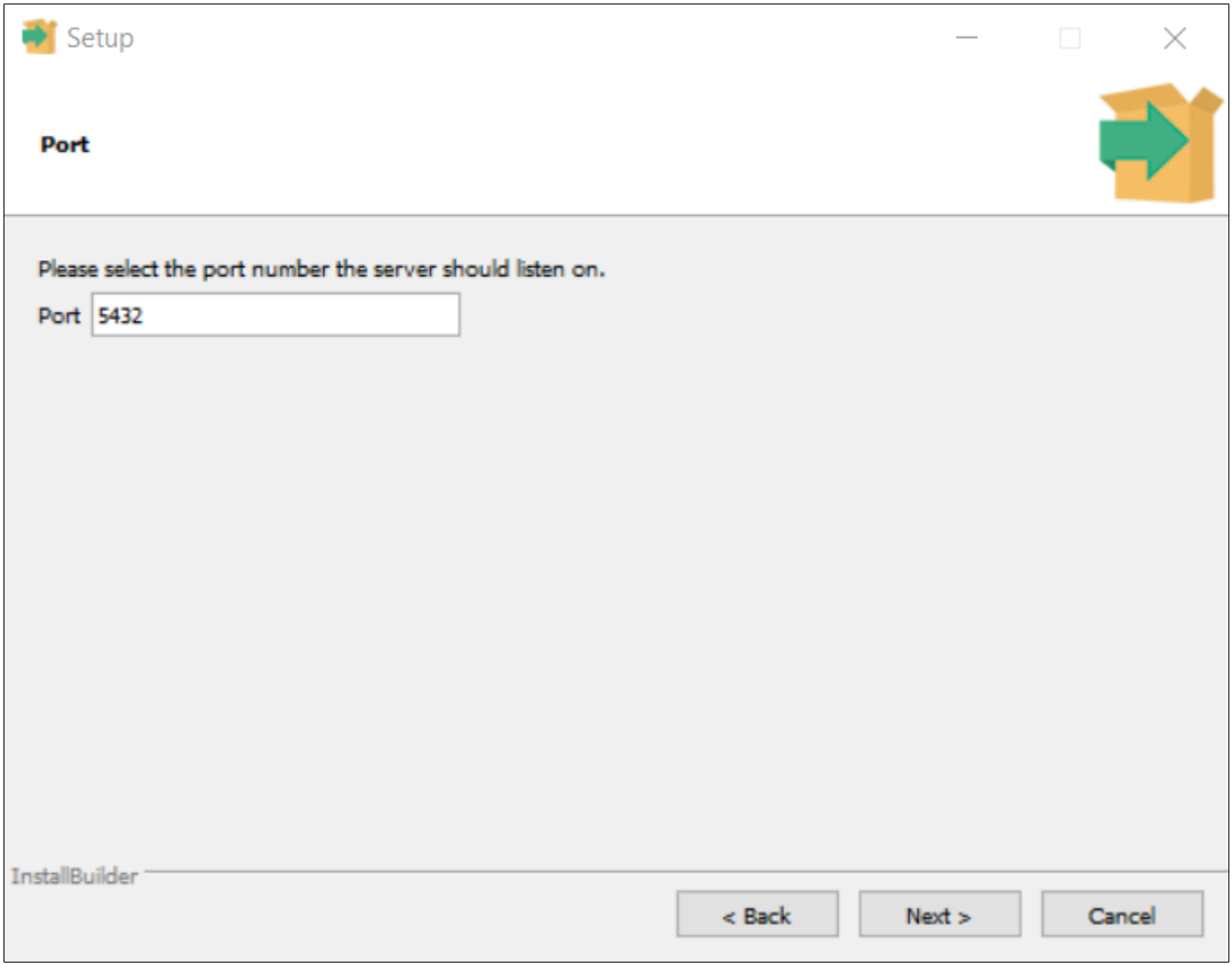


6. Введите пароль для суперпользователя "postgres" (например, пароль "123") и нажмите кнопку "Next".

Этот пароль будет необходим для настройки подключения системы отчетности.

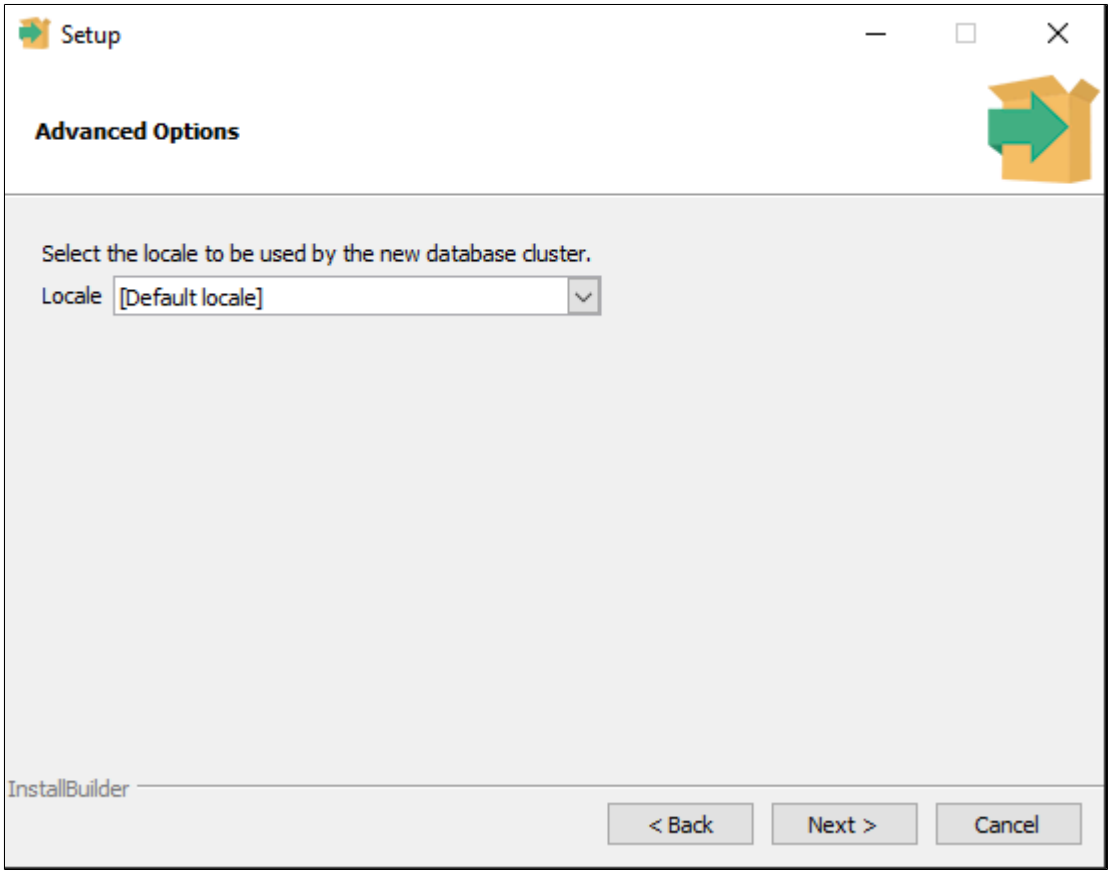


7. Укажите порт сервера "5432" и нажмите кнопку "Next".

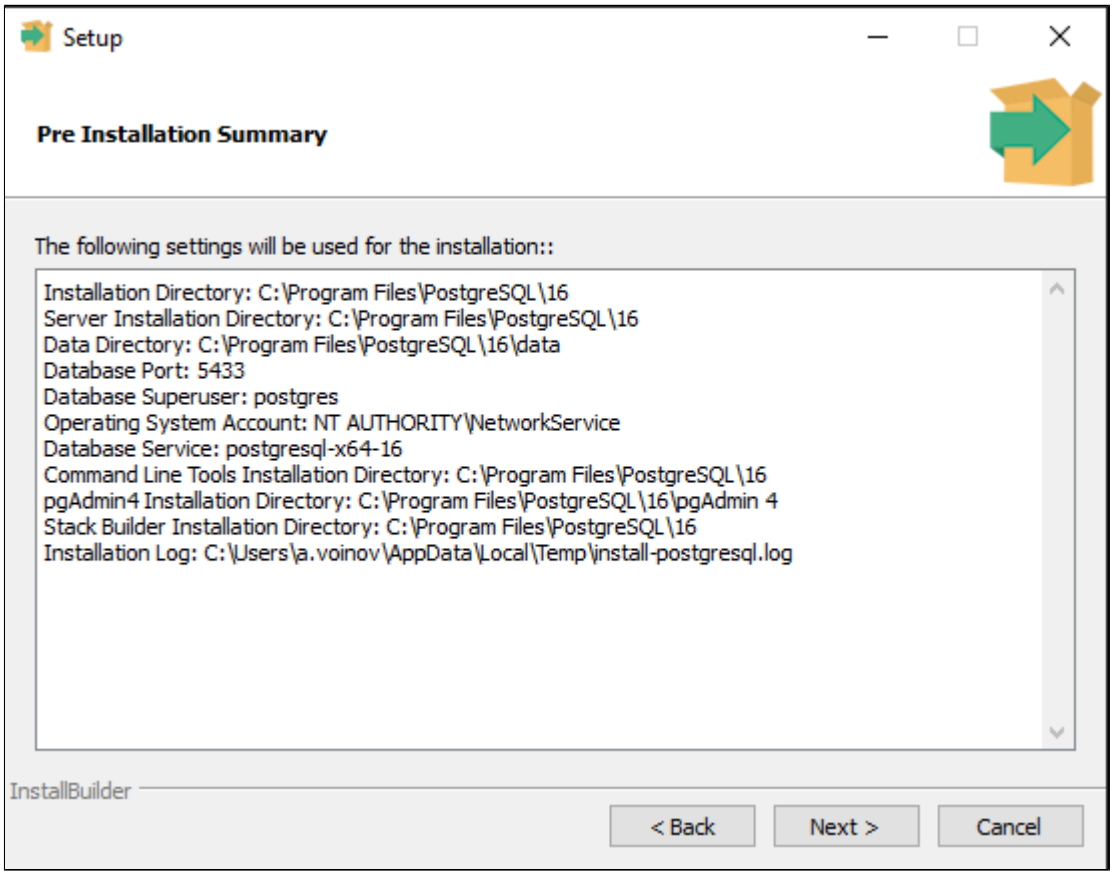


Если на ПК уже был установлен PostgreSQL другой версии, то программа автоматичсеки выставит другой порт для подключения (например 5433).

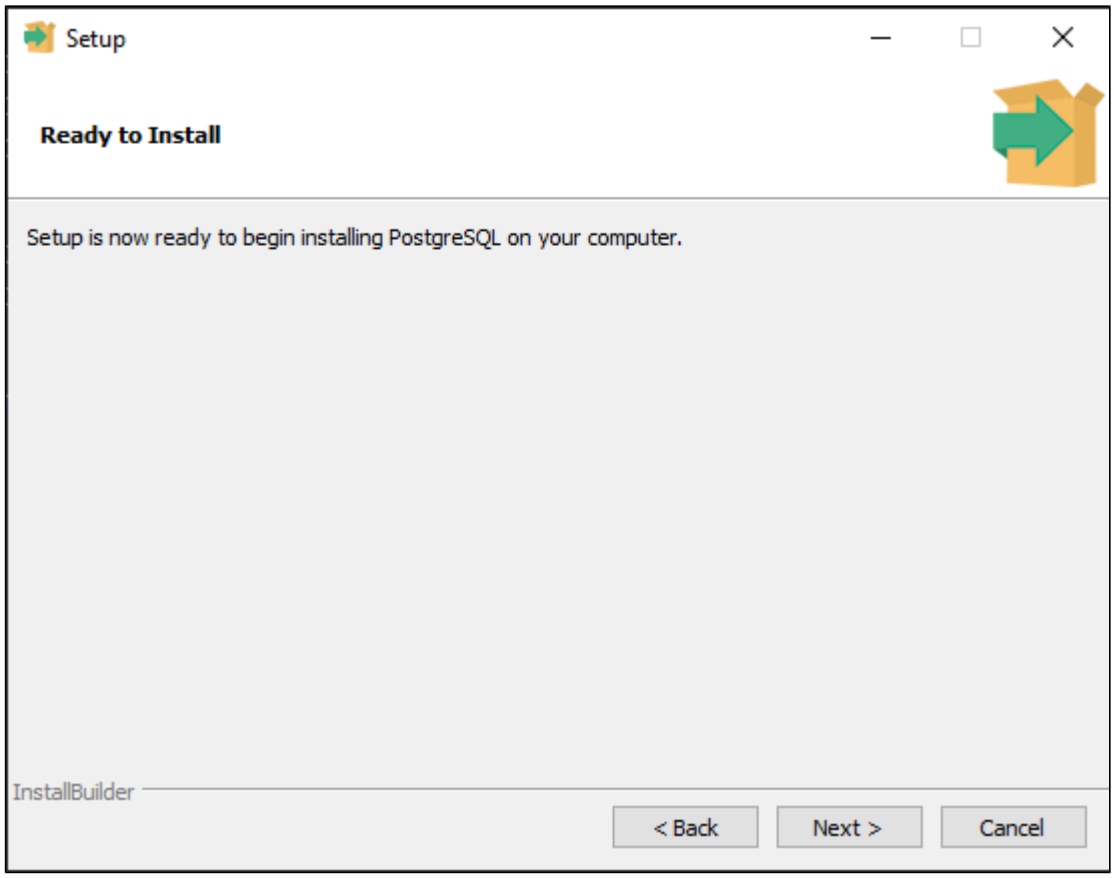
8. Выберите локацию и нажмите кнопку "Next".



9. Проверьте параметры установки и нажмите кнопку "Next".



10. Подтвердите установку, нажав кнопку "Next".



11. После завершения установки снимите флаг "Launch Stack Builder" и нажмите кнопку "Finish".

Подробная установка СУБД описана в инструкции «Alpha.Reports. Руководство администратора» пункт 3.

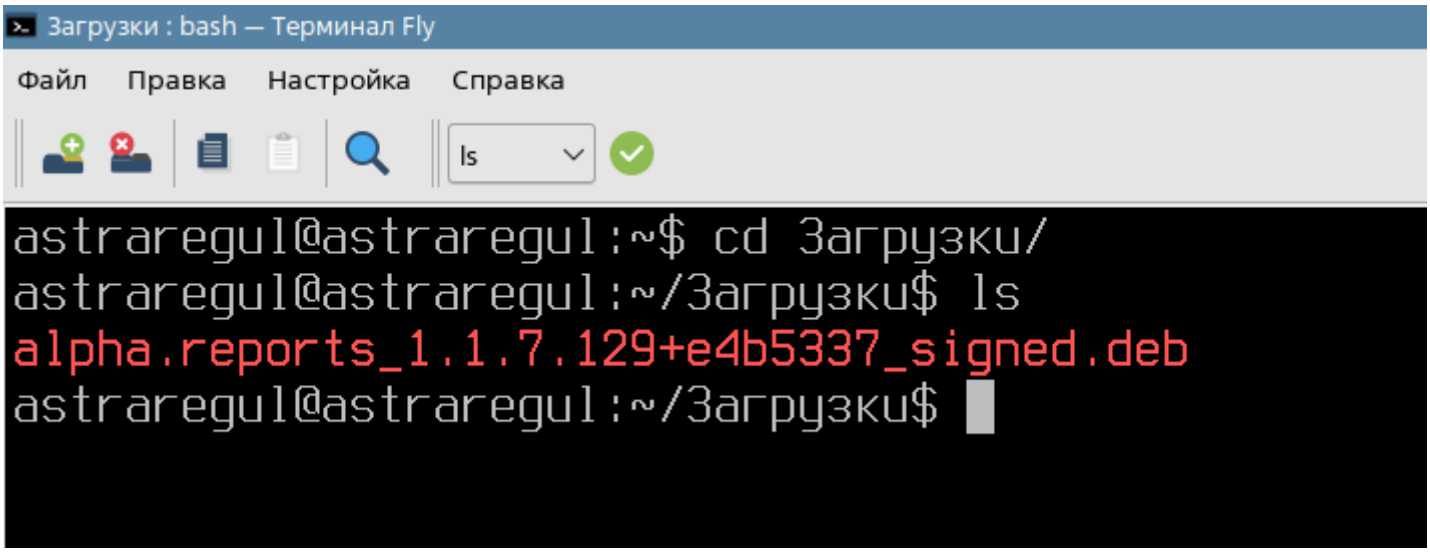
1.1.1.2. AstraLinux

Alpha.Reports
PostgreSQL

1.1.1.2.1. Alpha.Reports

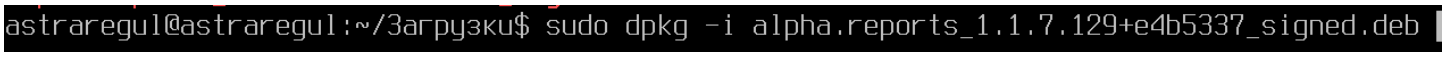
Чтобы установить программный компонент Alpha.Reports, выполните следующие действия:

1. Откройте терминал и перейдите в папку с пакетом установки в формате *.deb.

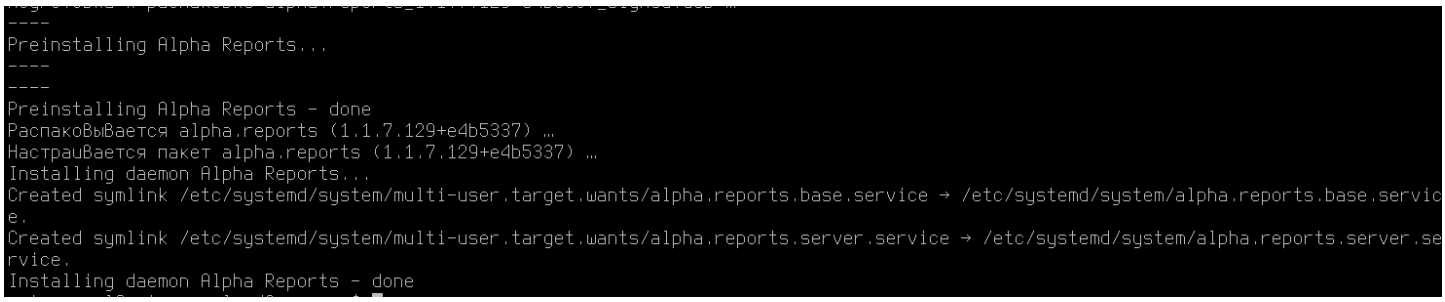


2. Выполните команду:

```
sudo dpkg -i alpha.reports_X.X.X.XXX+XXXXXXX_signed.deb
```



Дождитесь окончания установки:



Каталог установки:

/opt/Automiq/Alpha.Reports.Base
/opt/Automiq/Alpha.Reports.Server

1.1.1.2.2. PostgreSQL

Установку PostgreSQL производить с остановленными службами Alpha.Reports.Base и Alpha.Reports.Server.

1. Откройте терминал и выолните компанду обновления репозитория:

sudo apt update

2. Установите пакет postgresql-11 или выше, выполнив команду:

sudo apt install postgresql-1X

3. Установите дополнительные пакеты, выполнив команду:

sudo apt install libgdiplus
sudo apt install libc6-dev

1.1.1.3. РЕД ОС 7.3

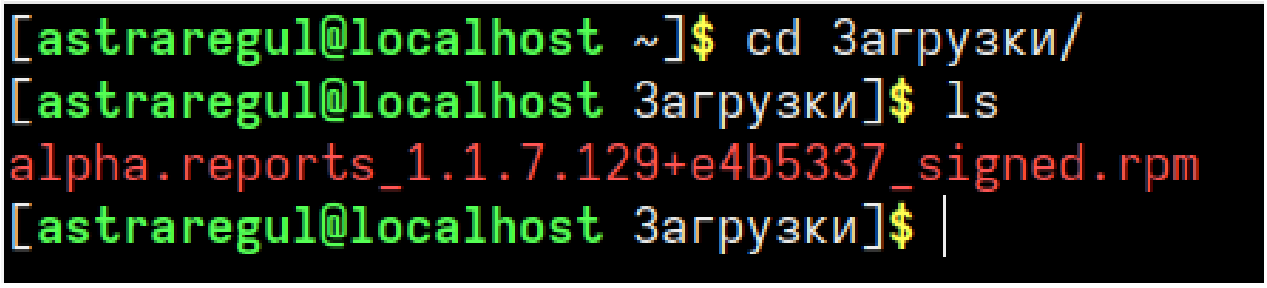
Alpha.Reports
PostgreSQL

Совместимость с РЕД ОС 8 на текущий момент не подтверждена.

1.1.1.3.1. Alpha.Reports

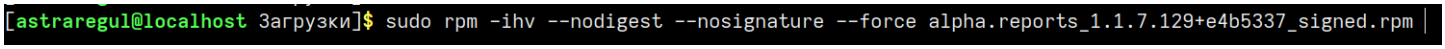
Чтобы установить программный компонент Alpha.Reports, выполните следующие действия:

1. Откройте терминал и перейдите в папку с пакетом установки в формате *.rpm.



2. Выполните команду:

sudo rpm -ihv --nodigest --nosignature --force alpha.reports_X.X.X.XXX+XXXXXXX_signed.rpm



Дождитесь окончания установки:

Каталог установки:

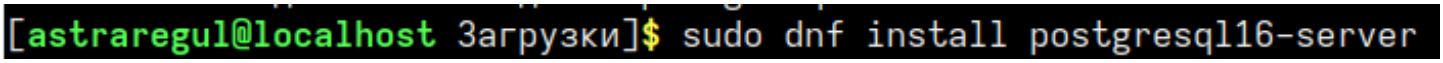
/opt/Automiq/Alpha.Reports.Base
/opt/Automiq/Alpha.Reports.Server

1.1.1.3.2. PostgreSQL

Установку PostgreSQL производить с остановленными службами Alpha.Reports.Base и Alpha.Reports.Server.

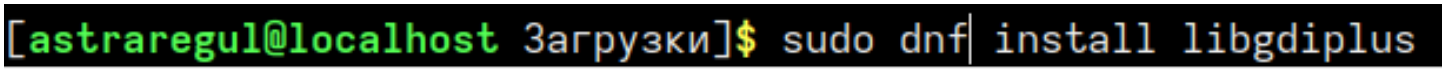
1. Установите Postgresql версии 11 или выше, выполнив команду:

sudo dnf install postgresql1X-server



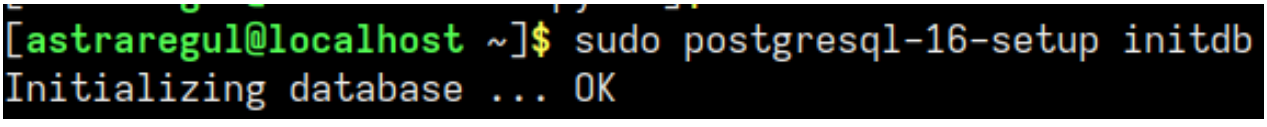
3. Установите дополнительный пакет, выполнив команду:

sudo apt install libgdiplus



4. Далее необходимо произвести инициализацию базы данных postgresql:

sudo postgresql-1X-setup initdb



5. После успешной инициализации запустите службу postgresql и добавьте ее в автозагрузку:

sudo systemctl enable postgresql-1X.service --now

```
[astraregul@localhost ~]$ systemctl enable postgresql-16.service --now
```

6. Установите дополнительный пакет для Postgresql:

```
sudo dnf install postgresql1X-contrib
```

```
[astraregul@localhost ~]$ sudo dnf install postgresql16-contrib
```

1.1.2. Настройка

Windows
AstraLinux
РЕД ОС 7.3

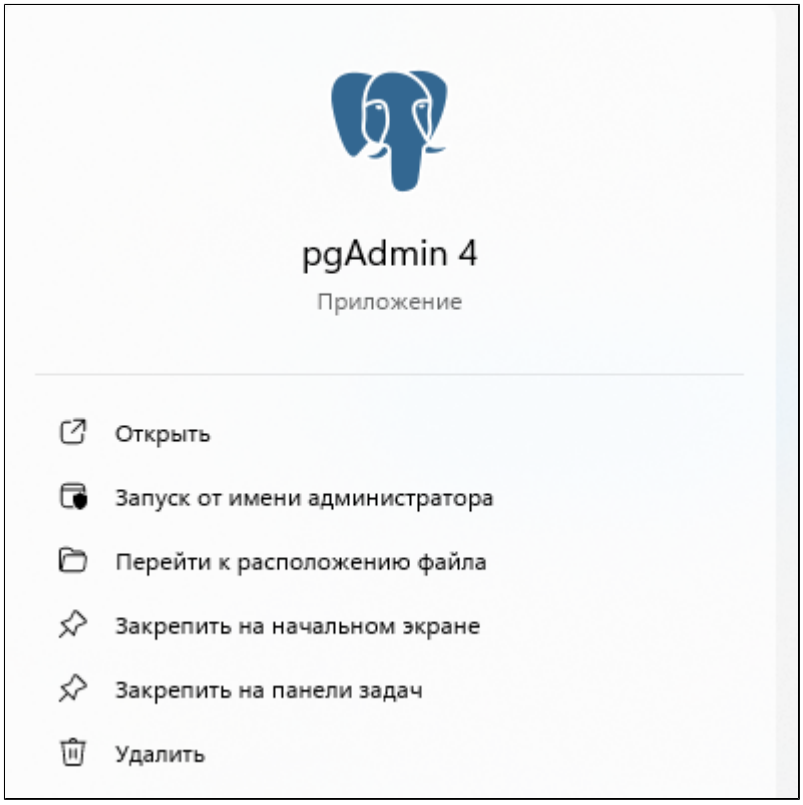
1.1.2.1. Windows

Настройка СУБД
Настройка Alpha.Reports.Base
Настройка Alpha.Reports.Server
Секретная строка
Источники данных

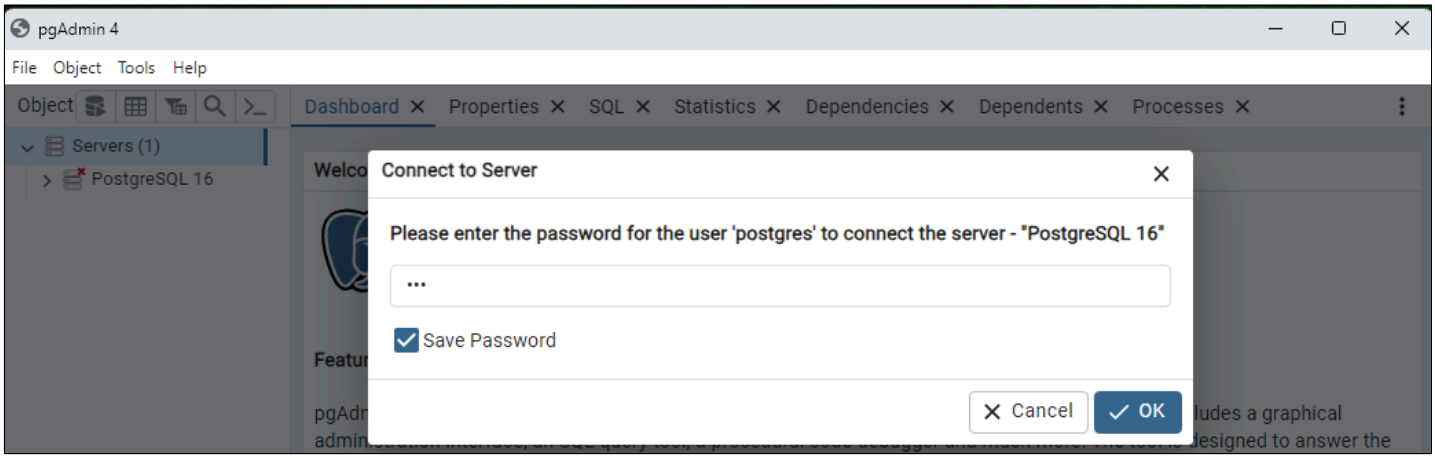
1.1.2.1.1. Настройка СУБД

Для настройки СУБД выполните следующие действия:

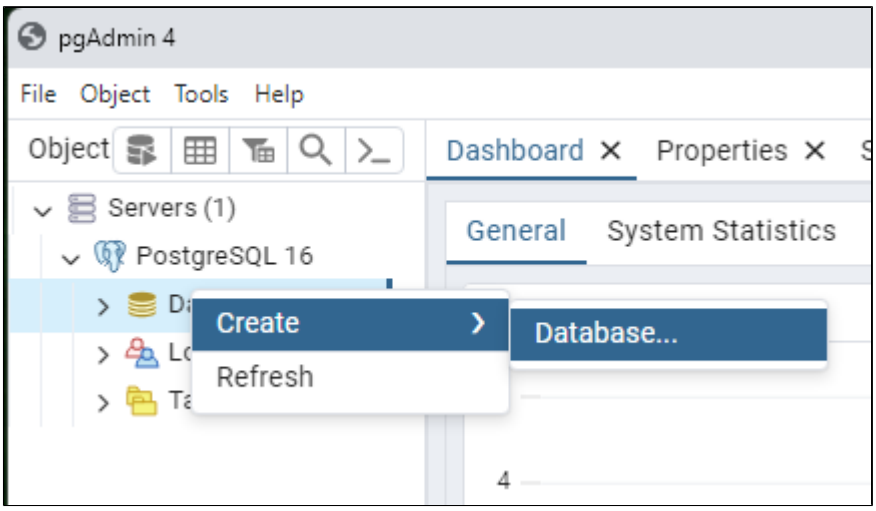
1. Запустите в "pgAdmin4".



