Arenadata™ Database

Версия - v6.21.0_arenadata35

Arenadata DB Command Center

2022 Arenadata™

Оглавление

1 Системные требования **3**

2 Руководство пользователя **4**

2.1 Query Monitor . . . 4

2.1.1 Информация о запросах . . . 4

2.1.2 Отмена выполнения запроса . . . 6

2.2 History . . . 6

2.3 Query Details . . . 9

2.3.1 Информация о запросе . . . 9

2.3.2 Отмена выполнения запроса . . . 12

3 Администрирование **14**

3.1 Уровни доступа . . . 14

3.2 Аутентификация . . . 14

3.2.1 Стандартная аутентификация . . . 15

3.2.2 LDAP-аутентификация . . . 15

3.3 Управление пользователями . . . 16

3.3.1 Стандартные пользователи . . . 16

3.3.2 Управление доступом для LDAP-групп . . . 16

3.3.3 Настройка SSL в ADCC . . . 17

3.4 Подключение внешнего ADCC и мониторинг нескольких кластеров . . . 20

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io

Глава 1

Системные требования

Для использования **ADCC** необходимо клиентское место, удовлетворяющее следующим техническим требованиям:

- CPU: 2 CPU Cores
- RAM: 4G
- HDD: 8G free space

Использовать **ADCC** следует в одном из следующих браузеров с минимальной версией:

- *Chrome* v. 88
- *Safari* v. 13.1.3
- *Yandex браузер* v. 22.3.3.888

Глава 2

Руководство пользователя

- *Query Monitor*
- *History*
- *Query Details*

2.1 Query Monitor

Страница *Query Monitor* отображает доступную для просмотра текущему пользователю **ADCC** информацию о запросах, выполняемых в данный момент времени пользователями **ADB** (Рис.2.1.). На странице отображаются выполняющиеся (Running), находящиеся в очереди ожидания исполнения (Queued), а также заблокированные запросы (Blocked). В зависимости от присвоенных прав доступа пользователю доступна отмена выполнения некоторых или всех запросов в списке.