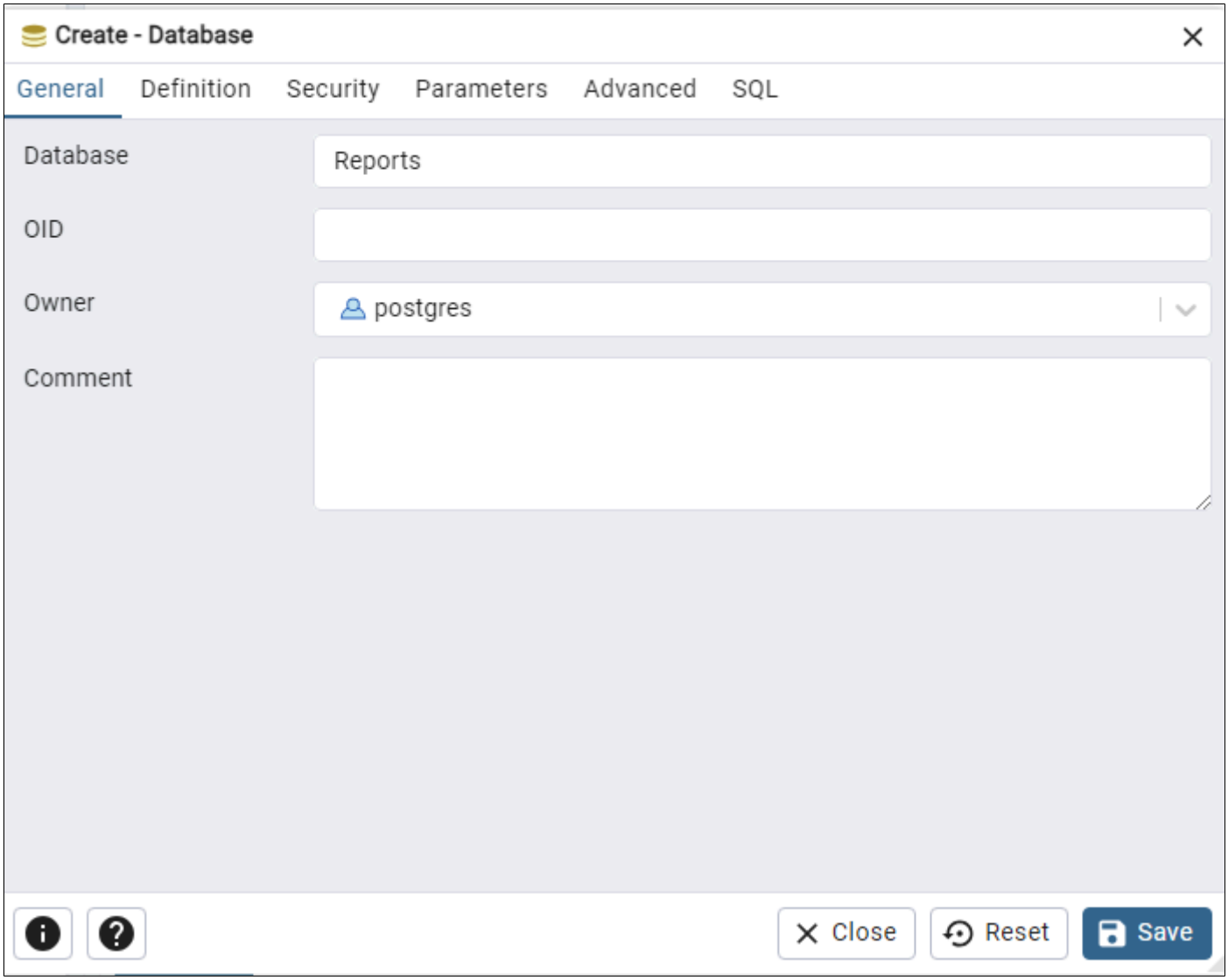
2. Раскройте узел "Servers" и введите пароль, заданный при установки PostgreSQL и нажмите кнопку "ОК".



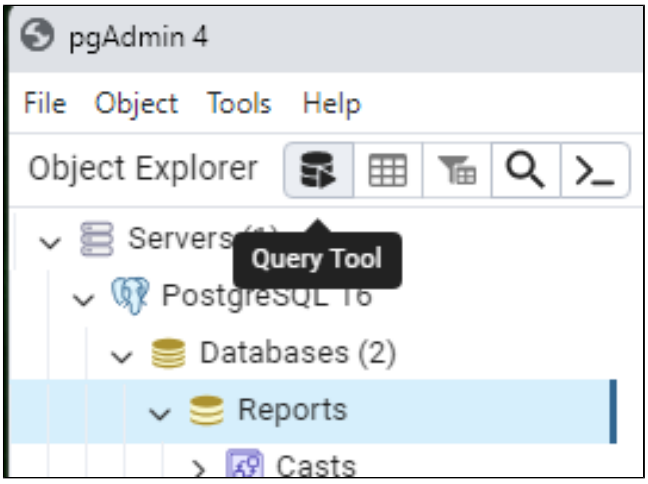
3. Раскройте узел "PostgreSQL 16". В контекстном меню узла "Databases" выберите команду "Create" -> "Database...".



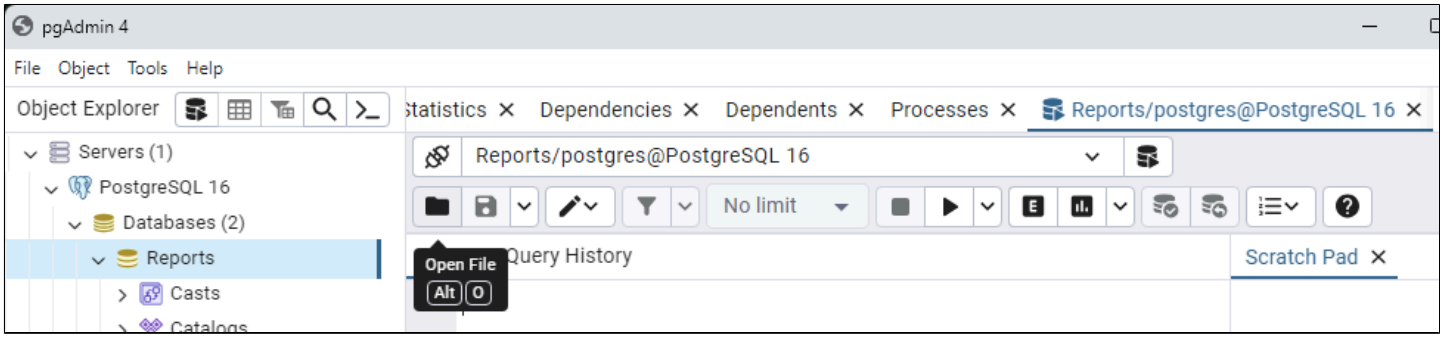
4. В открывшемся диалоговом окне укажите название БД - "Reports" и нажмите кнопку "Save".



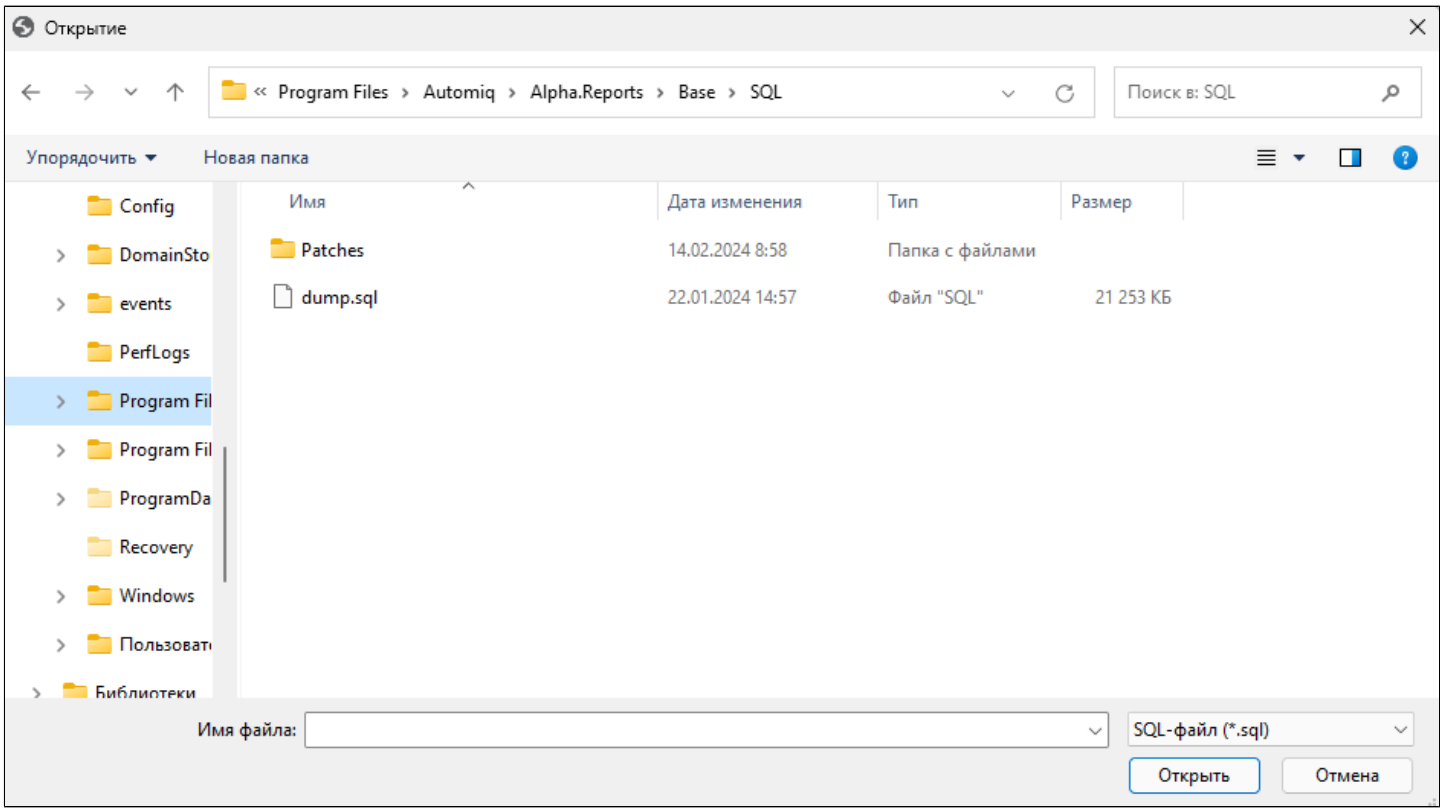
5. Выберите БД "Reports" и в панели инструментов нажмите кнопку "Query Tool". Откроется окно редактора скрипта.



6. На панели инструментов редактора нажмите кнопку "Open File".



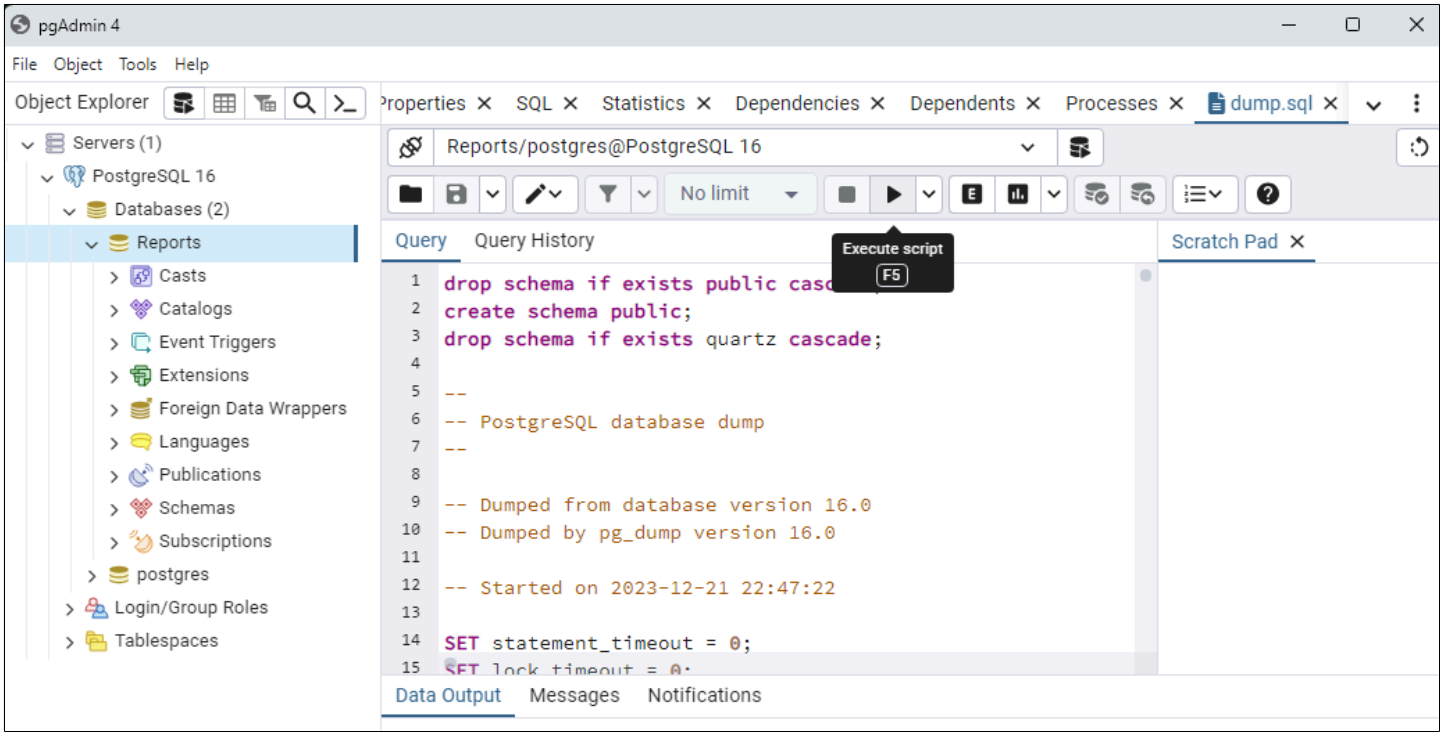
7. Выберите файл "dump.sql" и нажмите кнопку "Открыть".



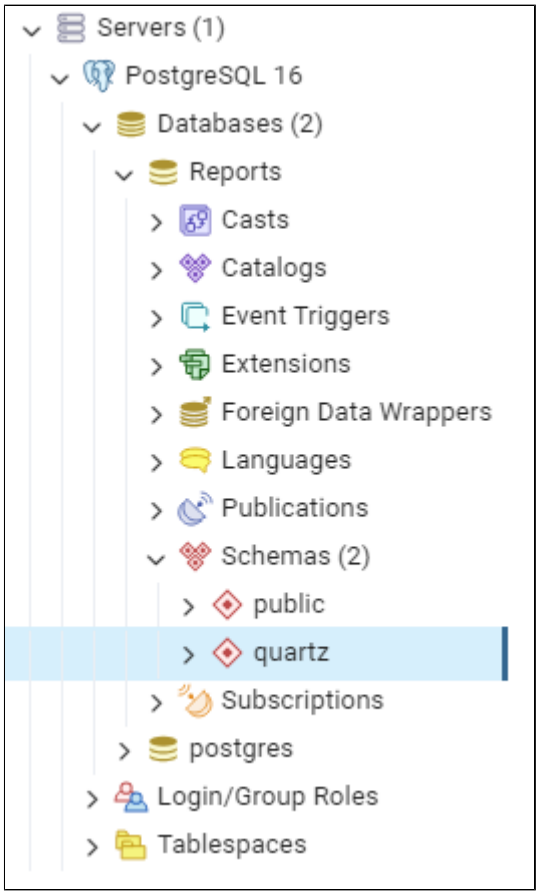
Путь до файла скрипта:

C:\Program Files\Automiq\Alpha.Reports\Base\SQL

8. На панели инструментов редактора скрипта нажмите кнопку "Execute script".



Проверьте, что скрипт успешно выполнился: должна появится схема "quartz".



Настройка аутентификации клиентов

Аутентификация клиентов управляется конфигурационным файлом "pg_hba.conf", который расположен в каталоге с данными кластера базы данных. Файл "pg_hba.conf" со стандартным содержимым, создаётся командой "initdb" при инициализации каталога с данными.

Обычный формат файла "pg_hba.conf" представляет собой набор записей, по одной в строке. Пустые строки игнорируются, как и любой текст комментария после знака #. Записи не продолжаются на следующей строке. Записи состоят из некоторого количества полей, разделённых между собой пробелами и/или отступами (TAB). В полях могут быть использованы пробелы, если они взяты в кавычки. Если в кавычки берётся какое-либо зарезервированное слово в поле базы данных, пользователя или адресации (например, all или replication), то слово теряет своё особое значение и просто обозначает базу данных, пользователя или сервер с данным именем.

Каждая зfпись обозначает тип соединения, диапазон IP-адресов клиента (если он соотносится с типом соединения), имя базы данных, имя пользователя, и способ аутентификации, который будет использован для соединения в соответствии с этими параметрами. Первая запись с соответствующим типом соединения, адресом клиента, указанной базой данных и именем пользователя применяется для аутентификации. Процедур «fall-through» или «backup» не предусмотрено: если выбрана запись и аутентификация не прошла, последующие записи не рассматриваются. Если же ни одна из записей не подошла, в доступе будет отказано.

Запись может быть сделана в одном из семи форматов:

```
local      база    пользователь    метод-аутентификации [параметры]
host       база    пользователь    адрес    метод-аутентификации
[параметры]
hostssl    база    пользователь    адрес    метод-аутентификации
[параметры]
hostnossl  база    пользователь    адрес    метод-аутентификации
[параметры]
```

```
host база пользователь IP-адрес IP-маска метод-аутентификации [параметры]
hostssl база пользователь IP-адрес IP-маска метод-аутентификации [параметры]
hostnossl база пользователь IP-адрес IP-маска метод-аутентификации [параметры]
```

Описания полей

Поле	Описание
local	Управляет подключениями через Unix-сокеты. Без подобной записи подключение через Unix-сокеты невозможно.
host	Управляет подключениями, устанавливаемыми по TCP/IP. Записи host соответствуют подключениям с SSL и без SSL. Удалённое соединение по TCP/IP невозможно, если сервер запущен без определения соответствующих значений для параметра конфигурации "listen_addresses".
hostssl	Управляет подключениями, устанавливаемыми по TCP/IP с применением шифрования SSL. Чтобы использовать эту возможность: › сервер должен быть собран с поддержкой SSL. › механизм SSL должен быть включён параметром конфигурации ssl.
hostnossl	Этот тип записей противоположен hostssl, ему соответствуют только подключения по TCP/IP без шифрования SSL.
база	Определяет, каким именам баз данных соответствует запись. Возможные значения: › «all» – подходят все базы данных. › «sameuser» – данная запись соответствует только, если имя запрашиваемой базы данных совпадает с именем запрашиваемого пользователя. › «samerole» (или устар. «samegroup») – запрашиваемый пользователь должен быть членом роли с таким же именем, как и у запрашиваемой базы данных.

	Суперпользователи не становятся членами роли автоматически, а только если они являются явными членами роли. › «replication» – запись соответствует, если запрашивается подключение для физической репликации. Для таких подключений не выбирается какая-то конкретная база данных. › Любое другое значение воспринимается как имя определённой базы данных. Несколько имён баз данных можно указать, разделяя их запятыми. Файл, содержащий имена баз данных, можно указать, поставив знак «@» в начале его имени
пользователь	Указывает, какому имени (или именам) пользователя базы данных соответствует запись. Возможные значения: › «all» – запись соответствует всем пользователям. › Любое другое значение задаёт либо имя конкретного пользователя базы данных, либо имя группы (если это значение начинается с +). Несколько имён пользователей можно указать, разделяя их запятыми. Файл, содержащий имена пользователей, можно указать, поставив знак «@» в начале его имени.
адрес	Указывает адрес (или адреса) клиентского компьютера, которому соответствует запись. Поле может содержать или имя компьютера, или диапазон IP-адресов, или одно из ключевых слов: › «all» – любой IP-адрес. › «samehost» – любые IP-адреса данного сервера. › «samenet» – любой адрес любой подсети, к которой сервер подключён напрямую.
IP-адрес IP-маска	Эти два поля могут быть использованы как альтернатива записи IP-адрес/длина-маски. Вместо того, чтобы указывать длину маски, в отдельном столбце указывается сама маска. Например, 255.0.0.0 представляет собой маску CIDR для IPv4 длиной 8 бит, а 255.255.255.255 представляет маску CIDR

	длиной 32 бита. Эти поля применимы только к записям host, hostssl и hostnossl
метод-аутентификации	Указывает метод аутентификации, когда подключение соответствует записи. Все значения воспринимаются с учётом регистра и должны быть записаны в нижнем регистре. Возможные значения: › «trust» – разрешает безусловное подключение. Этот метод позволяет тому, кто может подключиться к серверу с базой данных Postgres Pro, войти под любым желаемым пользователем Postgres Pro без введения пароля и без какой-либо другой аутентификации. › «reject» – отклоняет подключение безусловно. Эта возможность полезна для «фильтрации» некоторых серверов группы, например, строка с «reject» может отклонить попытку подключения одного компьютера, при этом следующая строка позволяет подключиться остальным компьютерам в той же сети. › «scram-sha-256» – проверяет пароль пользователя, производя аутентификацию SCRAM-SHA-256. › «md5» – проверяет пароль пользователя, производя аутентификацию SCRAM-SHA-256 или MD5. › «password» – требует для аутентификации введения клиентом незашифрованного пароля. Поскольку пароль посылается простым текстом через сеть, такой способ не стоит использовать, если сеть не вызывает доверия. › «gss» – для аутентификации пользователя использует GSSAPI. Этот способ доступен только для подключений по TCP/IP. › «sspi» – для аутентификации пользователя использует SSPI. Способ доступен только для Windows. › «ident» – получает имя пользователя операционной системы клиента, связываясь с сервером Ident, и проверяет, соответствует ли оно имени пользователя базы данных. Аутентификация «ident» может использоваться только для подключений по TCP/IP. Для

	локальных подключений применяется аутентификация «peer». › «peer» – получает имя пользователя операционной системы клиента из операционной системы и проверяет, соответствует ли оно имени пользователя запрашиваемой базы данных. Доступно только для локальных подключений. › «ldap» – проводит аутентификацию, используя сервер LDAP. › «radius» – проводит аутентификацию, используя сервер RADIUS. › «cert» – проводит аутентификацию, используя клиентский сертификат SSL. › «pam» – проводит аутентификацию, используя службу подключаемых модулей аутентификации (PAM), предоставляемую операционной системой. › «bsd» – проводит аутентификацию, используя службу аутентификации BSD, предоставляемую операционной системой.
параметры	Поле (поля) вида имя=значение, определяющее параметры метода аутентификации. Подробнее о параметрах, доступных для различных методов аутентификации, рассказывается ниже. Помимо описанных далее параметров, относящихся к различным методам, есть один общий параметр аутентификации clientcert, который можно задать в любой записи hostssl. Если он равен 1, клиент должен представить подходящий (доверенный) сертификат SSL, в дополнение к другим требованиям метода проверки подлинности.

Файлы, включённые в конструкции, начинающиеся с «@», читаются, как список имён, разделённых запятыми или пробелами. Комментарии предваряются знаком «#», как и в файле "pg_hba.conf",

и вложенные «@» конструкции допустимы. Если только имя файла, начинающегося с «@» не является абсолютным путём.

Поскольку записи файла "pg_hba.conf" рассматриваются последовательно для каждого подключения, порядок записей имеет большое значение. Обычно более ранние записи определяют чёткие критерии для соответствия параметров подключения, но для методов аутентификации допускают послабления. Напротив, записи более поздние смягчают требования к соответствию параметров подключения, но усиливают их в отношении методов аутентификации.

Например, некто желает использовать trust аутентификацию для локального подключения по TCP/IP, но при этом запрашивать пароль для удалённых подключений по TCP/IP. В этом случае запись, устанавливающая аутентификацию trust для подключения адреса 127.0.0.1, должна предшествовать записи, определяющей аутентификацию по паролю для более широкого диапазона клиентских IP-адресов.

Файл "pg_hba.conf" прочитывается при запуске системы, а также в тот момент, когда основной сервер (процесс "postmaster") получает сигнал "SIGHUP". Послать сигнал процессу можно любым из способов:

- используя pg_ctl reload
- вызвав SQL-функцию pg_reload_conf()
- выполнив kill -HUP

Предыдущее утверждение не касается Microsoft Windows: там любые изменения в "pg_hba.conf" сразу применяются к последующим подключениям.

Чтобы подключиться к конкретной базе данных, пользователь не только должен пройти все проверки файла "pg_hba.conf", но должен иметь привилегию "CONNECT" для подключения к базе данных. Если вы хотите ограничить доступ к базам данных для определённых пользователей,

проще предоставить/отозвать привилегию "CONNECT", нежели устанавливать правила в записях файла "pg_ hba.conf".

1.1.2.1.1.1. Примеры настройки конфигурационного файла

Позволяет любому пользователю локальной системы подключаться к любой базе данных, используя любое имя пользователя баз данных, через Unix-сокеты (по умолчанию для локальных подключений).

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
local       all             all                         trust
```

То же, но для локальных замкнутых подключений по TCP/IP.

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         127.0.0.1/32    trust
```

То же, что и в предыдущем примере, но с указанием сетевой маски в отдельном столбце

```
# TYPE    DATABASE    USER    ADDRESS    IP-MASK           METHOD
host      all         all     127.0.0.1  255.255.255.255   trust
```

То же для IPv6.

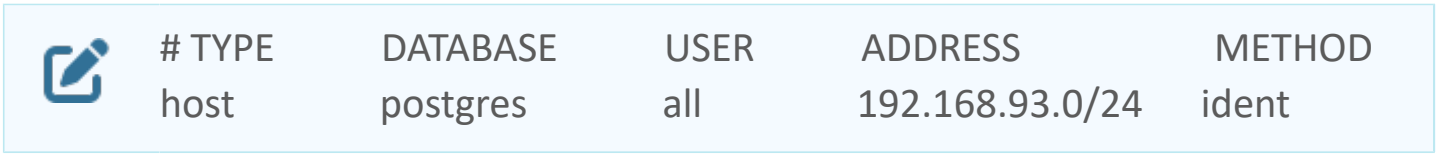
```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         ::1/128         trust
```

То же самое, но с использованием имени компьютера (обычно покрывает и IPv4, и IPv6).

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         localhost       trust
```

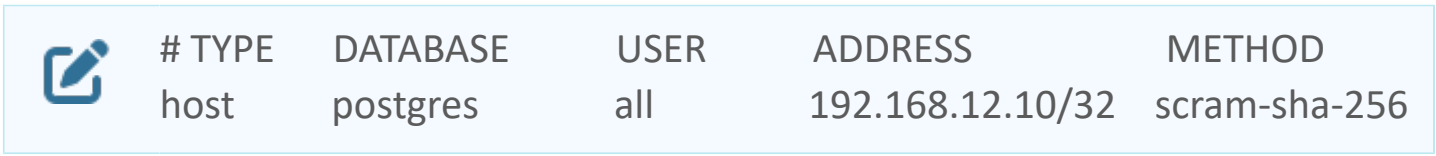
Позволяет любому пользователю любого компьютера с IP-адресом 192.168.93.x подключаться к базе данных "postgres" с именем, которое

сообщает для данного подключения ident (как правило, имя пользователя операционной системы).



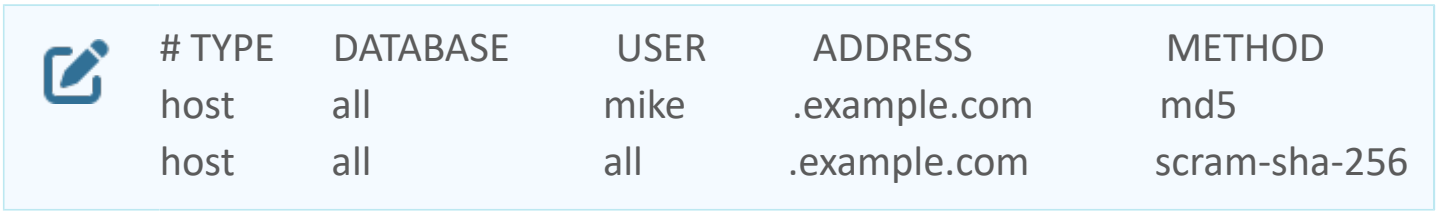
# TYPE	DATABASE	USER	ADDRESS	METHOD
host	postgres	all	192.168.93.0/24	ident

Позволяет любому пользователю компьютера 192.168.12.10 подключаться к базе данных "postgres", если он передаёт правильный пароль.



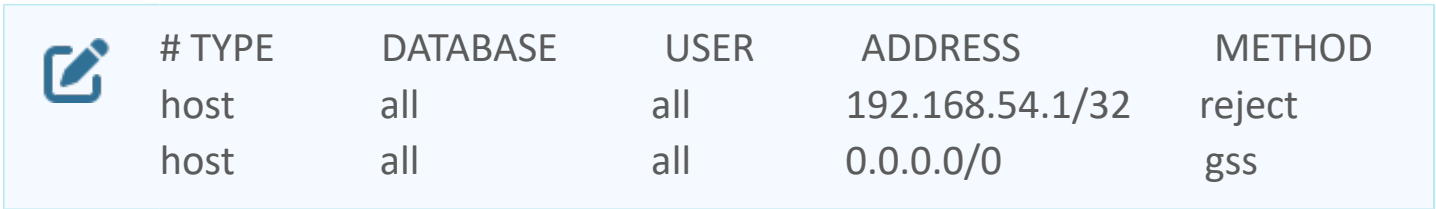
# TYPE	DATABASE	USER	ADDRESS	METHOD
host	postgres	all	192.168.12.10/32	scram-sha-256

Позволяет любым пользователям с компьютеров в домене example.com подключаться к любой базе данных, если передаётся правильный пароль. Для всех пользователей требуется аутентификация SCRAM, за исключением пользователя 'mike', который использует старый клиент, не поддерживающий аутентификацию SCRAM.



# TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	mike	.example.com	md5
host	all	all	.example.com	scram-sha-256

В случае отсутствия предшествующих строчек с "host", следующие две строки откажут в подключении с 192.168.54.1 (поскольку данная запись будет выбрана первой), но разрешат подключения GSSAPI с любых других адресов. С нулевой маской ни один бит из IP-адреса компьютера не учитывается, так что этой строке соответствует любой компьютер.



# TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	all	192.168.54.1/32	reject
host	all	all	0.0.0.0/0	gss

Позволяет пользователям с любого компьютера 192.168.x.x подключаться к любой базе данных, если они проходят проверку ident. Если же ident говорит, например, что это пользователь "bryanh" и он запрашивает подключение как пользователь Postgres Pro "guest1", подключение будет разрешено, если в

файле pg_ident.conf есть сопоставление "omicron", позволяющее пользователю "bryanh" подключаться как "guest1".

```
# TYPE    DATABASE        USER    ADDRESS          METHOD
host      all             all     192.168.0.0/16   ident map=omicron
```

Если для локальных подключений предусмотрены только эти три строки, они позволят локальным пользователям подключаться только к своим базам данных (базам данных с именами, совпадающими с именами пользователей баз данных), кроме администраторов или членов роли "support", которые могут подключиться к любой БД. Список имён администраторов содержится в файле $PGDATA/admins. Пароли запрашиваются в любом случае.

```
# TYPE    DATABASE        USER        ADDRESS          METHOD
local     sameuser        all                          md5
local     all             @admins                      md5
local     all             +support                     md5
```

Последние две строчки в примере выше могут быть объединены в одну:

```
# TYPE    DATABASE        USER                ADDRESS          METHOD
local     all             @admins,+support                     md5
```

В столбце DATABASE могут указываться списки и имена файлов:

```
# TYPE    DATABASE                USER        ADDRESS          METHOD
local     db1,db2,@demodbs        all                          md5
```

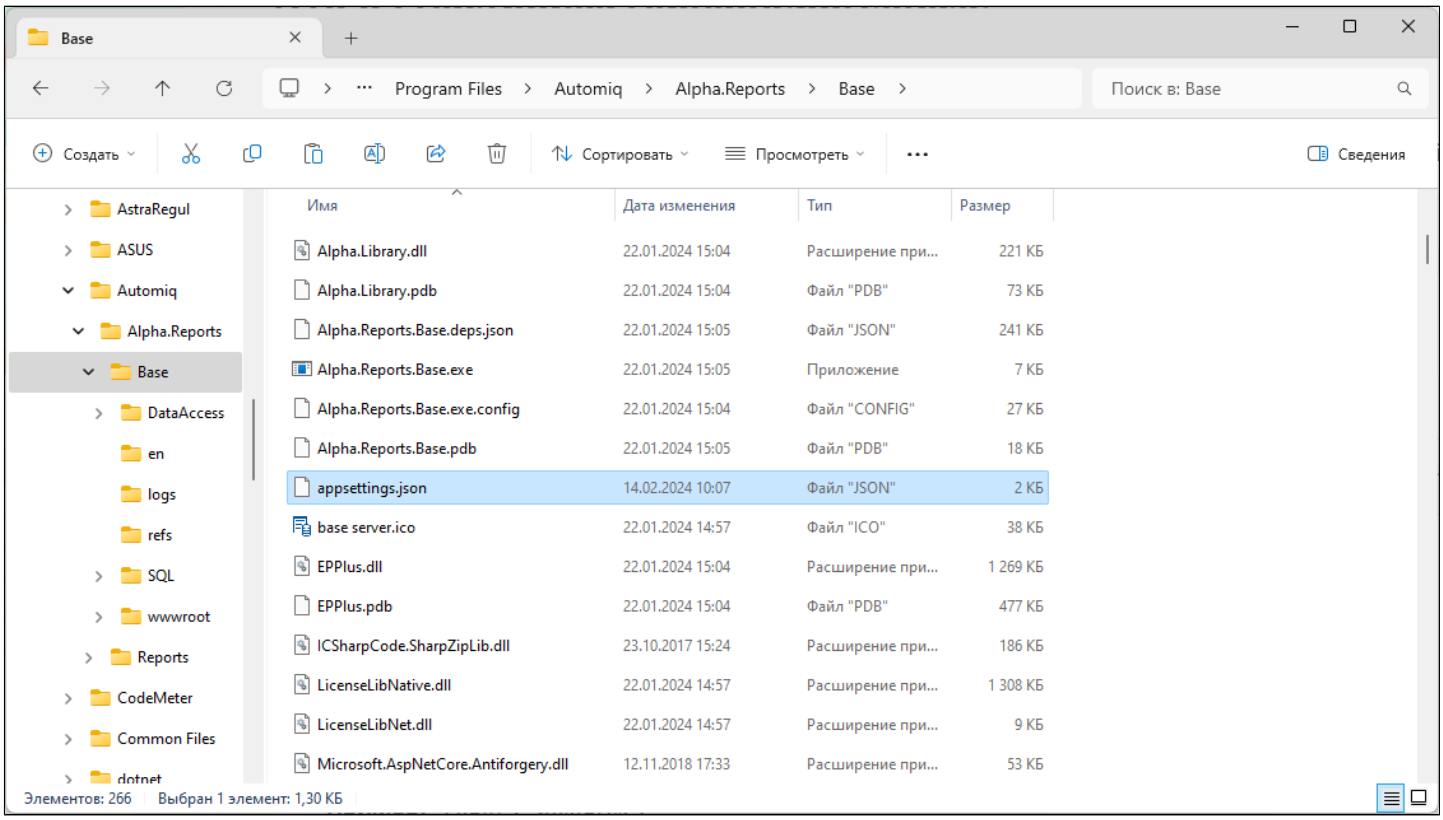
1.1.2.1.2. Настройка Alpha.Reports.Base

Все файлы настроек имеют расширение *.json и находятся рядом с исполняемым файлом в каталоге установки.

Для настройки Alpha.Reports.Base выполните следующие действия:

1. Перейдите в каталог установки Alpha.Reports.Base и откройте с помощью текстового редактора файл "appsettings.json".

C:\Program Files\Automiq\Alpha.Reports\Base



2. Отредактируйте файл конфигурации:

В разделе "DataBaseServer" укажите следующие параметры:

- В пункте "Address" введите IP-адрес базы данных или доменное имя "localhost".

  Пример: "Address": "localhost".

› В пункте "DataBase" введите имя базы данных
Пример: "DataBase": "Reports".
› В пункте "Login" введите логин для базы данных. По умолчанию: "postgres".
Пример: "Login": "postgres".
› В пункте "Password" введите пароль для базы данных, который Вы указали при установки PostgreSQL.
Пример: "Password": "123".

> Пароль не должен быть пустым.

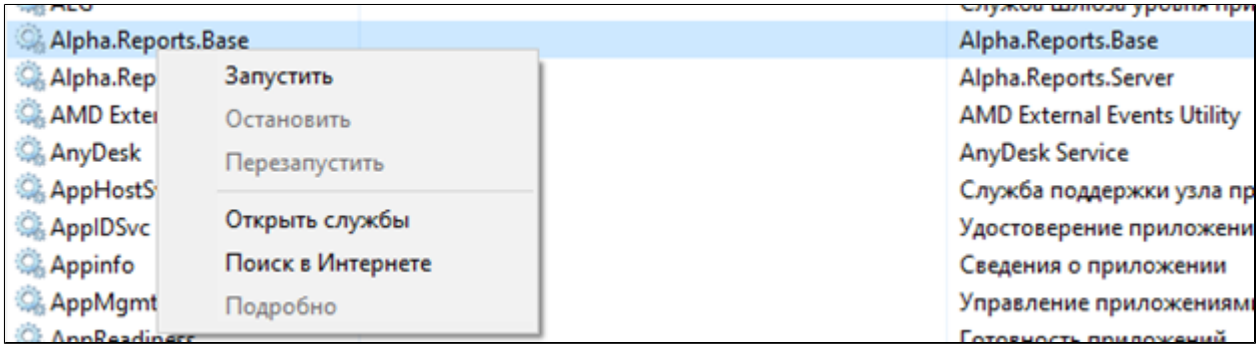
В разделе "LDAPSecurity" укажите следующие параметры:
› В пункте "BaseName" замените "AlphaSecurity" на "AstraSecurity":
Пример: "BaseName": "AstraSecurity".
› В пункте "Host" введите "IP-адрес компьютера, на котором установлен LDAP.
Пример: "Host": "127.0.0.1".
Остальные параметры оставьте по-умолчаню.

В списке "EntryPoints" нужно перечислить все адреса, по которым должен быть доступен базовый сервер, в том числе все сетевые карты компьютера (если их больше одной) – для CORS политики.

> Обращаться к серверу нужно строго по IP-адресу, указанному в поле "Application" → "ServerAddress".
> Значение «localhost» – не равно «127.0.0.1».

> IP-адрес, указанный в поле "Modules" → "Alpha.Reports.Report.Server" должен совпадать с IP-адресом, указанным в поле "Application" → "ServerAddress".

3. Запустите/перезапустите службу "Alpha.Reports.Base".


Alpha.Reports.Base
Alpha.Reports.Base
Alpha.Rep
Alpha.Reports.Server
AMD Exte
AMD External Events Utility
AnyDesk
AnyDesk Service
AppHostS
Служба поддержки узла пр
AppIDSvc
Удостоверение приложени
Appinfo
Сведения о приложении
AppMgmt
Управление приложениям
Запустить
Остановить
Перезапустить
Открыть службы
Поиск в Интернете
Подробно

1.1.2.1.1.2.1. Пример файла конфигурации

```
{
    "Kestrel": {
        "EndPoints": {
            "HTTP": {
                "Url": "http://*:5000"
            }
        }
    },

    "Application": {
        "Name": "Alpha.Reports.Base",
        "ServerAddress": "http://127.0.0.1:5000",
        "LicenseFile": "license.lic",
        "BasePath": "",
        "RedirectToHTTPS": "false",
        "LdapServer": "",

        "DataBaseServer": {
            "Address": "localhost",
            "DataBase": "Reports",
            "Login": "postgres",
            "Password": "123"
        },

        "LDAPSecurity": {
            "Host": "127.0.0.1",
            "Port": "389",
            "BaseName": "AstraSecurity",
            "AppName": "Reports",
            "Rights": {
                "AdminRight": "Administration",
                "ViewRight": "View"
            }
        },

        "Modules": {
```

```
                    "Alpha.Reports.Server": "http://127.0.0.1:5001"
                },

                "AlphaLicenseName": "",

                "DisableDataAccessAutoUpdate": "false",

                "AllowAllNIC": "false",
                "EntryPoints": [
                    "http://127.0.0.1:5000",
                    "http://192.168.0.1:5000",
                    "http://somehostname:5000"
                ],

                "CleanupInterval": 43200
            }
        }
```

1.1.2.1.3. Настройка Alpha.Reports.Server

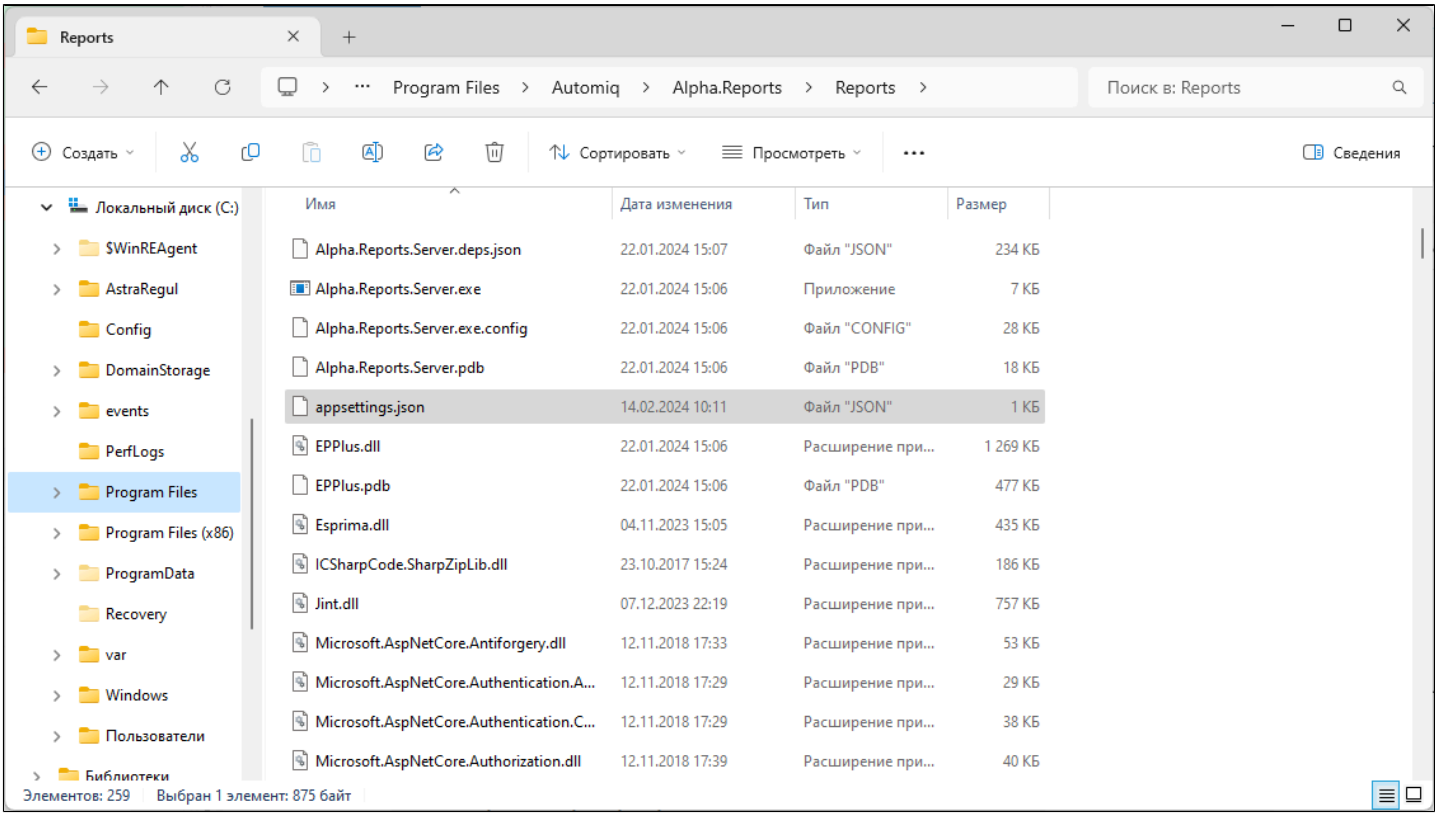
Если при формировании отчётов по расписанию необходимо отправлять их по списку электронных адресов, то требуется сконфигурировать работу сервера отчётов.

Все файлы настроек имеют расширение *.json и находятся рядом с исполняемым файлом в каталоге установки.

Для настройки Alpha.Reports.Server выполните следующие действия:

1. Перейдите в каталог установки Alpha.Reports.Server и откройте с помощью текстового редактора файл "appsettings.json".

C:\Program Files\Automiq\Alpha.Reports\Reports



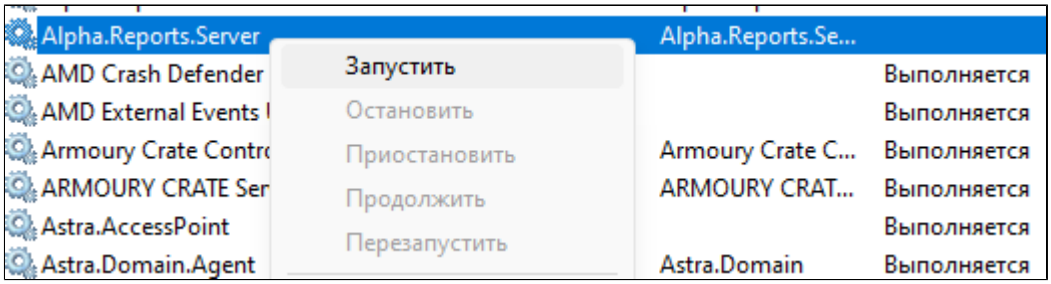
2. Отредактируйте файл конфигурации:

Раздел "DataBaseServer" необходимо заполнить аналогично Alpha.Reports.Base.

В разделе "SMTP" заполните параметры:

- Address – адрес smtp-сервера.
- Port – порт для подключения, по умолчанию 25.
- Login – почтовый адрес (полный), с которого будет осуществляться рассылка.
- Password – пароль для подключения к smtp-серверу.

3. Запустите/перезапустите службу "Alpha.Reports.Server".



1.1.2.1.3.1. Пример файла appsettings.json

```
{
    "Kestrel": {
        "EndPoints": {
            "HTTP": {
                "Url": "http://*:5001"
            }
        }
    },

    "Application": {
        "Name": "Alpha.Reports.Server",
        "BasePath": "",
        "RedirectToHTTPS": "false",

        "DataBaseServer": {
            "Address": "localhost",
            "DataBase": "Reports",
            "Login": "postgres",
            "Password": "123"
        },

        "Modules": {
            "Alpha.Base.Server": "http://127.0.0.1:5000"
        },

        "SMTP": {
            "Address": "smtp.example.com",
            "Port": "25",
            "Login": "alpha_reports@example.com",
            "Password":  "password"
        },

        "JobFile": "Alpha.Reports.Server.Jobs.xml"
    }
}
```

1.1.2.1.4. Секретная строка

Для работы с системой отчётности необходима секретная строка, которая генерируется при запуске "Alpha.Reports.Base".

Получить секретную строку можно любым из способов:

- Программно обратитесь к функции "GetAuthString()".
- В файле C:\ProgramFiles\Automiq\Alpha.Reports\Base\logs\Base.log найдите строку

<Дата> <Время>|Info|Alpha.Reports.Base|Строка авторизации:<Строка авторизации>

Пример

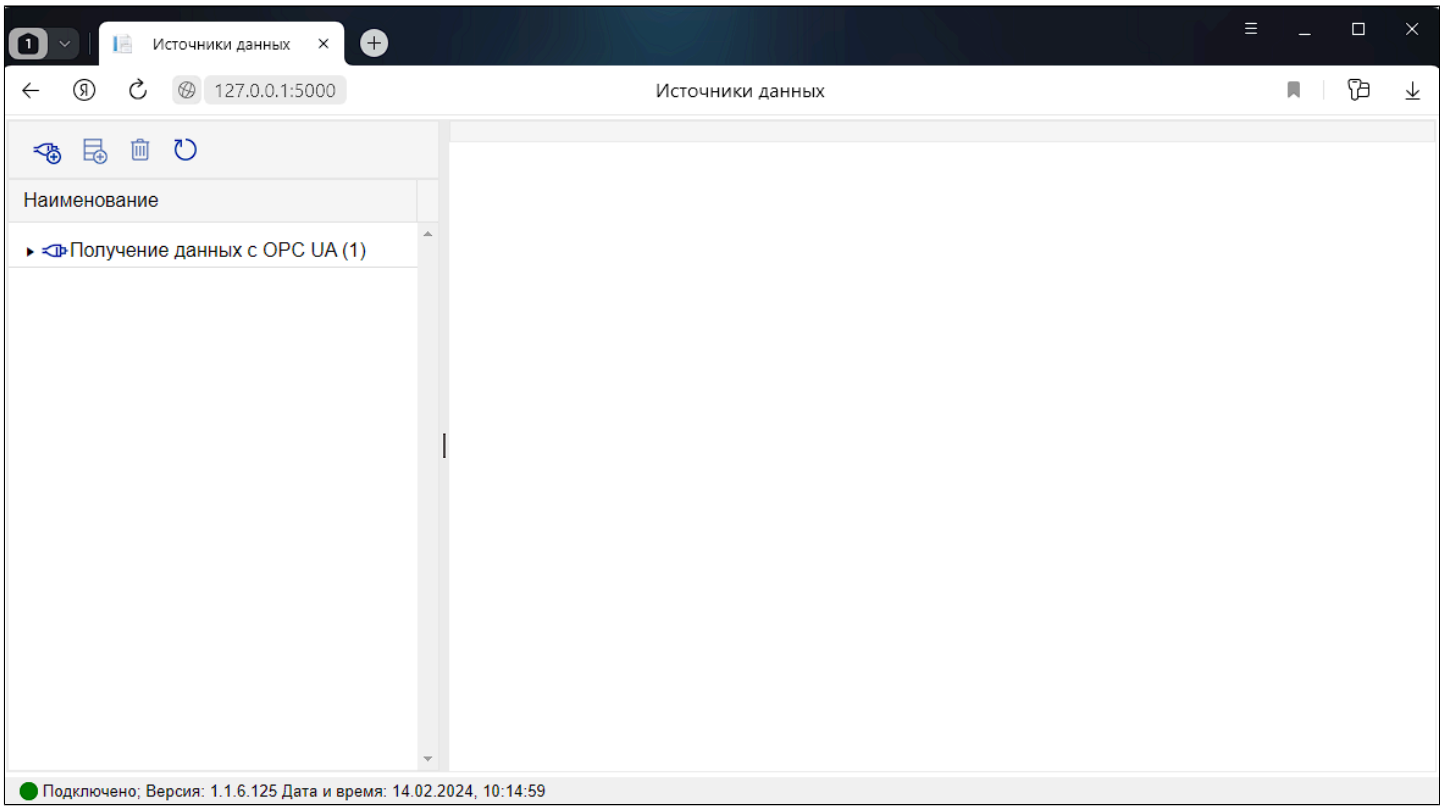
```
2024-03-05 11:01:25.3114|Warn|Alpha.Reports.Base|В системе отсутствуют компоненты лицензирования Alpha, р
2024-03-05 11:01:25.5432|Info|Alpha.Reports.Base|Подключение к базе данных PostgreSQL 16.2, compiled by V
2024-03-05 11:01:25.6120|Info|Alpha.Reports.Base|Строка авторизации: eyJod2lkIjoiRjJDNC1DQzk2LUEzRTEtM0NE
2024-03-05 11:01:25.6120|Info|Alpha.Reports.Base|Загрузка источников данных началась
2024-03-05 11:01:25.6120|Debug|Alpha.Reports.Base|Статус отключения автообновления файлов коннекторов с д
2024-03-05 11:01:25.7875|Debug|Alpha.Reports.Base|Файл Alpha.DataAccess.OPC.UA.dll коннектора «Получение
2024-03-05 11:01:25.8547|Debug|Alpha.Reports.Base|Файл Alpha.DataAccess.OPC.UA.Views.dll коннектора «Полу
```

1.1.2.1.5. Источники данных

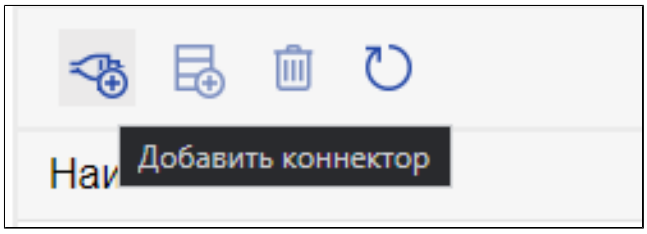
1. Чтобы открыть панель конфигурирования источников, в браузере введите:

http://127.0.0.1:5000/Sources/?u=<Секретная строка>

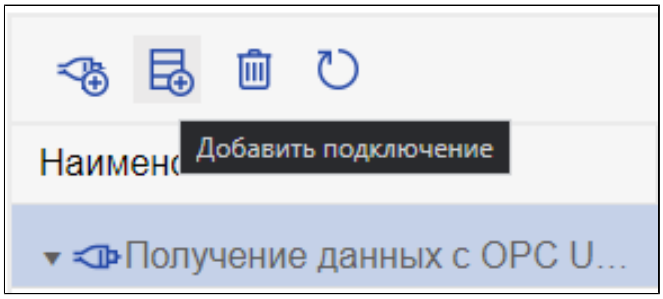
При успешном запуске в браузере отобразится окно:



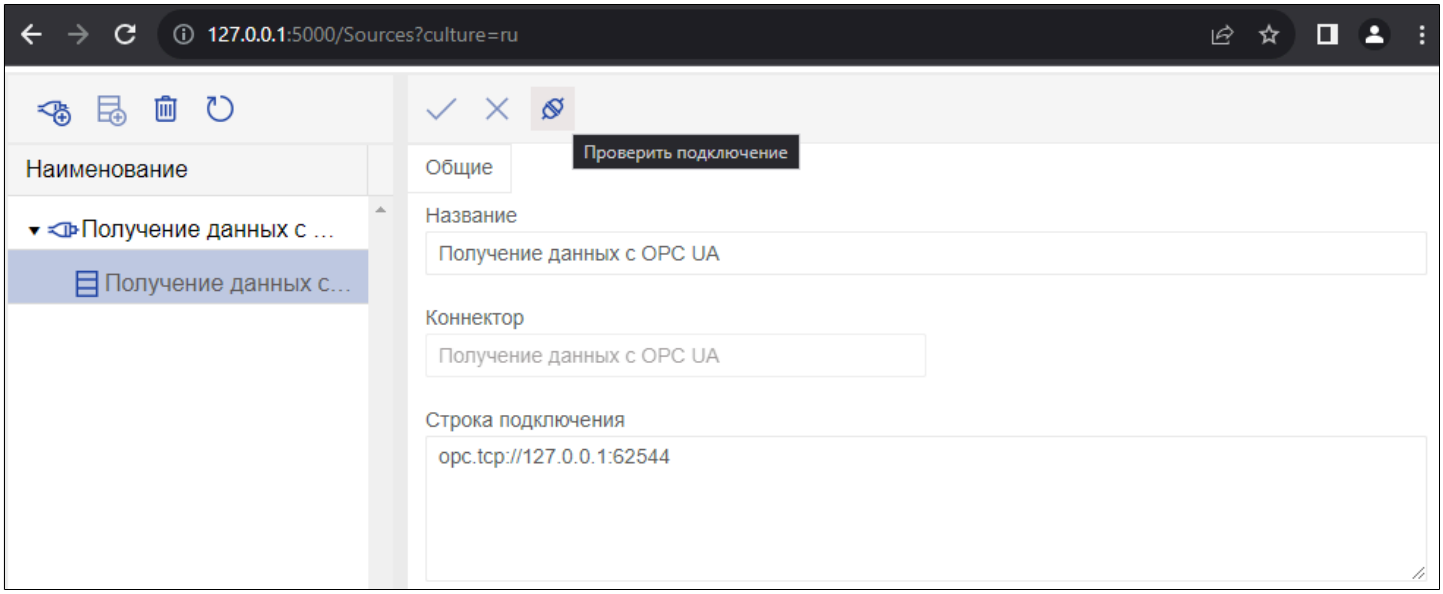
2. По умолчанию источники данных уже настроены и можно переходить к пункту 4. Если источники данных не настроены, нажмите кнопку "Добавить коннектор".



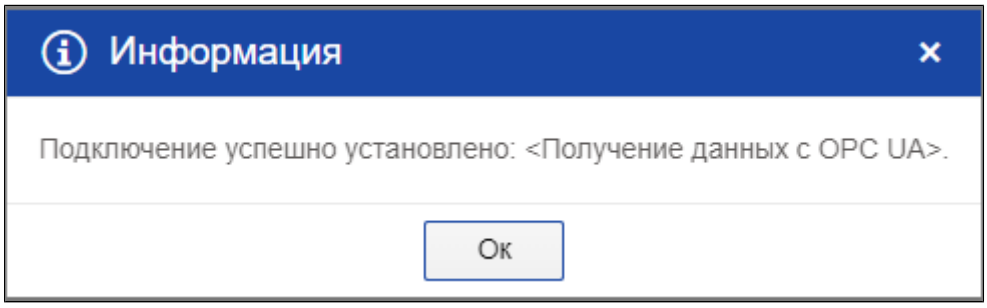
3. Выделите коннектор и нажмите кнопку «Добавить подключение».



4. Настройте строку подключения у коннектора и проверьте подключение.



При успешном подключении появится следующее окно:



Подробнее вкладка «Источники данных» описана в п.6 документа «Alpha.Reports. Руководство администратора».

1.1.2.2. AstraLinux

Настройка СУБД
Настройка Alpha.Reports.Base
Настройка Alpha.Reports.Server
Секретная строка
Источники данных

1.1.2.2.1. Настройка СУБД

Для настройки СУБД выполните следующие действия:

1. Откройте файл "postgresql.conf", выполнив команду:

```
sudo nano /etc/postgresql/1X/main/postgresql.conf
```

```
astraregul@astraregul:~$ sudo nano /etc/postgresql/14/main/postgresql.conf
```

2. Отредактируйте строку "port = 5433" на "port = 5432":

```
port = 5432
```

Для сохранения и выхода нажмите комбинацию клавиш "Ctrl+x". Для подтверждения изменений нажмите клавишу "y" и нажмите клавишу "Enter".

3. Перезапустите Postgresql, выполнив команду:

```
sudo systemctl restart postgresql@1X-main.service
```

```
astraregul@astraregul:~$ sudo systemctl restart postgresql@14-main.service
```

4. Введите в терминале следующую команду:

```
sudo su - postgres
```

```
astraregul@astraregul:~$ sudo su - postgres
```

5. Смените пароль пользователя "postgres":

psql -c "alter user postgres with password '<пароль>'"

```
postgres@astraregul:~$ psql -c "alter user postgres with password 'postgres'"
ALTER ROLE
```

6. Создайте новую базу данных для ситсемы отчетности, выполнив команду:

createdb -T template0 Reports

```
postgres@astraregul:~$ createdb -T template0 Reports
```

7. Выполните команду для восстановления базы данных из резрвной копии:

psql -d <название БД> -f /opt/Automiq/Alpha.Reports.Base/SQL/dump.sql

```
postgres@astraregul:~$ psql -d Reports -f /opt/Automiq/Alpha.Reports.Base/SQL/dump.sql
```

8. После настройки СУБД для возвращения в основного ползователя, выполните следующую команду:

exit

Настройка аутентификации клиентов

Аутентификация клиентов управляется конфигурационным файлом "pg_hba.conf", который расположен в каталоге с данными кластера базы данных. Файл "pg_hba.conf" со стандартным содержимым, создаётся командой "initdb" при инициализации каталога с данными.

Обычный формат файла "pg_hba.conf" представляет собой набор записей, по одной в строке. Пустые строки игнорируются, как и любой текст комментария

после знака #. Записи не продолжаются на следующей строке. Записи состоят из некоторого количества полей, разделённых между собой пробелами и/или отступами (TAB). В полях могут быть использованы пробелы, если они взяты в кавычки. Если в кавычки берётся какое-либо зарезервированное слово в поле базы данных, пользователя или адресации (например, all или replication), то слово теряет своё особое значение и просто обозначает базу данных, пользователя или сервер с данным именем.

Каждая зfпись обозначает тип соединения, диапазон IP-адресов клиента (если он соотносится с типом соединения), имя базы данных, имя пользователя, и способ аутентификации, который будет использован для соединения в соответствии с этими параметрами. Первая запись с соответствующим типом соединения, адресом клиента, указанной базой данных и именем пользователя применяется для аутентификации. Процедур «fall-through» или «backup» не предусмотрено: если выбрана запись и аутентификация не прошла, последующие записи не рассматриваются. Если же ни одна из записей не подошла, в доступе будет отказано.

Запись может быть сделана в одном из семи форматов:

```
local      база    пользователь    метод-аутентификации [параметры]
host       база    пользователь    адрес     метод-аутентификации
[параметры]
hostssl    база    пользователь    адрес     метод-аутентификации
[параметры]
hostnossl  база    пользователь    адрес     метод-аутентификации
[параметры]
host       база    пользователь    IP-адрес  IP-маска  метод-
аутентификации [параметры]
hostssl    база    пользователь    IP-адрес  IP-маска  метод-
аутентификации [параметры]
hostnossl  база    пользователь    IP-адрес  IP-маска  метод-
аутентификации [параметры]
```

Описания полей

Поле	Описание

local	Управляет подключениями через Unix-сокеты. Без подобной записи подключение через Unix-сокеты невозможно.
host	Управляет подключениями, устанавливаемыми по TCP/IP. Записи host соответствуют подключениям с SSL и без SSL. Удалённое соединение по TCP/IP невозможно, если сервер запущен без определения соответствующих значений для параметра конфигурации "listen_addresses".
hostssl	Управляет подключениями, устанавливаемыми по TCP/IP с применением шифрования SSL. Чтобы использовать эту возможность: › сервер должен быть собран с поддержкой SSL. › механизм SSL должен быть включён параметром конфигурации ssl.
hostnossl	Этот тип записей противоположен hostssl, ему соответствуют только подключения по TCP/IP без шифрования SSL.
база	Определяет, каким именам баз данных соответствует запись. Возможные значения: › «all» – подходят все базы данных. › «sameuser» – данная запись соответствует только, если имя запрашиваемой базы данных совпадает с именем запрашиваемого пользователя. › «samerole» (или устар. «samegroup») – запрашиваемый пользователь должен быть членом роли с таким же именем, как и у запрашиваемой базы данных. Суперпользователи не становятся членами роли автоматически, а только если они являются явными членами роли. › «replication» – запись соответствует, если запрашивается подключение для физической репликации. Для таких подключений не выбирается какая-то конкретная база данных. › Любое другое значение воспринимается как имя определённой базы данных. Несколько имён баз

	данных можно указать, разделяя их запятыми. Файл, содержащий имена баз данных, можно указать, поставив знак «@» в начале его имени
пользователь	Указывает, какому имени (или именам) пользователя базы данных соответствует запись. Возможные значения: › «all» – запись соответствует всем пользователям. › Любое другое значение задаёт либо имя конкретного пользователя базы данных, либо имя группы (если это значение начинается с +). Несколько имён пользователей можно указать, разделяя их запятыми. Файл, содержащий имена пользователей, можно указать, поставив знак «@» в начале его имени.
адрес	Указывает адрес (или адреса) клиентского компьютера, которому соответствует запись. Поле может содержать или имя компьютера, или диапазон IP-адресов, или одно из ключевых слов: › «all» – любой IP-адрес. › «samehost» – любые IP-адреса данного сервера. › «samenet» – любой адрес любой подсети, к которой сервер подключён напрямую.
IP-адрес IP-маска	Эти два поля могут быть использованы как альтернатива записи IP-адрес/длина-маски. Вместо того, чтобы указывать длину маски, в отдельном столбце указывается сама маска. Например, 255.0.0.0 представляет собой маску CIDR для IPv4 длиной 8 бит, а 255.255.255.255 представляет маску CIDR длиной 32 бита. Эти поля применимы только к записям host, hostssl и hostnossl
метод-аутентификации	Указывает метод аутентификации, когда подключение соответствует записи. Все значения воспринимаются с учётом регистра и должны быть записаны в нижнем регистре. Возможные значения: › «trust» – разрешает безусловное подключение. Этот метод позволяет тому, кто может подключиться к серверу с базой данных Postgres Pro, войти под любым желаемым

пользователем Postgres Pro без введения пароля и без какой-либо другой аутентификации.

- «reject» – отклоняет подключение безусловно. Эта возможность полезна для «фильтрации» некоторых серверов группы, например, строка с «reject» может отклонить попытку подключения одного компьютера, при этом следующая строка позволяет подключиться остальным компьютерам в той же сети.
- «scram-sha-256» – проверяет пароль пользователя, производя аутентификацию SCRAM-SHA-256.
- «md5» – проверяет пароль пользователя, производя аутентификацию SCRAM-SHA-256 или MD5.
- «password» – требует для аутентификации введения клиентом незашифрованного пароля. Поскольку пароль посылается простым текстом через сеть, такой способ не стоит использовать, если сеть не вызывает доверия.
- «gss» – для аутентификации пользователя использует GSSAPI. Этот способ доступен только для подключений по TCP/IP.
- «sspi» – для аутентификации пользователя использует SSPI. Способ доступен только для Windows.
- «ident» – получает имя пользователя операционной системы клиента, связываясь с сервером Ident, и проверяет, соответствует ли оно имени пользователя базы данных. Аутентификация «ident» может использоваться только для подключений по TCP/IP. Для локальных подключений применяется аутентификация «peer».
- «peer» – получает имя пользователя операционной системы клиента из операционной системы и проверяет, соответствует ли оно имени пользователя запрашиваемой базы данных. Доступно только для локальных подключений.
- «ldap» – проводит аутентификацию, используя сервер LDAP.

	› «radius» – проводит аутентификацию, используя сервер RADIUS. › «cert» – проводит аутентификацию, используя клиентский сертификат SSL. › «pam» – проводит аутентификацию, используя службу подключаемых модулей аутентификации (PAM), предоставляемую операционной системой. › «bsd» – проводит аутентификацию, используя службу аутентификации BSD, предоставляемую операционной системой.
параметры	Поле (поля) вида имя=значение, определяющее параметры метода аутентификации. Подробнее о параметрах, доступных для различных методов аутентификации, рассказывается ниже. Помимо описанных далее параметров, относящихся к различным методам, есть один общий параметр аутентификации clientcert, который можно задать в любой записи hostssl. Если он равен 1, клиент должен представить подходящий (доверенный) сертификат SSL, в дополнение к другим требованиям метода проверки подлинности.

Файлы, включённые в конструкции, начинающиеся с «@», читаются, как список имён, разделённых запятыми или пробелами. Комментарии предваряются знаком «#», как и в файле "pg_hba.conf", и вложенные «@» конструкции допустимы. Если только имя файла, начинающегося с «@» не является абсолютным путём.

Поскольку записи файла "pg_hba.conf" рассматриваются последовательно для каждого подключения, порядок записей имеет большое значение. Обычно более ранние записи определяют чёткие критерии для соответствия параметров подключения, но для методов аутентификации допускают послабления. Напротив, записи более поздние смягчают требования к соответствию параметров подключения, но усиливают их в отношении методов аутентификации.

Например, некто желает использовать trust аутентификацию для локального подключения по TCP/IP, но при этом запрашивать пароль для удалённых подключений по TCP/IP. В этом случае запись, устанавливающая аутентификацию trust для подключения адреса 127.0.0.1, должна предшествовать записи, определяющей аутентификацию по паролю для более широкого диапазона клиентских IP-адресов.

Файл "pg_hba.conf" прочитывается при запуске системы, а также в тот момент, когда основной сервер (процесс "postmaster") получает сигнал "SIGHUP". Послать сигнал процессу можно любым из способов:

- используя pg_ctl reload
- вызвав SQL-функцию pg_reload_conf()
- выполнив kill -HUP

Предыдущее утверждение не касается Microsoft Windows: там любые изменения в "pg_hba.conf" сразу применяются к последующим подключениям.

Чтобы подключиться к конкретной базе данных, пользователь не только должен пройти все проверки файла "pg_hba.conf", но должен иметь привилегию "CONNECT" для подключения к базе данных. Если вы хотите ограничить доступ к базам данных для определённых пользователей, проще предоставить/отозвать привилегию "CONNECT", нежели устанавливать правила в записях файла "pg_ hba.conf".

1.1.2.2.1.1. Примеры настройки конфигурационного файла

Позволяет любому пользователю локальной системы подключаться к любой базе данных, используя любое имя пользователя баз данных, через Unix-сокеты (по умолчанию для локальных подключений).

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
local       all             all                         trust
```

То же, но для локальных замкнутых подключений по TCP/IP.

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         127.0.0.1/32    trust
```

То же, что и в предыдущем примере, но с указанием сетевой маски в отдельном столбце

```
# TYPE    DATABASE    USER    ADDRESS    IP-MASK           METHOD
host      all         all     127.0.0.1  255.255.255.255   trust
```

То же для IPv6.

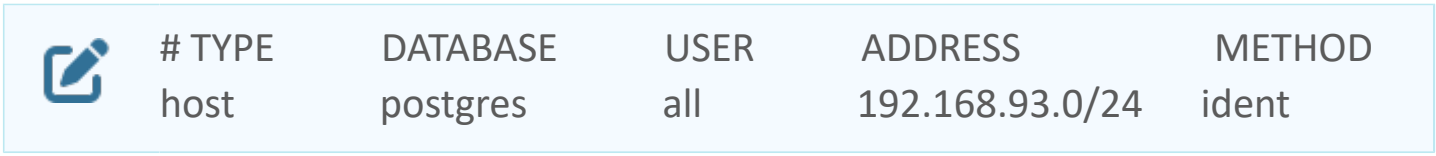
```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         ::1/128         trust
```

То же самое, но с использованием имени компьютера (обычно покрывает и IPv4, и IPv6).

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         localhost       trust
```

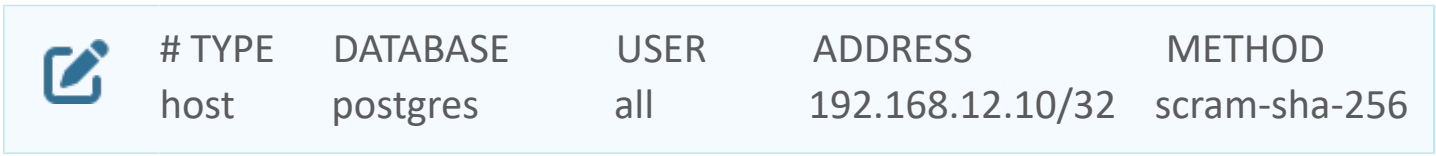
Позволяет любому пользователю любого компьютера с IP-адресом 192.168.93.x подключаться к базе данных "postgres" с именем, которое

сообщает для данного подключения ident (как правило, имя пользователя операционной системы).



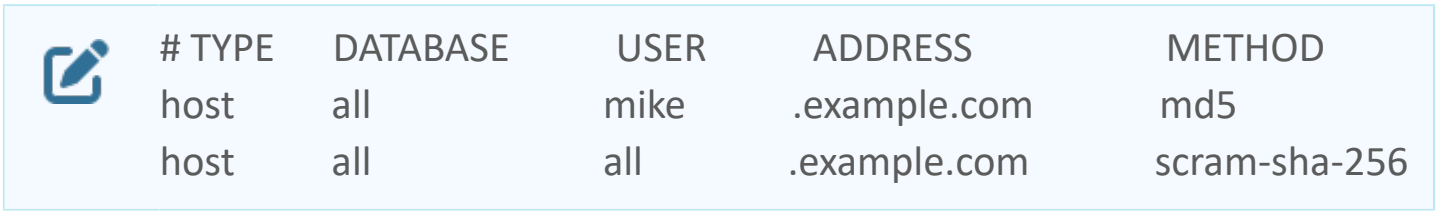
# TYPE	DATABASE	USER	ADDRESS	METHOD
host	postgres	all	192.168.93.0/24	ident

Позволяет любому пользователю компьютера 192.168.12.10 подключаться к базе данных "postgres", если он передаёт правильный пароль.



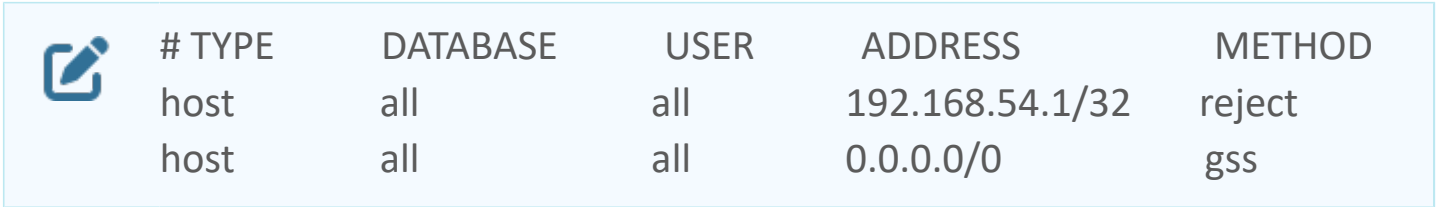
# TYPE	DATABASE	USER	ADDRESS	METHOD
host	postgres	all	192.168.12.10/32	scram-sha-256

Позволяет любым пользователям с компьютеров в домене example.com подключаться к любой базе данных, если передаётся правильный пароль. Для всех пользователей требуется аутентификация SCRAM, за исключением пользователя 'mike', который использует старый клиент, не поддерживающий аутентификацию SCRAM.



# TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	mike	.example.com	md5
host	all	all	.example.com	scram-sha-256

В случае отсутствия предшествующих строчек с "host", следующие две строки откажут в подключении с 192.168.54.1 (поскольку данная запись будет выбрана первой), но разрешат подключения GSSAPI с любых других адресов. С нулевой маской ни один бит из IP-адреса компьютера не учитывается, так что этой строке соответствует любой компьютер.



# TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	all	192.168.54.1/32	reject
host	all	all	0.0.0.0/0	gss

Позволяет пользователям с любого компьютера 192.168.x.x подключаться к любой базе данных, если они проходят проверку ident. Если же ident говорит, например, что это пользователь "bryanh" и он запрашивает подключение как пользователь Postgres Pro "guest1", подключение будет разрешено, если в

файле pg_ident.conf есть сопоставление "omicron", позволяющее пользователю "bryanh" подключаться как "guest1".

```
# TYPE    DATABASE        USER     ADDRESS          METHOD
host      all             all      192.168.0.0/16   ident map=omicron
```

Если для локальных подключений предусмотрены только эти три строки, они позволят локальным пользователям подключаться только к своим базам данных (базам данных с именами, совпадающими с именами пользователей баз данных), кроме администраторов или членов роли "support", которые могут подключиться к любой БД. Список имён администраторов содержится в файле $PGDATA/admins. Пароли запрашиваются в любом случае.

```
# TYPE    DATABASE        USER         ADDRESS          METHOD
local     sameuser        all                           md5
local     all             @admins                       md5
local     all             +support                      md5
```

Последние две строчки в примере выше могут быть объединены в одну:

```
# TYPE    DATABASE        USER                 ADDRESS          METHOD
local     all             @admins,+support                      md5
```

В столбце DATABASE могут указываться списки и имена файлов:

```
# TYPE    DATABASE                USER         ADDRESS          METHOD
local     db1,db2,@demodbs        all                           md5
```

1.1.2.2.2. Настройка Alpha.Reports.Base

Все файлы настроек имеют расширение *.json и находятся рядом с исполняемым файлом в каталоге установки /opt/Automiq/Alpha.Reports.Base

Для настройки Alpha.Reports.Base выполните следующие действия:

1. В каталоге установки Alpha.Reports.Base откройте с помощью следующей команды файл "appsettings.json".

sudo nano /opt/Automiq/Alpha.Reports.Base/appsettings.json

```
astraregul@astraregul:~$ sudo nano /opt/Automiq/Alpha.Reports.Base/appsettings.json
```

2. Отредактируйте файл конфигурации:

В разделе "DataBaseServer" укажите следующие параметры:

- В пункте "Address" введите IP-адрес базы данных или доменное имя "localhost".
  Пример: "Address": "localhost".
- В пункте "DataBase" введите имя базы данных
  Пример: "DataBase": "Reports".
- В пункте "Login" введите логин для базы данных. По умолчанию: "postgres".
  Пример: "Login": "postgres".
- В пункте "Password" введите пароль для базы данных, который Вы указали при настройке PostgreSQL.
  Пример: "Password": "postgres".

Пароль не должен быть пустым.

В разделе "LDAPSecurity" укажите следующие параметры:

- В пункте "BaseName" замените "AlphaSecurity" на "AstraSecurity":
  Пример: "BaseName": "AstraSecurity".
- В пункте "Host" введите "IP-адрес компьютера, на котором установлен LDAP.
  Пример: "Host": "127.0.0.1".

Остальные параметры оставьте по-умолчаню.

В списке "EntryPoints" нужно перечислить все адреса, по которым должен быть доступен базовый сервер, в том числе все сетевые карты компьютера (если их больше одной) – для CORS политики.

Обращаться к серверу нужно строго по IP-адресу, указанному в поле "Application" → "ServerAddress".
Значение «localhost» – не равно «127.0.0.1».

IP-адрес, указанный в поле "Modules" → "Alpha.Reports.Report.Server" должен совпадать с IP-адресом, указанным в поле "Application" → "ServerAddress".

3. Запустите/перезапустите службу "Alpha.Reports.Base" выполнив команду:

sudo systemctl restart alpha.reports.base.service

```
astraregul@astraregul:~$ sudo systemctl restart alpha.reports.base.service
```

1.1.2.2.2.1. Пример файла конфигурации

```
{
    "Kestrel": {
        "EndPoints": {
            "HTTP": {
                "Url": "http://*:5000"
            }
        }
    },

    "Application": {
        "Name": "Alpha.Reports.Base",
        "ServerAddress": "http://127.0.0.1:5000",
        "LicenseFile": "license.lic",
        "BasePath": "",
        "RedirectToHTTPS": "false",
        "LdapServer": "",

        "DataBaseServer": {
            "Address": "localhost",
            "DataBase": "Reports",
            "Login": "postgres",
            "Password": "postgres"
        },

        "LDAPSecurity": {
            "Host": "127.0.0.1",
            "Port": "389",
            "BaseName": "AstraSecurity",
            "AppName": "Reports",
            "Rights": {
                "AdminRight": "Administration",
                "ViewRight": "View"
            }
        },

        "Modules": {
```

```
                    "Alpha.Reports.Server": "http://127.0.0.1:5001"
                },

                "AlphaLicenseName": "",

                "DisableDataAccessAutoUpdate": "false",

                "AllowAllNIC": "false",
                "EntryPoints": [
                    "http://127.0.0.1:5000",
                    "http://192.168.0.1:5000",
                    "http://somehostname:5000"
                ],

                "CleanupInterval": 43200
            }
        }
```

1.1.2.2.3. Настройка Alpha.Reports.Server

Если при формировании отчётов по расписанию необходимо отправлять их по списку электронных адресов, то требуется сконфигурировать работу сервера отчётов.

Все файлы настроек имеют расширение *.json и находятся рядом с исполняемым файлом в каталоге установки /opt/Automiq/Alpha.Reports.Server

Для настройки Alpha.Reports.Server выполните следующие действия:

1. В каталоге установки Alpha.Reports.Server откройте с помощью следующей команды файл "appsettings.json".

sudo nano /opt/Automiq/Alpha.Reports.Server/appsettings.json

2. Отредактируйте файл конфигурации:

Раздел "DataBaseServer" необходимо заполнить аналогично Alpha.Reports.Base.

В разделе "SMTP" заполните параметры:

- Address – адрес smtp-сервера.
- Port – порт для подключения, по умолчанию 25.
- Login – почтовый адрес (полный), с которого будет осуществляться рассылка.
- Password – пароль для подключения к smtp-серверу.

3. Запустите/перезапустите службу "Alpha.Reports.Server", выполнив команду:

sudo systemctl restart alpha.reports.server.service

```
astraregul@astraregul:~$ sudo systemctl restart alpha.reports.server.service
```

1.1.2.2.3.1. Пример файла appsettings.json

```
{
    "Kestrel": {
        "EndPoints": {
            "HTTP": {
                "Url": "http://*:5001"
            }
        }
    },

    "Application": {
        "Name": "Alpha.Reports.Server",
        "BasePath": "",
        "RedirectToHTTPS": "false",

        "DataBaseServer": {
            "Address": "localhost",
            "DataBase": "Reports",
            "Login": "postgres",
            "Password": "postgres"
        },

        "Modules": {
            "Alpha.Base.Server": "http://127.0.0.1:5000"
        },

        "SMTP": {
            "Address": "smtp.example.com",
            "Port": "25",
            "Login": "alpha_reports@example.com",
            "Password":  "password"
        },

        "JobFile": "Alpha.Reports.Server.Jobs.xml"
    }
}
```

1.1.2.2.4. Секретная строка

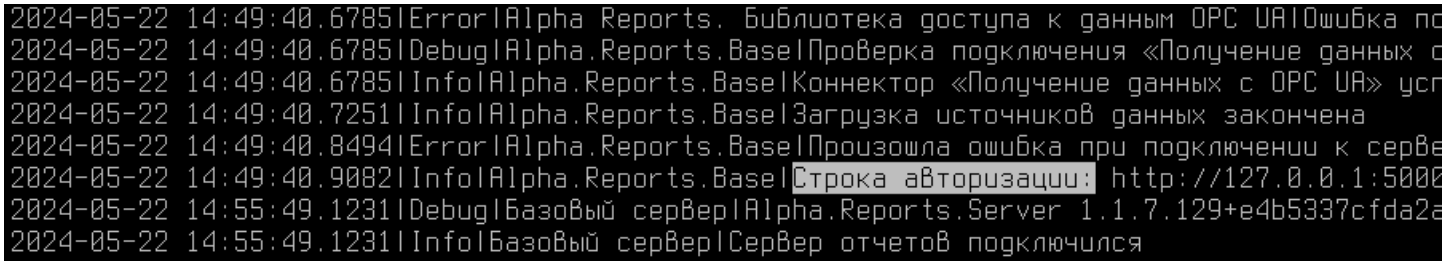
Для работы с системой отчётности необходима секретная строка, которая генерируется при запуске "Alpha.Reports.Base".

Получить секретную строку можно любым из способов:

- Программно обратитесь к функции "GetAuthString()".
- В файле /var/log/Automiq/Alpha.Reports/Base.log найдите строку

<Дата> <Время>|Info|Alpha.Reports.Base|Строка авторизации:<Строка авторизации>

Пример

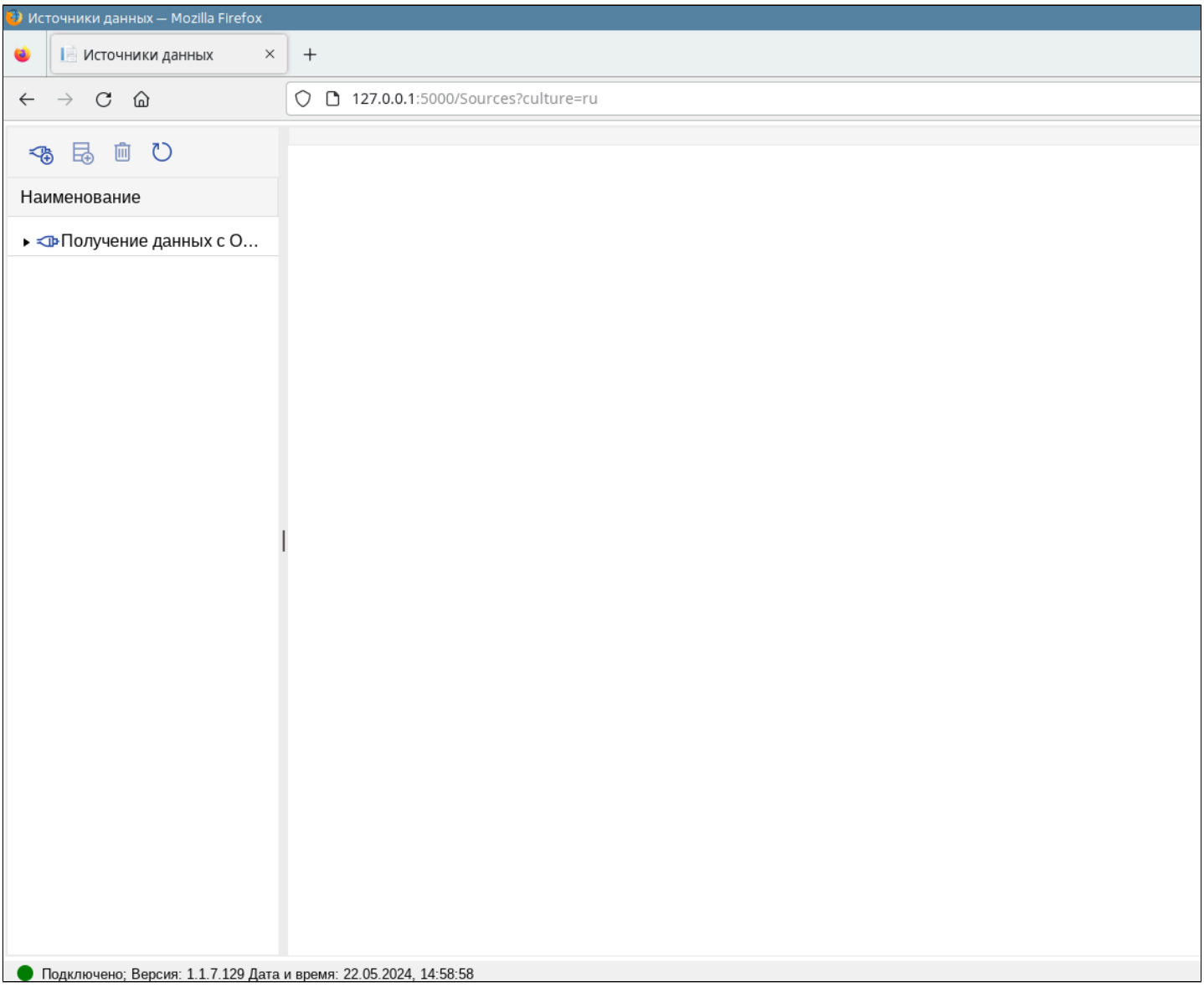


1.1.2.2.5. Источники данных

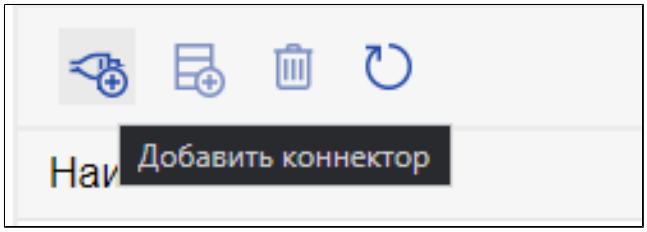
1. Чтобы открыть панель конфигурирования источников, в браузере введите:

http://127.0.0.1:5000/Sources/?u=<Секретная строка>

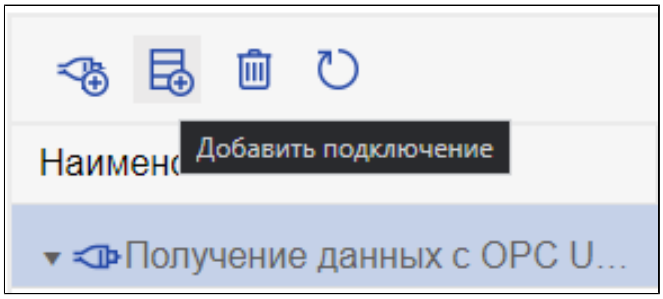
При успешном запуске в браузере отобразится окно:



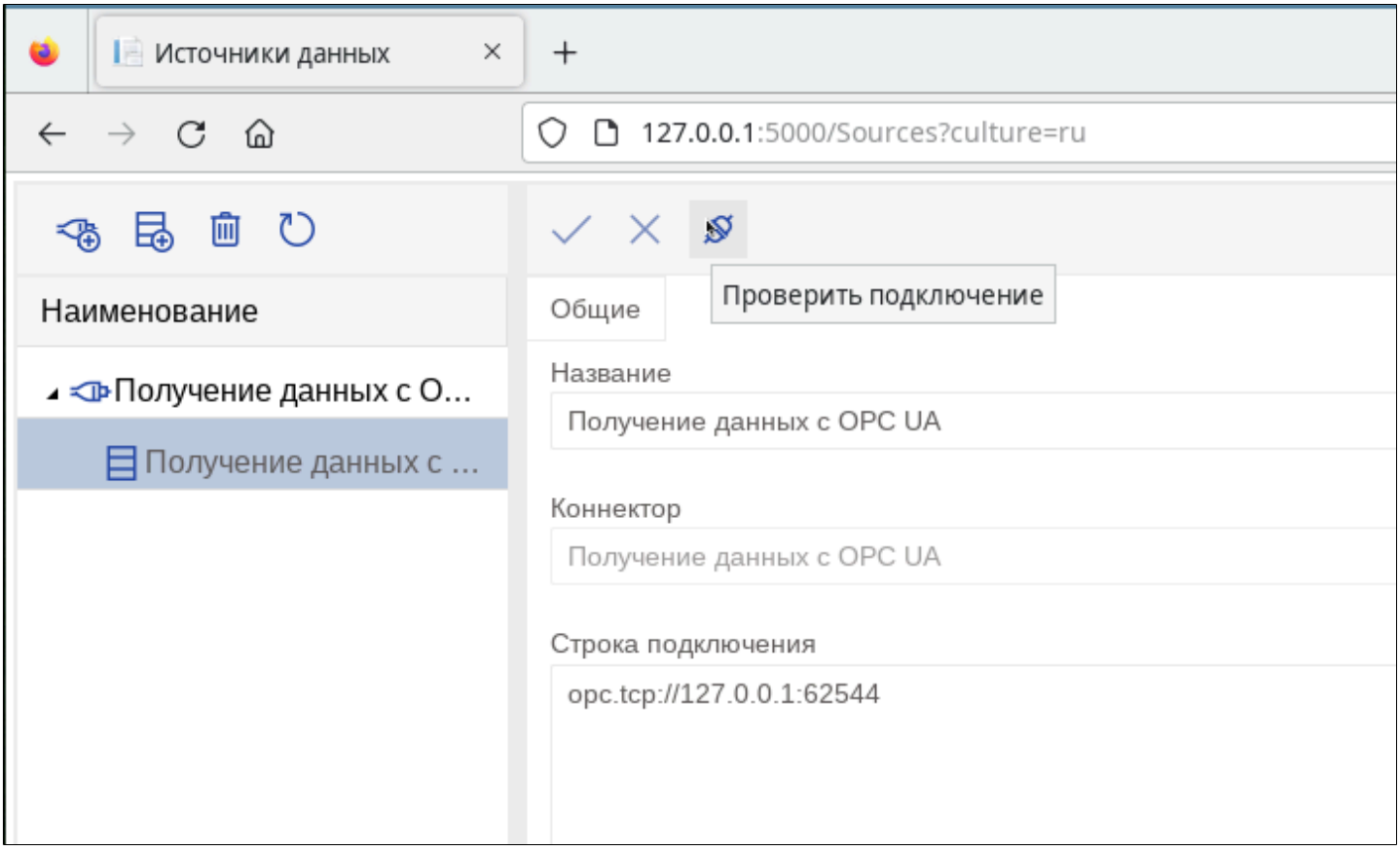
2. По умолчанию источники данных уже настроены и можно переходить к пункту 4. Если источники данных не настроены, нажмите кнопку "Добавить коннектор".



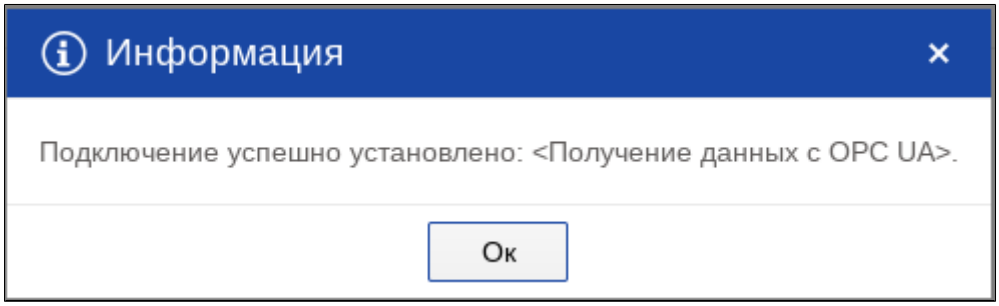
3. Выделите коннектор и нажмите кнопку «Добавить подключение».



4. Настройте строку подключения у коннектора и проверьте подключение.



При успешном подключении появится следующее окно:


Информация
Подключение успешно установлено: <Получение данных с OPC UA>.
Ок

1.1.2.3. РЕД ОС 7.3

Настройка СУБД
Настройка Alpha.Reports.Base
Настройка Alpha.Reports.Server
Секретная строка
Источники данных

Совместимость с РЕД ОС 8 на текущий момент не подтверждена.

1.1.2.3.1. Настройка СУБД

Для настройки СУБД выполните следующие действия:

1. Войдите под пользователем root, выполнив следующую команду и введите пароль администратора:

```
su -
```

```
[astraregul@localhost ~]$ su -
Пароль:
[root@localhost ~]#
```

2. Откройте файл "postgresql.conf", выполнив команду:

```
sudo nano /var/lib/pgsql/1X/data/postgresql.conf
```

```
[root@localhost ~]# udo nano /var/lib/pgsql/16/data/postgresql.conf
```

3. Расскоментируйте строку "port = 5432":

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

#listen_addresses = 'localhost'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#reserved_connections = 0               # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/var/run/postgresql' # comma-separated list of director
```

Для сохранения и выхода нажмите комбинацию клавиш "Ctrl+x". Для подтверждения изменений нажмите клавишу "y" и нажмите клавишу "Enter".

4. Перезапустите Postgresql, выполнив команду:

sudo systemctl restart postgresql-1X.service

```
[root@localhost ~]# sudo systemctl restart postgresql-16.service
```

5. Введите в терминале следующую команду:

sudo su - postgres

```
[root@localhost ~]# sudo su - postgres
```

6. Смените пароль пользователя "postgres":

psql -c "alter user postgres with password '<пароль>'"

```
[postgres@localhost ~]$ psql -c "alter user postgres with password 'postgres'"
ALTER ROLE
```

7. Создайте новую базу данных для ситсемы отчетности, выполнив команду:

createdb -T template0 Reports

```
[postgres@localhost ~]$ createdb -T template0 Reports
```

8. Выполните команду для восстановления базы данных из резрвной копии:

psql -d <название БД> -f /opt/Automiq/Alpha.Reports.Base/SQL/dump.sql

```
[postgres@localhost ~]$ psql -d Reports -f /opt/Automiq/Alpha.Reports.Base/SQL/dump.sql
```

9. После настройки СУБД для возвращения в основного ползователя, выполните следующую команду:

exit

Настройка аутентификации клиентов

Аутентификация клиентов управляется конфигурационным файлом "pg_hba.conf", который расположен в каталоге с данными кластера базы данных. Файл "pg_hba.conf" со стандартным содержимым, создаётся командой "initdb" при инициализации каталога с данными.

Обычный формат файла "pg_hba.conf" представляет собой набор записей, по одной в строке. Пустые строки игнорируются, как и любой текст комментария после знака #. Записи не продолжаются на следующей строке. Записи состоят из некоторого количества полей, разделённых между собой пробелами и/или отступами (TAB). В полях могут быть использованы пробелы, если они взяты в кавычки. Если в кавычки берётся какое-либо зарезервированное слово в поле базы данных, пользователя или адресации (например, all или replication), то слово теряет своё особое значение и просто обозначает базу данных, пользователя или сервер с данным именем.

Каждая зfпись обозначает тип соединения, диапазон IP-адресов клиента (если он соотносится с типом соединения), имя базы данных, имя пользователя, и способ аутентификации, который будет использован для соединения в соответствии с этими параметрами. Первая запись с соответствующим типом соединения, адресом клиента, указанной базой данных и именем пользователя применяется для аутентификации. Процедур «fall-through» или

«backup» не предусмотрено: если выбрана запись и аутентификация не прошла, последующие записи не рассматриваются. Если же ни одна из записей не подошла, в доступе будет отказано.

Запись может быть сделана в одном из семи форматов:

local база пользователь метод-аутентификации [параметры]
host база пользователь адрес метод-аутентификации [параметры]
hostssl база пользователь адрес метод-аутентификации [параметры]
hostnossl база пользователь адрес метод-аутентификации [параметры]
host база пользователь IP-адрес IP-маска метод-аутентификации [параметры]
hostssl база пользователь IP-адрес IP-маска метод-аутентификации [параметры]
hostnossl база пользователь IP-адрес IP-маска метод-аутентификации [параметры]

Описания полей

Поле	Описание
local	Управляет подключениями через Unix-сокеты. Без подобной записи подключение через Unix-сокеты невозможно.
host	Управляет подключениями, устанавливаемыми по TCP/IP. Записи host соответствуют подключениям с SSL и без SSL. Удалённое соединение по TCP/IP невозможно, если сервер запущен без определения соответствующих значений для параметра конфигурации "listen_addresses".
hostssl	Управляет подключениями, устанавливаемыми по TCP/IP с применением шифрования SSL. Чтобы использовать эту возможность: › сервер должен быть собран с поддержкой SSL. › механизм SSL должен быть включён параметром конфигурации ssl.

hostnossl	Этот тип записей противоположен hostssl, ему соответствуют только подключения по TCP/IP без шифрования SSL.
база	Определяет, каким именам баз данных соответствует запись. Возможные значения: › «all» – подходят все базы данных. › «sameuser» – данная запись соответствует только, если имя запрашиваемой базы данных совпадает с именем запрашиваемого пользователя. › «samerole» (или устар. «samegroup») – запрашиваемый пользователь должен быть членом роли с таким же именем, как и у запрашиваемой базы данных. Суперпользователи не становятся членами роли автоматически, а только если они являются явными членами роли. › «replication» – запись соответствует, если запрашивается подключение для физической репликации. Для таких подключений не выбирается какая-то конкретная база данных. ›Любое другое значение воспринимается как имя определённой базы данных. Несколько имён баз данных можно указать, разделяя их запятыми. Файл, содержащий имена баз данных, можно указать, поставив знак «@» в начале его имени
пользователь	Указывает, какому имени (или именам) пользователя базы данных соответствует запись. Возможные значения: ›«all» – запись соответствует всем пользователям. ›Любое другое значение задаёт либо имя конкретного пользователя базы данных, либо имя группы (если это значение начинается с +). Несколько имён пользователей можно указать, разделяя их запятыми. Файл, содержащий имена пользователей, можно указать, поставив знак «@» в начале его имени.

адрес	Указывает адрес (или адреса) клиентского компьютера, которому соответствует запись. Поле может содержать или имя компьютера, или диапазон IP-адресов, или одно из ключевых слов: › «all» – любой IP-адрес. › «samehost» – любые IP-адреса данного сервера. › «samenet» – любой адрес любой подсети, к которой сервер подключён напрямую.
IP-адрес IP-маска	Эти два поля могут быть использованы как альтернатива записи IP-адрес/длина-маски. Вместо того, чтобы указывать длину маски, в отдельном столбце указывается сама маска. Например, 255.0.0.0 представляет собой маску CIDR для IPv4 длиной 8 бит, а 255.255.255.255 представляет маску CIDR длиной 32 бита. Эти поля применимы только к записям host, hostssl и hostnossl
метод-аутентификации	Указывает метод аутентификации, когда подключение соответствует записи. Все значения воспринимаются с учётом регистра и должны быть записаны в нижнем регистре. Возможные значения: › «trust» – разрешает безусловное подключение. Этот метод позволяет тому, кто может подключиться к серверу с базой данных Postgres Pro, войти под любым желаемым пользователем Postgres Pro без введения пароля и без какой-либо другой аутентификации. › «reject» – отклоняет подключение безусловно. Эта возможность полезна для «фильтрации» некоторых серверов группы, например, строка с «reject» может отклонить попытку подключения одного компьютера, при этом следующая строка позволяет подключиться остальным компьютерам в той же сети. › «scram-sha-256» – проверяет пароль пользователя, производя аутентификацию SCRAM-SHA-256. › «md5» – проверяет пароль пользователя, производя аутентификацию SCRAM-SHA-256 или MD5.

	› «password» – требует для аутентификации введения клиентом незашифрованного пароля. Поскольку пароль посылается простым текстом через сеть, такой способ не стоит использовать, если сеть не вызывает доверия. › «gss» – для аутентификации пользователя использует GSSAPI. Этот способ доступен только для подключений по TCP/IP. › «sspi» – для аутентификации пользователя использует SSPI. Способ доступен только для Windows. › «ident» – получает имя пользователя операционной системы клиента, связываясь с сервером Ident, и проверяет, соответствует ли оно имени пользователя базы данных. Аутентификация «ident» может использоваться только для подключений по TCP/IP. Для локальных подключений применяется аутентификация «peer». › «peer» – получает имя пользователя операционной системы клиента из операционной системы и проверяет, соответствует ли оно имени пользователя запрашиваемой базы данных. Доступно только для локальных подключений. › «ldap» – проводит аутентификацию, используя сервер LDAP. › «radius» – проводит аутентификацию, используя сервер RADIUS. › «cert» – проводит аутентификацию, используя клиентский сертификат SSL. › «pam» – проводит аутентификацию, используя службу подключаемых модулей аутентификации (PAM), предоставляемую операционной системой. › «bsd» – проводит аутентификацию, используя службу аутентификации BSD, предоставляемую операционной системой.
параметры	Поле (поля) вида имя=значение, определяющее параметры метода аутентификации. Подробнее о параметрах,

	доступных для различных методов аутентификации, рассказывается ниже. Помимо описанных далее параметров, относящихся к различным методам, есть один общий параметр аутентификации clientcert, который можно задать в любой записи hostssl. Если он равен 1, клиент должен представить подходящий (доверенный) сертификат SSL, в дополнение к другим требованиям метода проверки подлинности.

Файлы, включённые в конструкции, начинающиеся с «@», читаются, как список имён, разделённых запятыми или пробелами. Комментарии предваряются знаком «#», как и в файле "pg_hba.conf", и вложенные «@» конструкции допустимы. Если только имя файла, начинающегося с «@» не является абсолютным путём.

Поскольку записи файла "pg_hba.conf" рассматриваются последовательно для каждого подключения, порядок записей имеет большое значение. Обычно более ранние записи определяют чёткие критерии для соответствия параметров подключения, но для методов аутентификации допускают послабления. Напротив, записи более поздние смягчают требования к соответствию параметров подключения, но усиливают их в отношении методов аутентификации.

Например, некто желает использовать trust аутентификацию для локального подключения по TCP/IP, но при этом запрашивать пароль для удалённых подключений по TCP/IP. В этом случае запись, устанавливающая аутентификацию trust для подключения адреса 127.0.0.1, должна предшествовать записи, определяющей аутентификацию по паролю для более широкого диапазона клиентских IP-адресов.

Файл "pg_hba.conf" прочитывается при запуске системы, а также в тот момент, когда основной сервер (процесс "postmaster") получает сигнал "SIGHUP". Послать сигнал процессу можно любым из способов:

- используя pg_ctl reload
- вызвав SQL-функцию pg_reload_conf()
- выполнив kill -HUP

Предыдущее утверждение не касается Microsoft Windows: там любые изменения в "pg_hba.conf" сразу применяются к последующим подключениям.

Чтобы подключиться к конкретной базе данных, пользователь не только должен пройти все проверки файла "pg_hba.conf", но должен иметь привилегию "CONNECT" для подключения к базе данных. Если вы хотите ограничить доступ к базам данных для определённых пользователей, проще предоставить/отозвать привилегию "CONNECT", нежели устанавливать правила в записях файла "pg_ hba.conf".

1.1.2.3.1.1. Примеры настройки конфигурационного файла

Позволяет любому пользователю локальной системы подключаться к любой базе данных, используя любое имя пользователя баз данных, через Unix-сокеты (по умолчанию для локальных подключений).

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
local       all             all                         trust
```

То же, но для локальных замкнутых подключений по TCP/IP.

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         127.0.0.1/32    trust
```

То же, что и в предыдущем примере, но с указанием сетевой маски в отдельном столбце

```
# TYPE    DATABASE    USER    ADDRESS    IP-MASK           METHOD
host      all         all     127.0.0.1  255.255.255.255   trust
```

То же для IPv6.

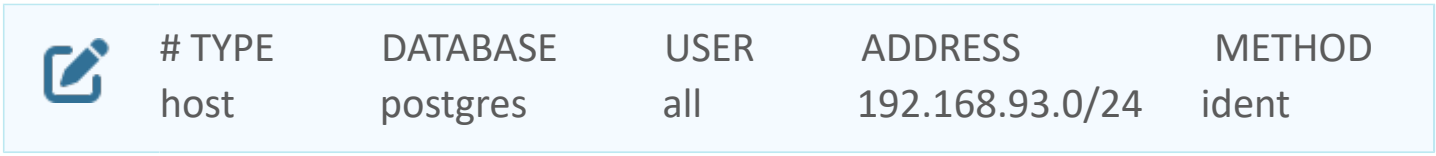
```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         ::1/128         trust
```

То же самое, но с использованием имени компьютера (обычно покрывает и IPv4, и IPv6).

```
# TYPE      DATABASE        USER        ADDRESS         METHOD
host        all             all         localhost       trust
```

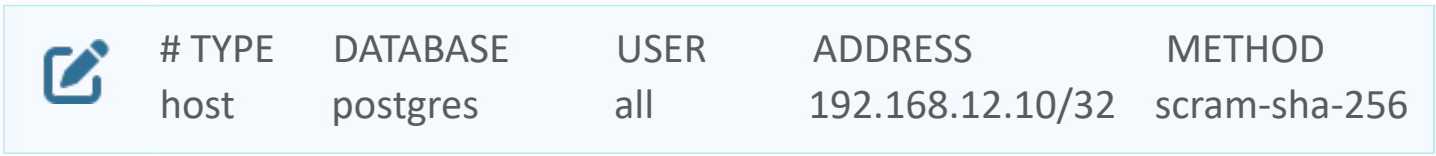
Позволяет любому пользователю любого компьютера с IP-адресом 192.168.93.x подключаться к базе данных "postgres" с именем, которое

сообщает для данного подключения ident (как правило, имя пользователя операционной системы).



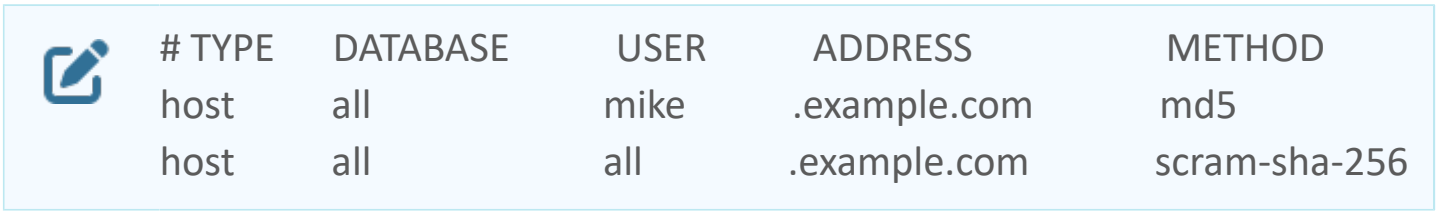
# TYPE	DATABASE	USER	ADDRESS	METHOD
host	postgres	all	192.168.93.0/24	ident

Позволяет любому пользователю компьютера 192.168.12.10 подключаться к базе данных "postgres", если он передаёт правильный пароль.



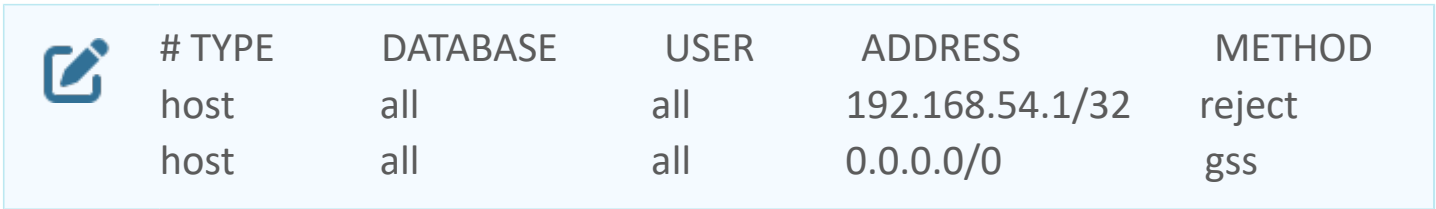
# TYPE	DATABASE	USER	ADDRESS	METHOD
host	postgres	all	192.168.12.10/32	scram-sha-256

Позволяет любым пользователям с компьютеров в домене example.com подключаться к любой базе данных, если передаётся правильный пароль. Для всех пользователей требуется аутентификация SCRAM, за исключением пользователя 'mike', который использует старый клиент, не поддерживающий аутентификацию SCRAM.



# TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	mike	.example.com	md5
host	all	all	.example.com	scram-sha-256

В случае отсутствия предшествующих строчек с "host", следующие две строки откажут в подключении с 192.168.54.1 (поскольку данная запись будет выбрана первой), но разрешат подключения GSSAPI с любых других адресов. С нулевой маской ни один бит из IP-адреса компьютера не учитывается, так что этой строке соответствует любой компьютер.



# TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	all	192.168.54.1/32	reject
host	all	all	0.0.0.0/0	gss

Позволяет пользователям с любого компьютера 192.168.x.x подключаться к любой базе данных, если они проходят проверку ident. Если же ident говорит, например, что это пользователь "bryanh" и он запрашивает подключение как пользователь Postgres Pro "guest1", подключение будет разрешено, если в

файле pg_ident.conf есть сопоставление "omicron", позволяющее пользователю "bryanh" подключаться как "guest1".

```
# TYPE    DATABASE        USER     ADDRESS          METHOD
host      all             all      192.168.0.0/16   ident map=omicron
```

Если для локальных подключений предусмотрены только эти три строки, они позволят локальным пользователям подключаться только к своим базам данных (базам данных с именами, совпадающими с именами пользователей баз данных), кроме администраторов или членов роли "support", которые могут подключиться к любой БД. Список имён администраторов содержится в файле $PGDATA/admins. Пароли запрашиваются в любом случае.

```
# TYPE    DATABASE        USER         ADDRESS          METHOD
local     sameuser        all                           md5
local     all             @admins                       md5
local     all             +support                      md5
```

Последние две строчки в примере выше могут быть объединены в одну:

```
# TYPE    DATABASE        USER                 ADDRESS          METHOD
local     all             @admins,+support                      md5
```

В столбце DATABASE могут указываться списки и имена файлов:

```
# TYPE    DATABASE                USER        ADDRESS          METHOD
local     db1,db2,@demodbs        all                          md5
```

1.1.2.3.2. Настройка Alpha.Reports.Base

Все файлы настроек имеют расширение *.json и находятся рядом с исполняемым файлом в каталоге установки /opt/Automiq/Alpha.Reports.Base

Для настройки Alpha.Reports.Base выполните следующие действия:

1. В каталоге установки Alpha.Reports.Base откройте с помощью следующей команды файл "appsettings.json".

sudo nano /opt/Automiq/Alpha.Reports.Base/appsettings.json

2. Отредактируйте файл конфигурации:

В разделе "DataBaseServer" укажите следующие параметры:

- В пункте "Address" введите IP-адрес базы данных или доменное имя "localhost".
  Пример: "Address": "localhost".
- В пункте "DataBase" введите имя базы данных
  Пример: "DataBase": "Reports".
- В пункте "Login" введите логин для базы данных. По умолчанию: "postgres".
  Пример: "Login": "postgres".
- В пункте "Password" введите пароль для базы данных, который Вы указали при настройке PostgreSQL.
  Пример: "Password": "postgres".

Пароль не должен быть пустым.

В разделе "LDAPSecurity" укажите следующие параметры:

› В пункте "BaseName" замените "AlphaSecurity" на "AstraSecurity":

Пример: "BaseName": "AstraSecurity".

› В пункте "Host" введите "IP-адрес компьютера, на котором установлен LDAP.

Пример: "Host": "127.0.0.1".

Остальные параметры оставьте по-умолчаню.

В списке "EntryPoints" нужно перечислить все адреса, по которым должен быть доступен базовый сервер, в том числе все сетевые карты компьютера (если их больше одной) – для CORS политики.

Обращаться к серверу нужно строго по IP-адресу, указанному в поле "Application" → "ServerAddress".
Значение «localhost» – не равно «127.0.0.1».

IP-адрес, указанный в поле "Modules" → "Alpha.Reports.Report.Server" должен совпадать с IP-адресом, указанным в поле "Application" → "ServerAddress".

3. Запустите/перезапустите службу "Alpha.Reports.Base" выполнив команду:

sudo systemctl restart alpha.reports.base.service

```
[root@localhost ~]# sudo systemctl restart alpha.reports.base.service
```

1.1.2.3.2.1. Пример файла конфигурации

```
{
    "Kestrel": {
        "EndPoints": {
            "HTTP": {
                "Url": "http://*:5000"
            }
        }
    },

    "Application": {
        "Name": "Alpha.Reports.Base",
        "ServerAddress": "http://127.0.0.1:5000",
        "LicenseFile": "license.lic",
        "BasePath": "",
        "RedirectToHTTPS": "false",
        "LdapServer": "",

        "DataBaseServer": {
            "Address": "localhost",
            "DataBase": "Reports",
            "Login": "postgres",
            "Password": "postgres"
        },

        "LDAPSecurity": {
            "Host": "127.0.0.1",
            "Port": "389",
            "BaseName": "AstraSecurity",
            "AppName": "Reports",
            "Rights": {
                "AdminRight": "Administration",
                "ViewRight": "View"
            }
        },

        "Modules": {
```

```
                "Alpha.Reports.Server": "http://127.0.0.1:5001"
            },

            "AlphaLicenseName": "",

            "DisableDataAccessAutoUpdate": "false",

            "AllowAllNIC": "false",
            "EntryPoints": [
                "http://127.0.0.1:5000",
                "http://192.168.0.1:5000",
                "http://somehostname:5000"
            ],

            "CleanupInterval": 43200
        }
    }
```

1.1.2.3.3. Настройка Alpha.Reports.Server

Если при формировании отчётов по расписанию необходимо отправлять их по списку электронных адресов, то требуется сконфигурировать работу сервера отчётов.

> Все файлы настроек имеют расширение *.json и находятся рядом с исполняемым файлом в каталоге установки /opt/Automiq/Alpha.Reports.Server

Для настройки Alpha.Reports.Server выполните следующие действия:

1. В каталоге установки Alpha.Reports.Server откройте с помощью следующей команды файл "appsettings.json".

> sudo nano /opt/Automiq/Alpha.Reports.Server/appsettings.json

```
[root@localhost ~]# sudo nano /opt/Automiq/Alpha.Reports.Server/appsettings.json
```

2. Отредактируйте файл конфигурации:

> Раздел "DataBaseServer" необходимо заполнить аналогично Alpha.Reports.Base.

В разделе "SMTP" заполните параметры:

- Address – адрес smtp-сервера.
- Port – порт для подключения, по умолчанию 25.
- Login – почтовый адрес (полный), с которого будет осуществляться рассылка.
- Password – пароль для подключения к smtp-серверу.

3. Запустите/перезапустите службу "Alpha.Reports.Server", выполнив команду:

sudo systemctl restart alpha.reports.server.service

```
[root@localhost ~]# sudo systemctl restart alpha.reports.server.service
```

1.1.2.3.3.1. Пример файла appsettings.json

```
{
    "Kestrel": {
        "EndPoints": {
            "HTTP": {
                "Url": "http://*:5001"
            }
        }
    },

    "Application": {
        "Name": "Alpha.Reports.Server",
        "BasePath": "",
        "RedirectToHTTPS": "false",

        "DataBaseServer": {
            "Address": "localhost",
            "DataBase": "Reports",
            "Login": "postgres",
            "Password": "postgres"
        },

        "Modules": {
            "Alpha.Base.Server": "http://127.0.0.1:5000"
        },

        "SMTP": {
            "Address": "smtp.example.com",
            "Port": "25",
            "Login": "alpha_reports@example.com",
            "Password":  "password"
        },

        "JobFile": "Alpha.Reports.Server.Jobs.xml"
    }
}
```

1.1.2.3.4. Секретная строка

Для работы с системой отчётности необходима секретная строка, которая генерируется при запуске "Alpha.Reports.Base".

Получить секретную строку можно любым из способов:

- Программно обратитесь к функции "GetAuthString()".
- В файле /var/log/Automiq/Alpha.Reports/Base.log найдите строку

<Дата> <Время>|Info|Alpha.Reports.Base|Строка авторизации:<Строка авторизации>

Пример

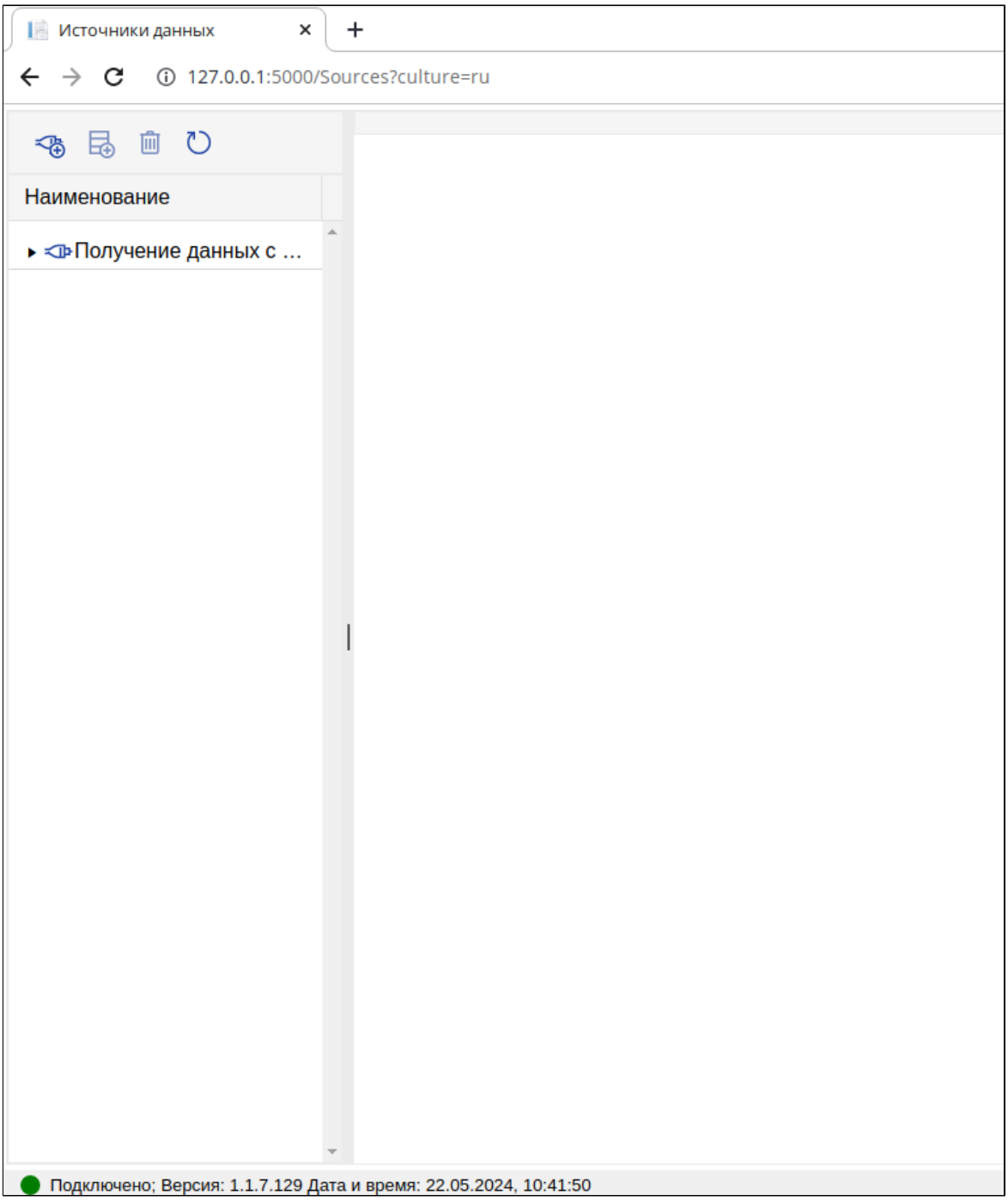
```
2024-05-22 13:37:15.0882|Info|Alpha.Reports.Base|Коннектор «Получение данных с OPC UA» успешно загружен
2024-05-22 13:37:15.0976|Info|Alpha.Reports.Base|Загрузка источников данных закончена
2024-05-22 13:37:15.2538|Error|Alpha.Reports.Base|Произошла ошибка при подключении к серверу отчетов|One or mo
2024-05-22 13:37:15.2614|Debug|Alpha.Reports.Base|Обновление блока [Application:DataBaseServer] в конфигурацио
2024-05-22 13:37:15.3788|Info|Alpha.Reports.Base|Строка авторизации: http://127.0.0.1:5000/?auth=38hrV1%2BTwJC
2024-05-22 13:38:57.8656|Debug|Базовый сервер|Alpha.Reports.Server 1.1.7.129+e4b5337cfda2a7543f3173b2c8b59da95
2024-05-22 13:38:57.8656|Info|Базовый сервер|Сервер отчетов подключился
2024-05-22 13:39:00.8547|Info|Alpha.Reports.Base|Выполнено подключение к серверу отчетов
```

1.1.2.3.5. Источники данных

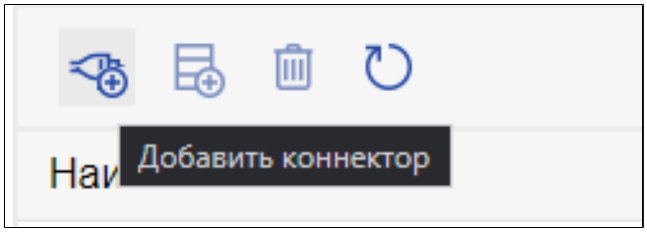
1. Чтобы открыть панель конфигурирования источников, в браузере введите:

http://127.0.0.1:5000/Sources/?u=<Секретная строка>

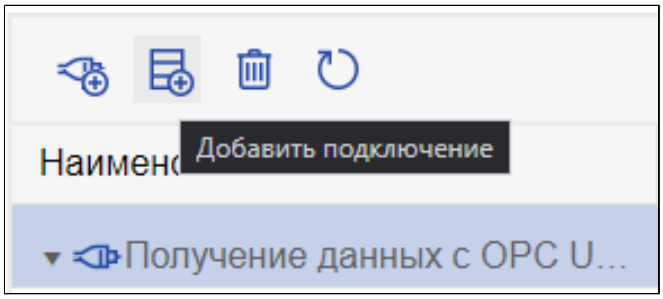
При успешном запуске в браузере отобразится окно:



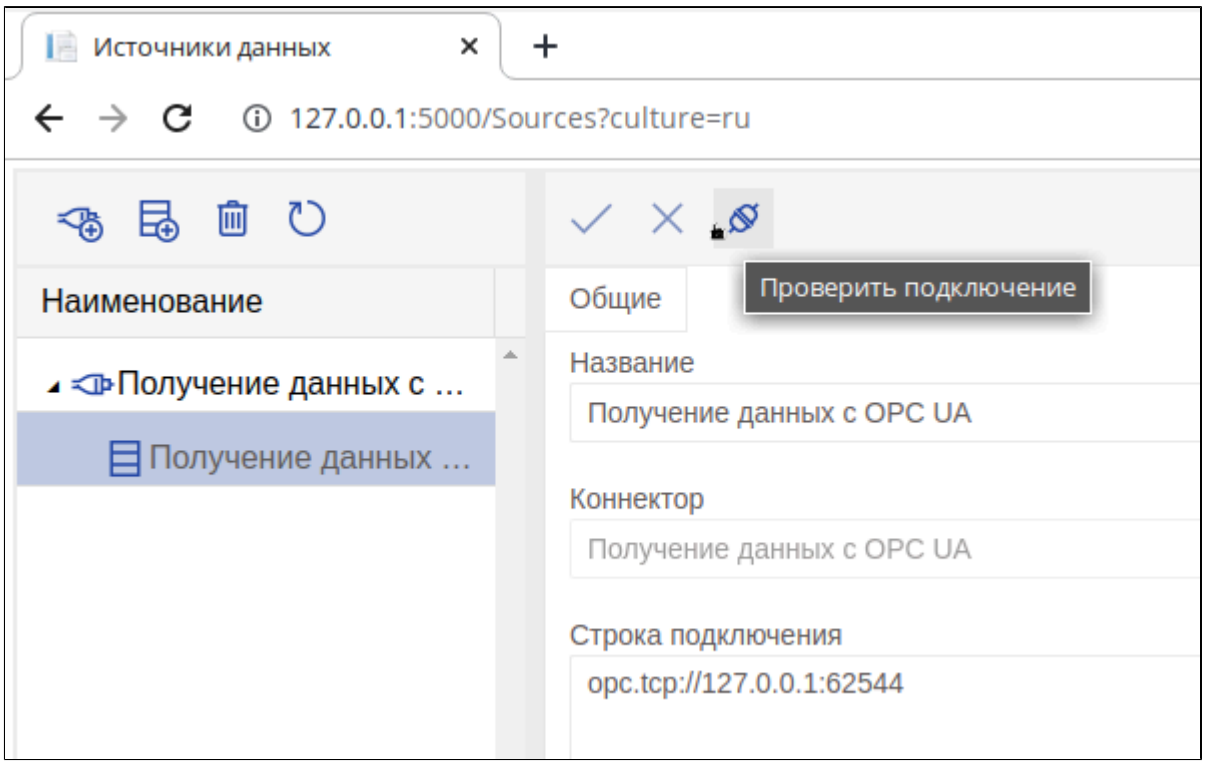
2. По умолчанию источники данных уже настроены и можно переходить к пункту 4. Если источники данных не настроены, нажмите кнопку "Добавить коннектор".



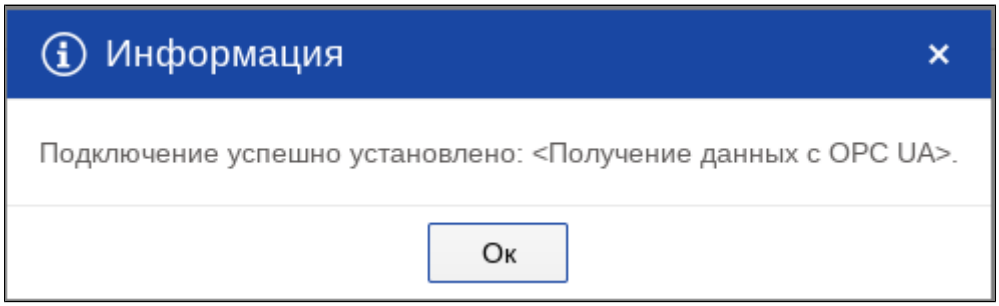
3. Выделите коннектор и нажмите кнопку «Добавить подключение».



4. Настройте строку подключения у коннектора и проверьте подключение.



При успешном подключении появится следующее окно:

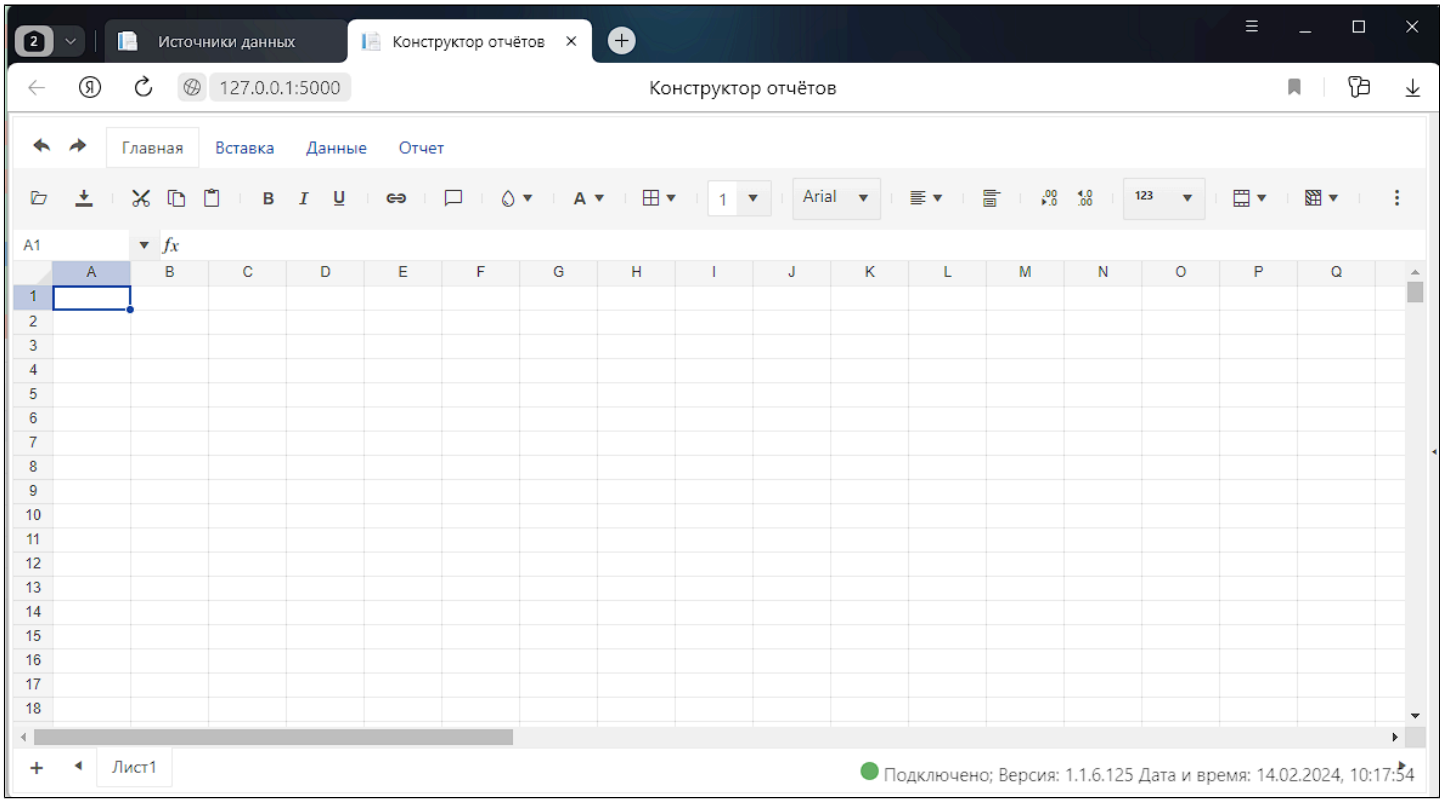

Информация
Подключение успешно установлено: <Получение данных с OPC UA>.
Ок

1.1.3. Дизайнер отчетов

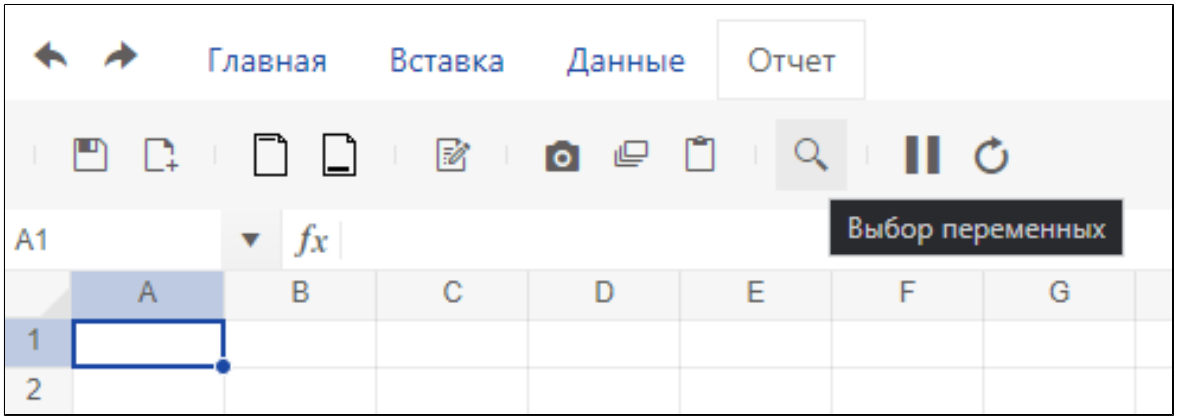
1. Чтобы открыть дизайнер отчетов, введите в браузере

> http://127.0.0.1:5000/Report/Designer/?u=<Секретная строка>

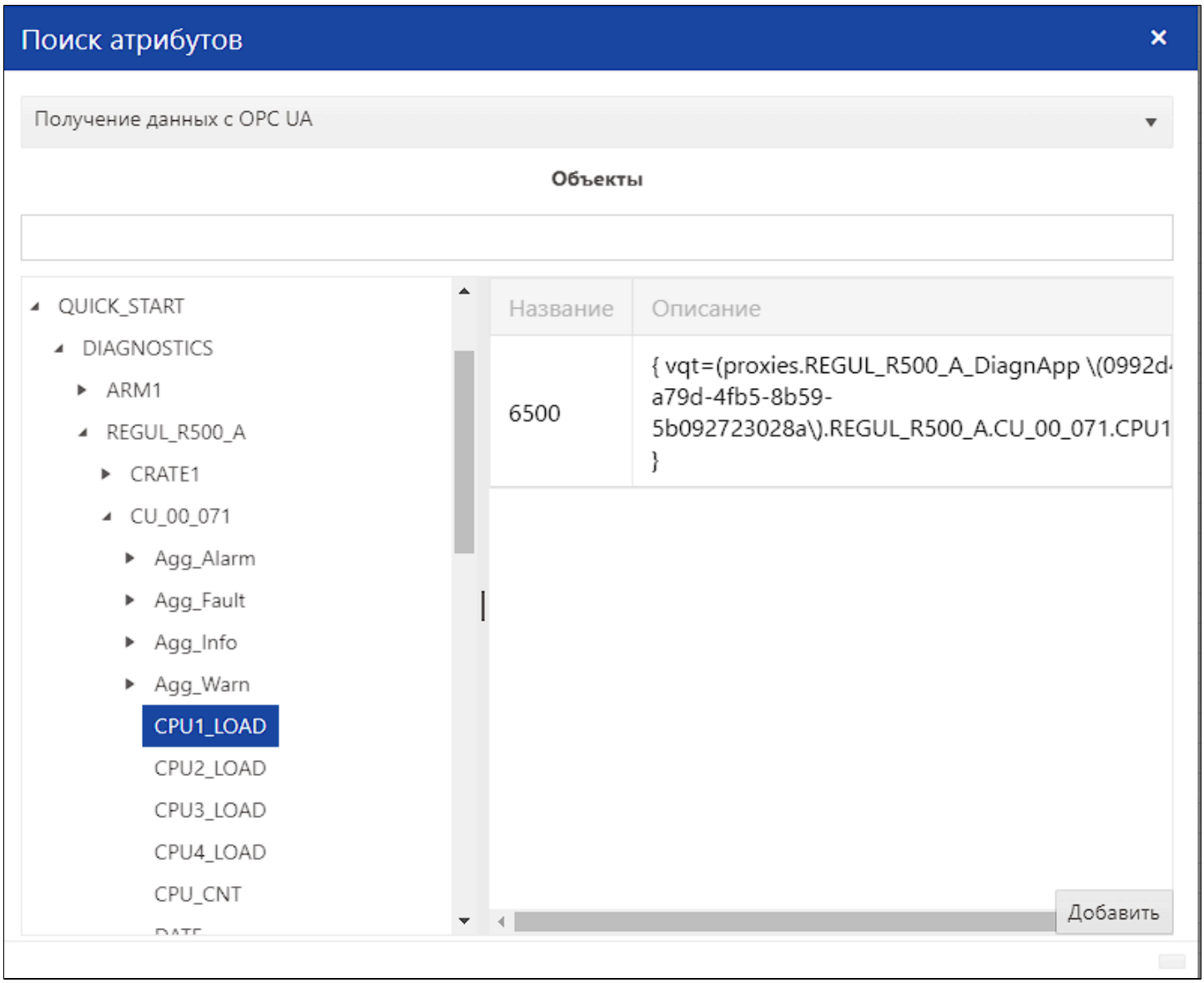
При успешном запуске в браузере отобразится окно:



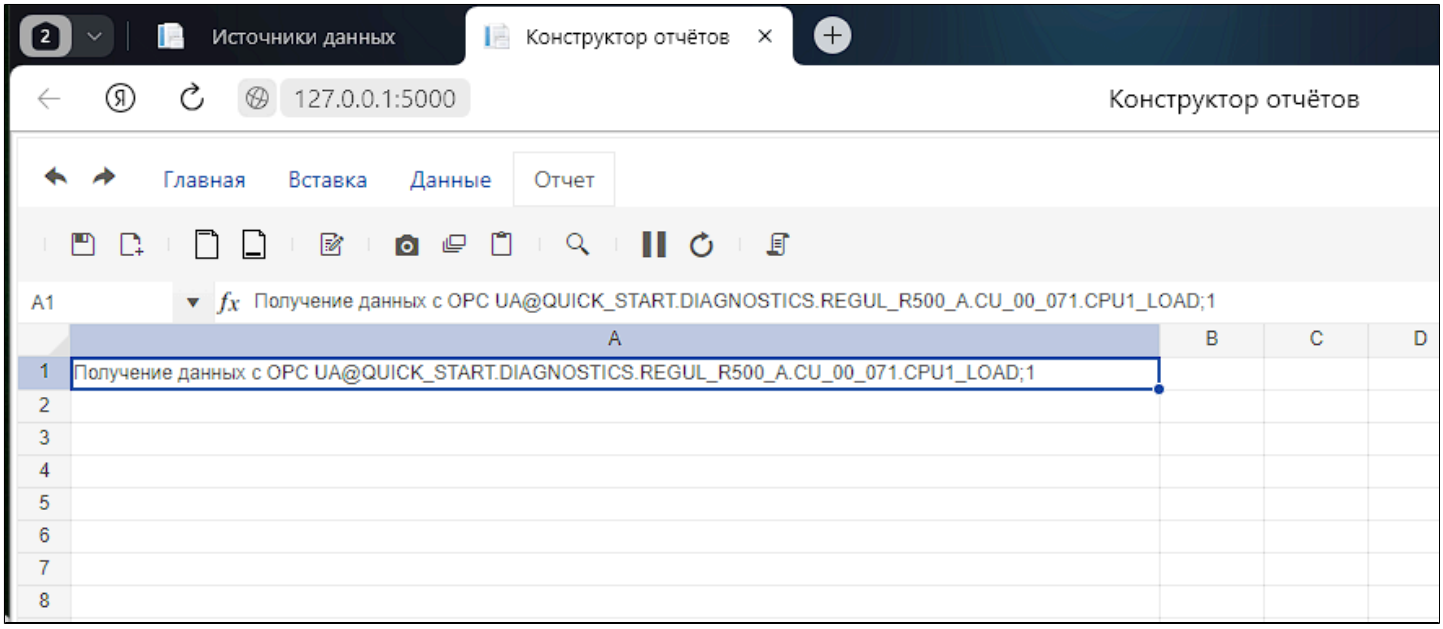
2. Для добавления сигналов, на основе которых будет создаваться отчет, перейдите во вкладку «Отчет» и нажмите кнопку «Выбор переменных».



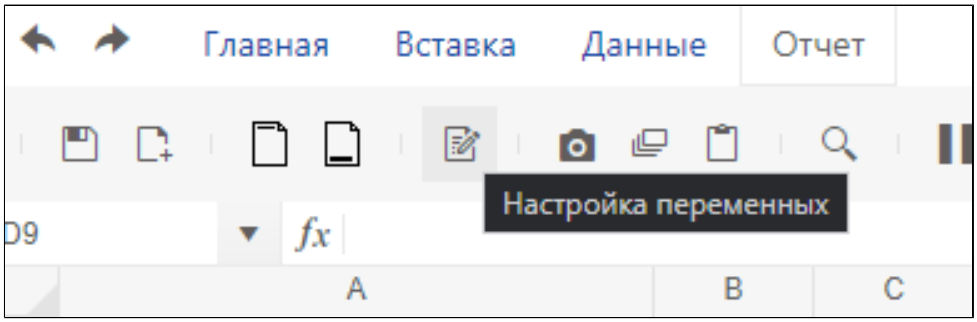
3. В выпадающем списке выберите "Получение данных с OPC UA". Выделите необходимый сигнал и нажмите кнопку «Добавить».



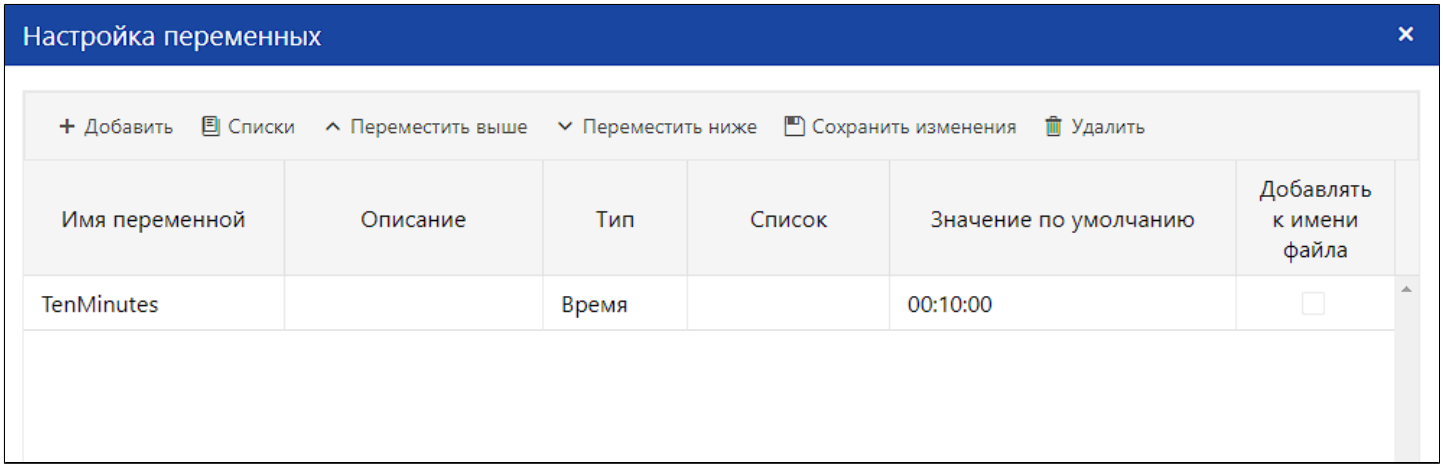
В выделенной ячейке таблицы появится атрибут сигнала.



4. Чтобы создать вспомогательные переменные для задания интервала времени, нажмите кнопку «Настройка переменных».

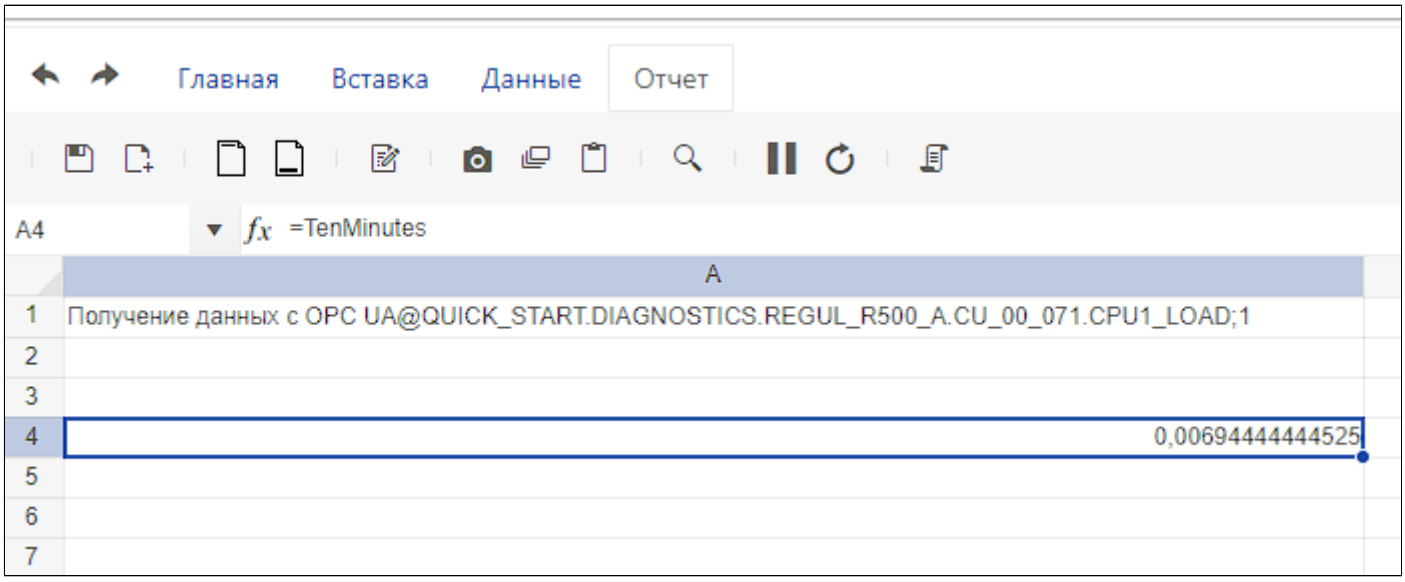


5. В окне нажмите кнопку "Добавить" и укажите имя переменной, её тип и значение по умолчанию.



В нашем случае тип время, а значение по умолчанию 10 минут. Это позволит далее создать отчет со значениями за последние 10 минут.

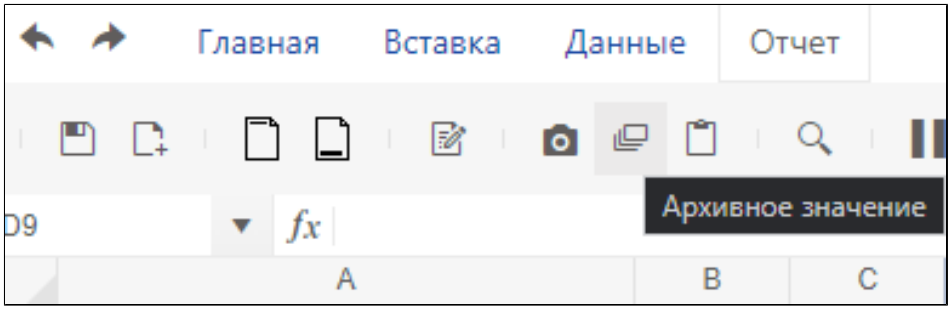
6. Выделите произвольную ячейку и введите «=<название переменной>». Это позволит проверить, что переменная добавилась в шаблон и считывается.



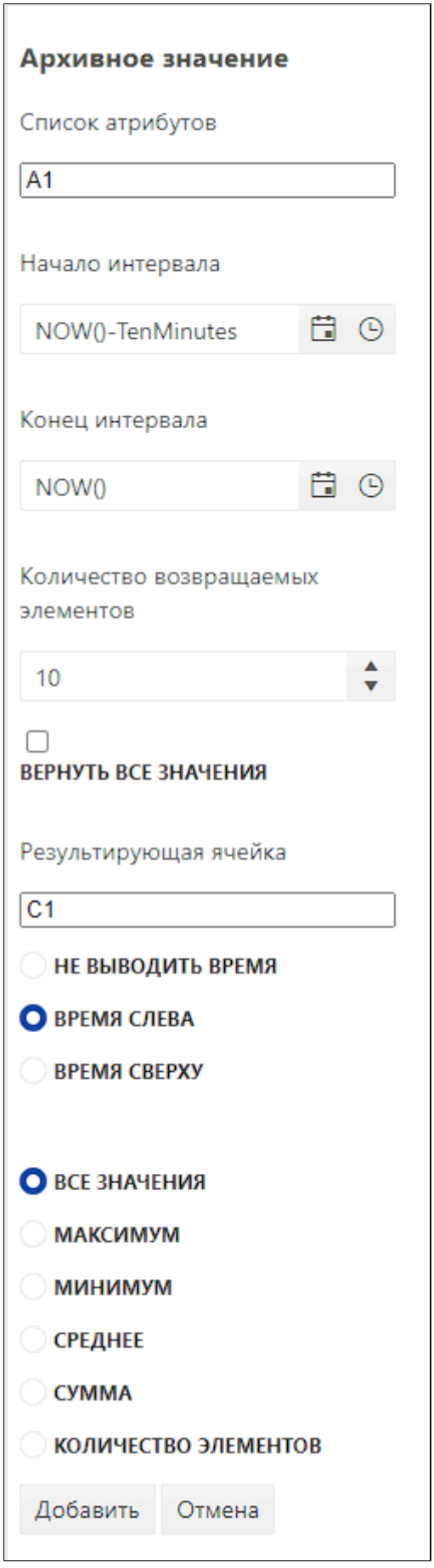
1.1.3.1. Создание отчета

Для формирования отчета, выполните следующие действия:

1. Нажмите кнопку «Архивное значение».



2. Справа появится поле для конфигурирования архивных значений:


Архивное значение
Список атрибутов
A1
Начало интервала
NOW()-TenMinutes
Конец интервала
NOW()
Количество возвращаемых элементов
10
ВЕРНУТЬ ВСЕ ЗНАЧЕНИЯ
Результирующая ячейка
C1
НЕ ВЫВОДИТЬ ВРЕМЯ
ВРЕМЯ СЛЕВА
ВРЕМЯ СВЕРХУ
ВСЕ ЗНАЧЕНИЯ
МАКСИМУМ
МИНИМУМ
СРЕДНЕЕ
СУММА
КОЛИЧЕСТВО ЭЛЕМЕНТОВ
Добавить
Отмена

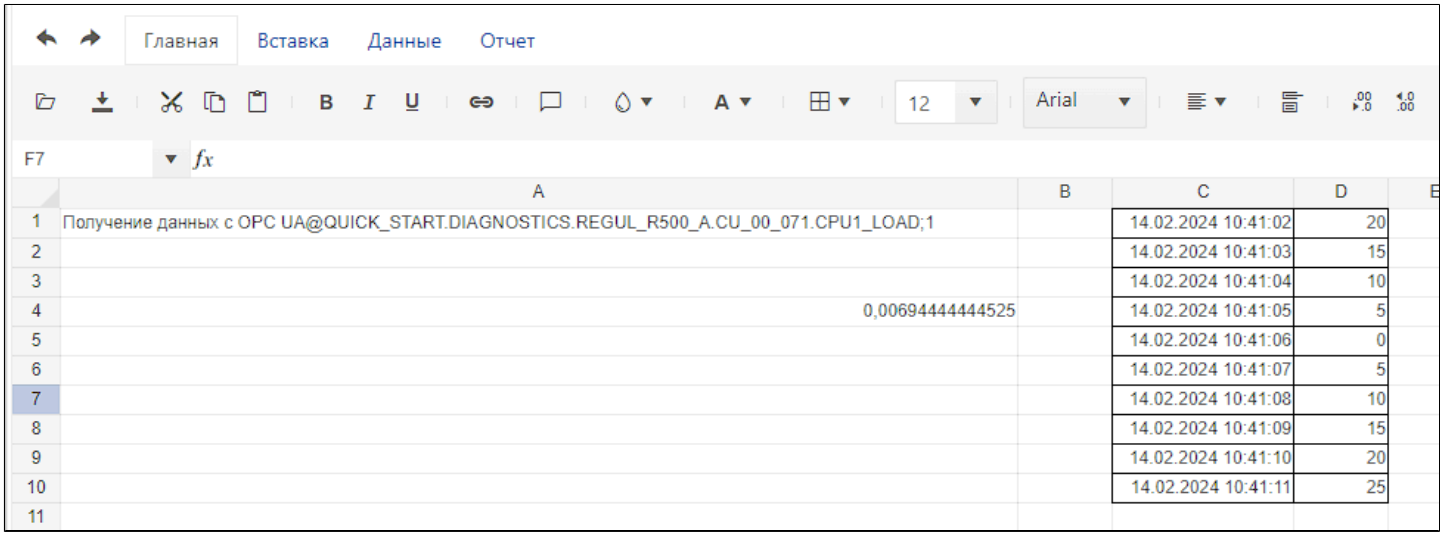
- В поле "Список атрибутов" укажите ячейку, в которую Вы добавили атрибуты с помощью кнопки «Выбор переменных».
- В поле "Начало интервала" введите формулу: NOW()-TenMinutes. Она вычисляет текущее значение времени и вычитает из него значение переменной, которую мы создали ранее.
- В поле "Конец интервала" введите функцию NOW().
- Укажите "Количество возвращаемых элементов" или установите флаг "Вернуть все значения".
- В поле «Результирующая ячейка» укажите ячейку, с которой начнет формироваться отчет.

> При желании отметьте сторону для вывода времени и какие значения Вы хотите вывести.

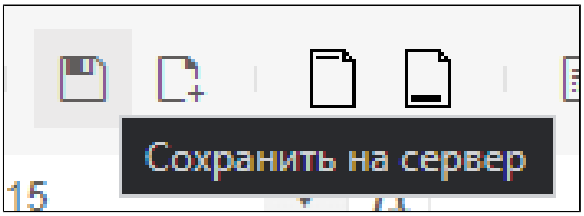
В результате сформируется таблица:

C	D
14.02.2024 10:41:02	20
14.02.2024 10:41:03	15
14.02.2024 10:41:04	10
14.02.2024 10:41:05	5
14.02.2024 10:41:06	0
14.02.2024 10:41:07	5
14.02.2024 10:41:08	10
14.02.2024 10:41:09	15
14.02.2024 10:41:10	20
14.02.2024 10:41:11	25

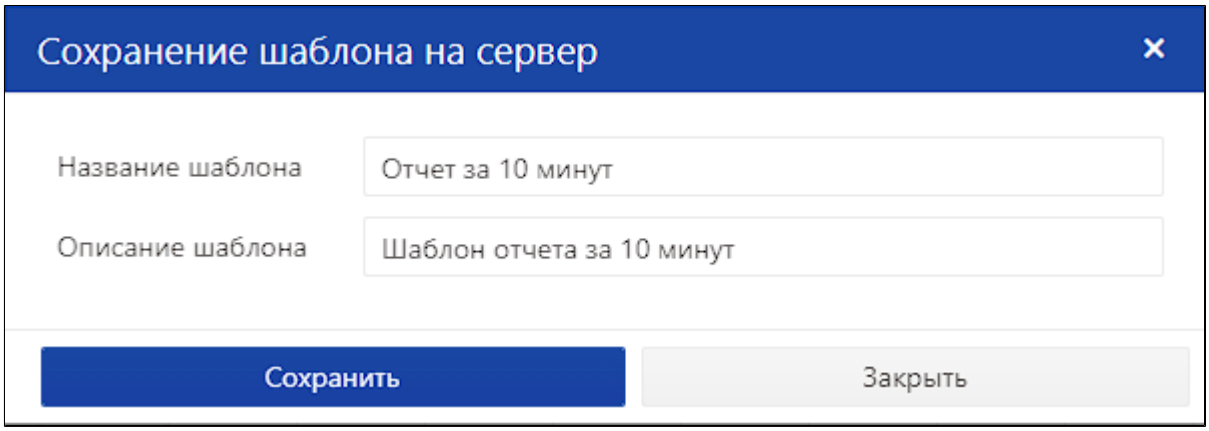
3. Настройки отображения таблицы настраиваются во вкладке "Главная".



4. Сохраните отчет, нажав кнопку «Сохранить на сервер» во вкладке "Отчет", или сохраните отчет как Excel-таблицу.



5. Введите название шаблона и его описание. Нажмите кнопку "Сохранить"



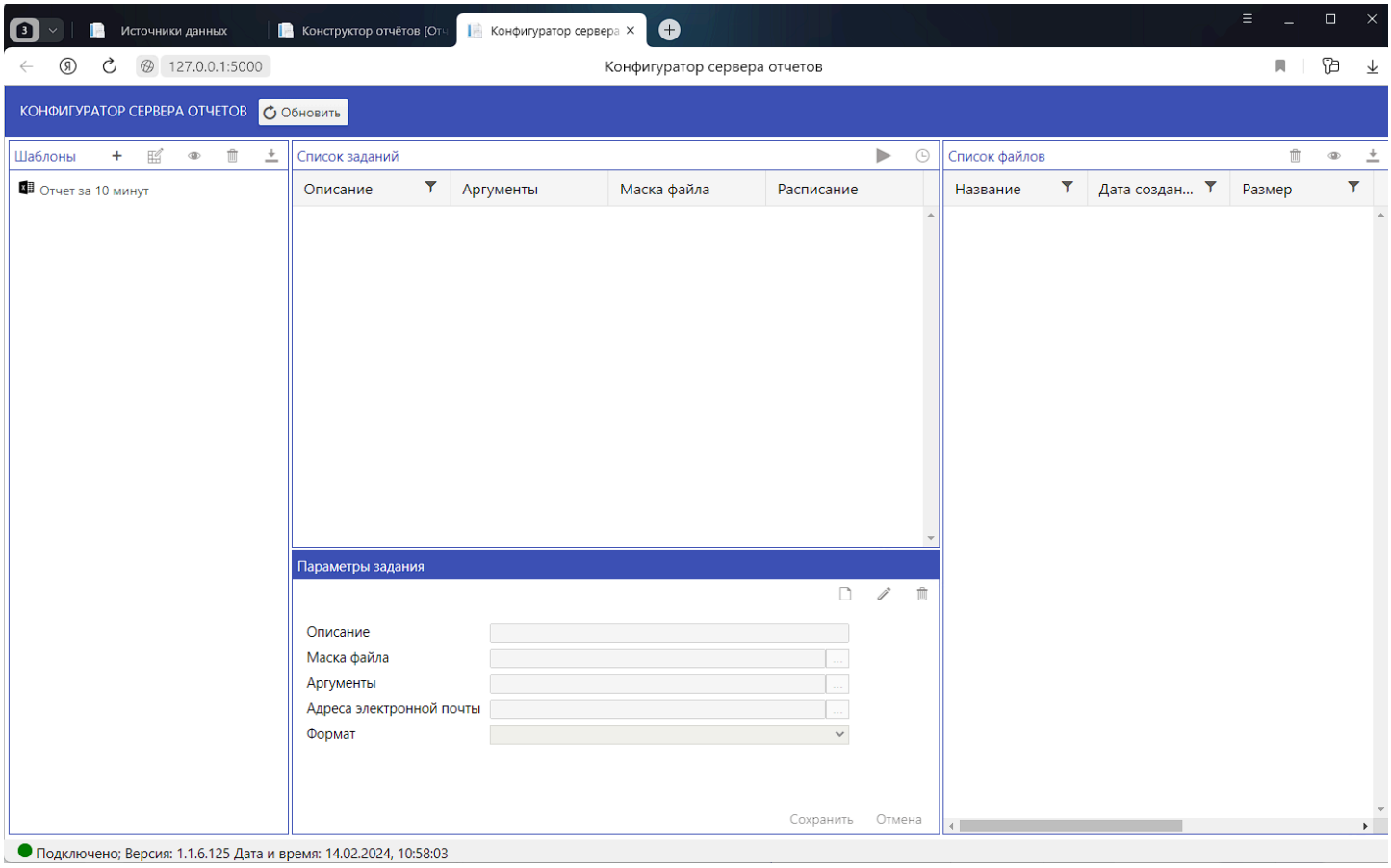
Подробнее вкладка «Дизайнер отчетов» описана в документе «Конструктор отчётов. Руководство пользователя».

1.1.4. Конфигуратор сервера отчетов

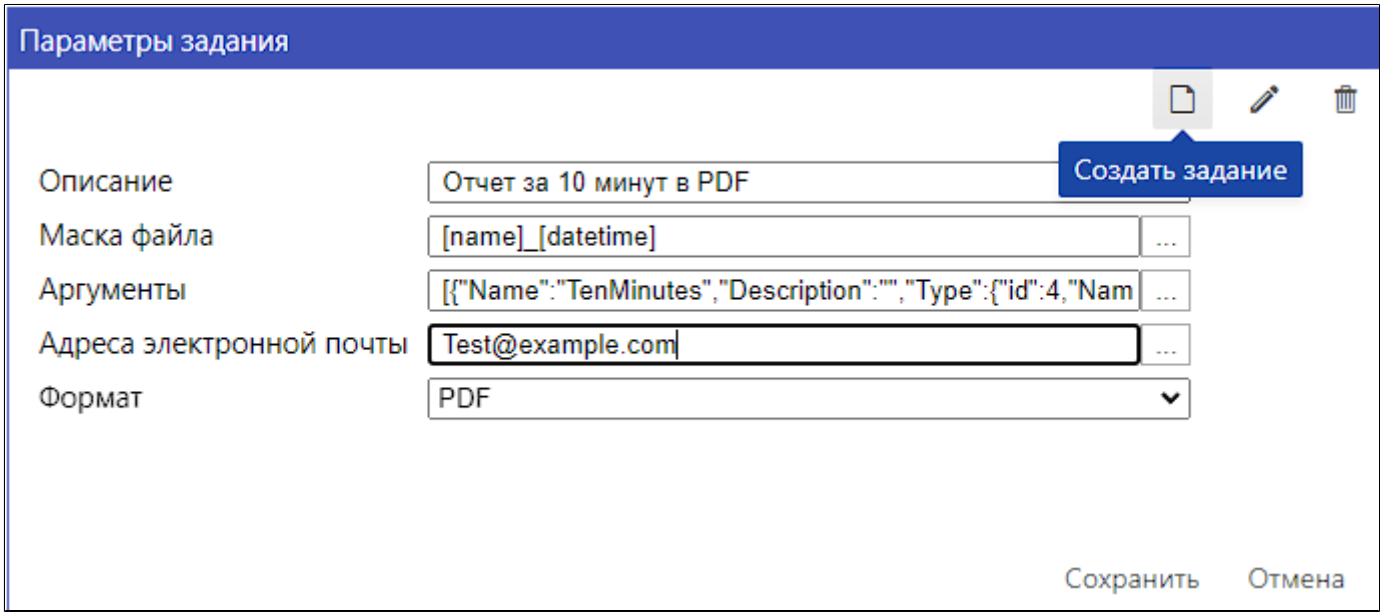
1. Чтобы открыть конфигуратор отчетов, введите в браузере:

http://127.0.0.1:5000/Report/Configurator/?u=<Секретная строка>

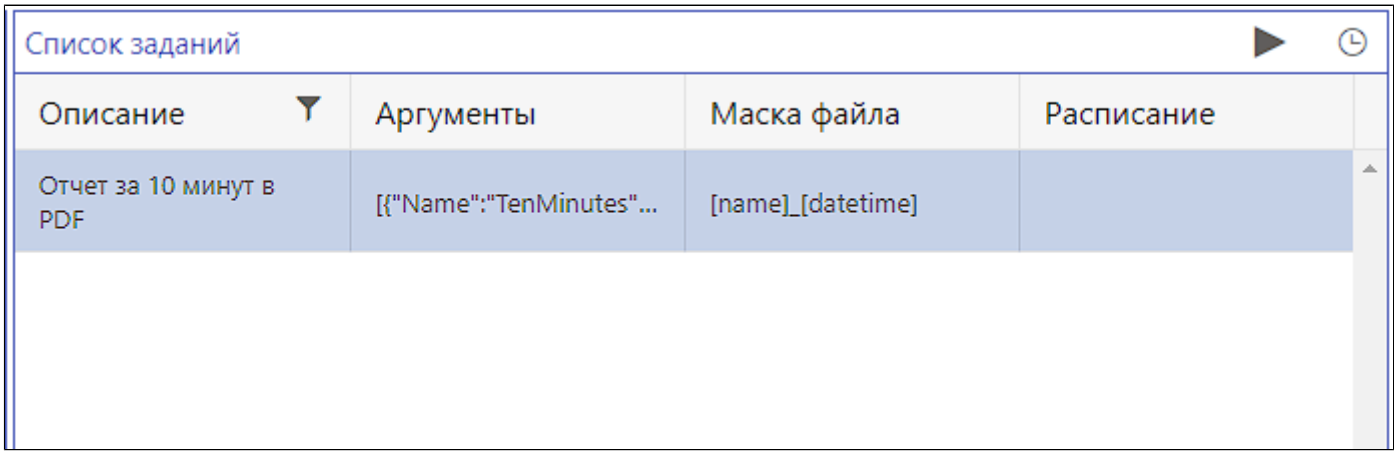
При успешном запуске в браузере отобразится окно:



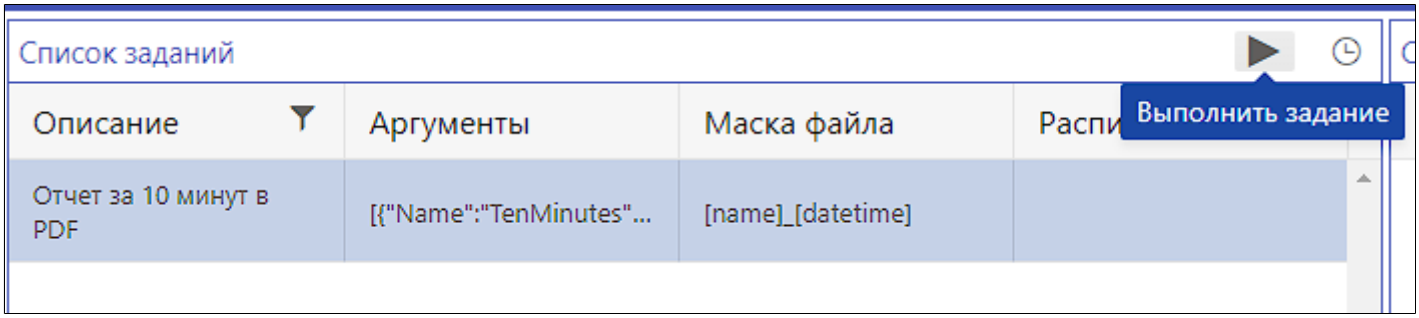
2. В разделе "Шаблоны" выберите созданный ранеее шаблон и перейдите в раздел "Параметры задания":



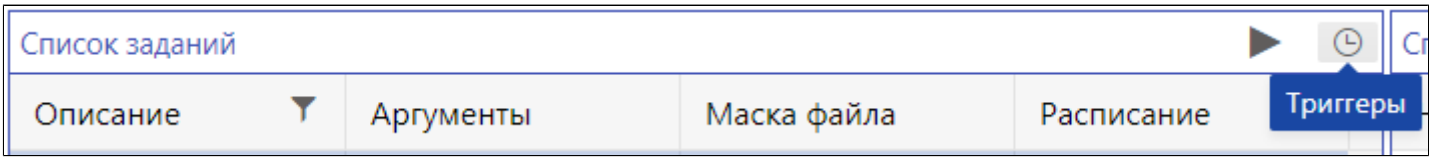
3. Нажмите «Создать задание», введите его параметры и нажмите кнопку «Сохранить». Задание появится во вкладке «Список заданий».



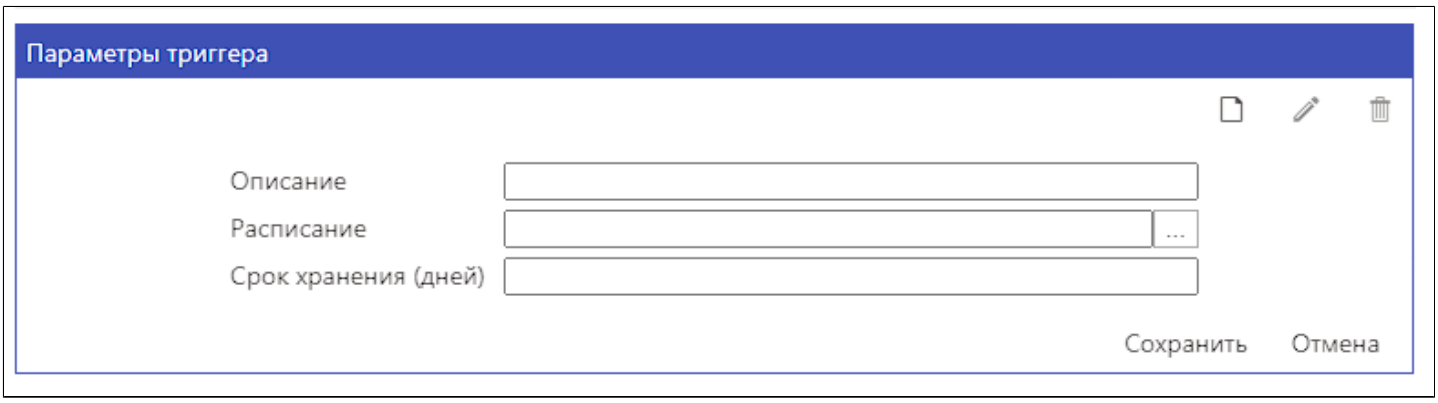
4. Сформируйте отчет, нажав кнопку «Выполнить задание».



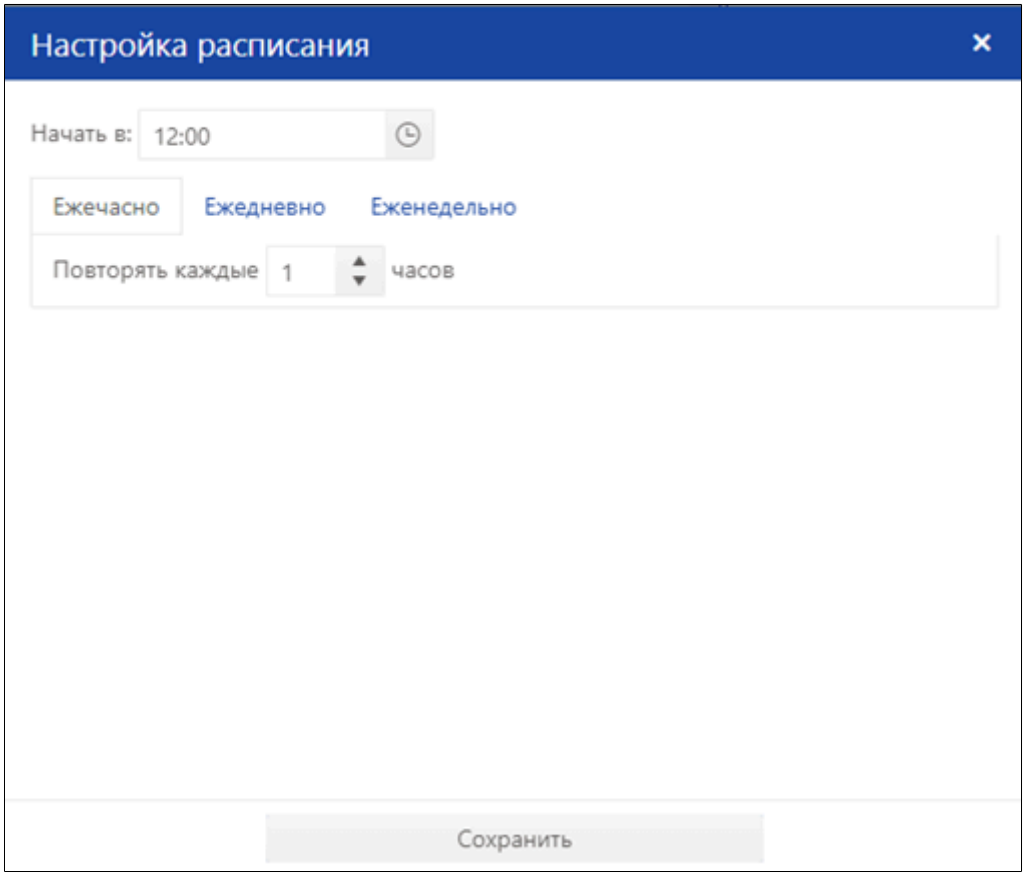
5. Чтобы создать расписание для автоматического формирования отчета, нажмите кнопку «Триггеры».



6. В открывшемся окне нажмите кнопку «Создать триггер» и настройте расписание.

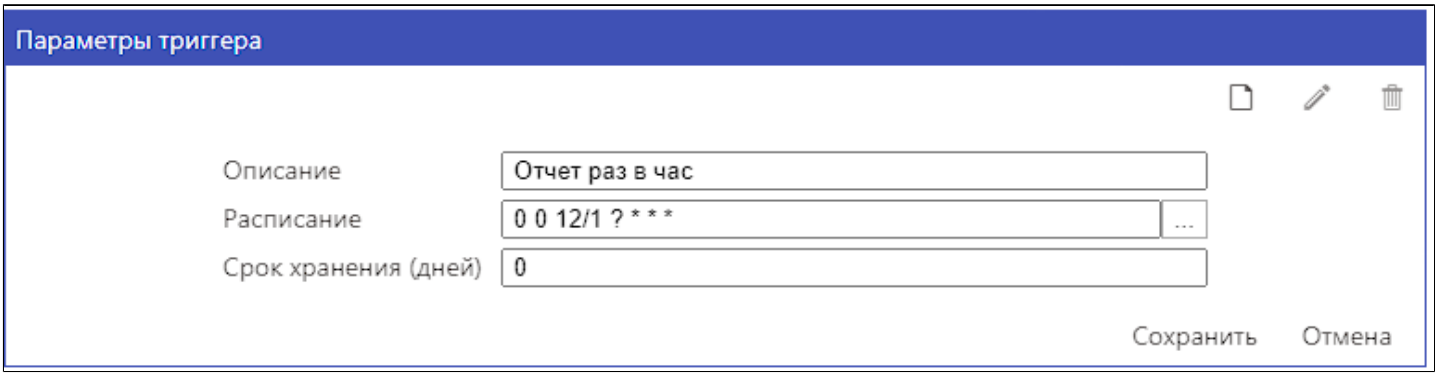


7. Для настройки расписания нажмите на кнопку в поле расписания. В открывшемся окне настройте периодичность и время, когда триггер начнет свою работу после создания.

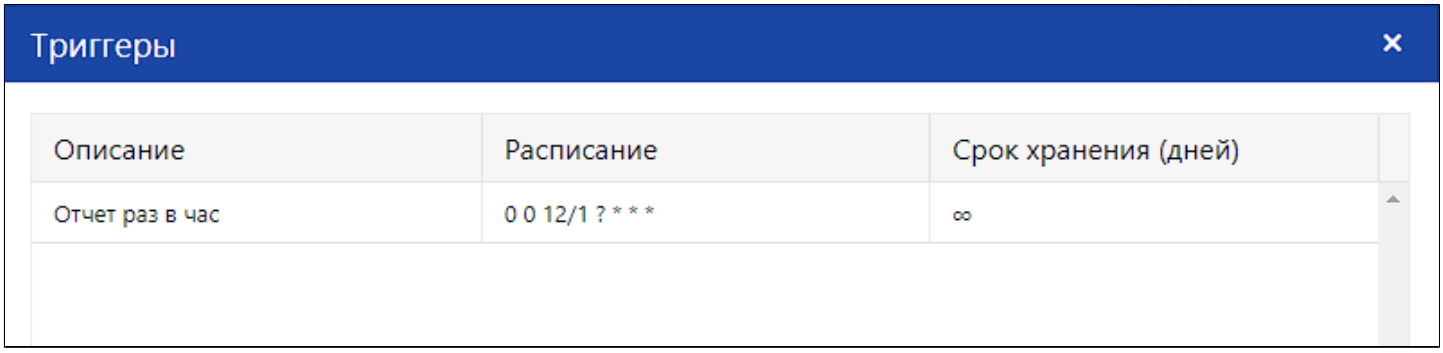


8. В поле "Срок хранение (дней)" укажите количество дней хранения отчёта.

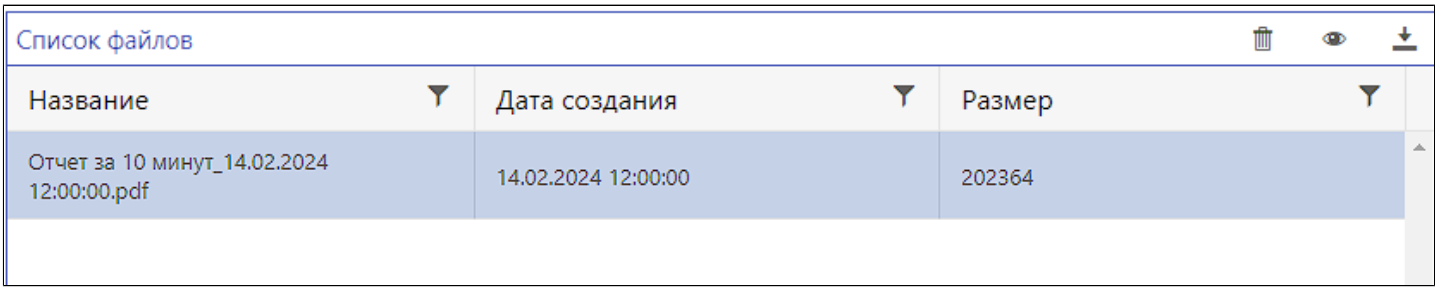
При значении «0» – отчёт удаляться не будет.



9. Нажмите кнопку «Сохранить». Триггер с расписанием начнет свою работу.



10. При срабатывании триггера в списке файлов появится файл отчета.



Вы можете удалить, просмотреть и скачать отчет нажав на соответствующие кнопки в правом верхнем углу.

Подробнее вкладка «Конфигуратор отчетов» описана в документе «Конфигуратор сервера отчётов. Руководство пользователя».

Настройка просмотров отчетов описана в документе «Просмотр отчётов. Руководство пользователя».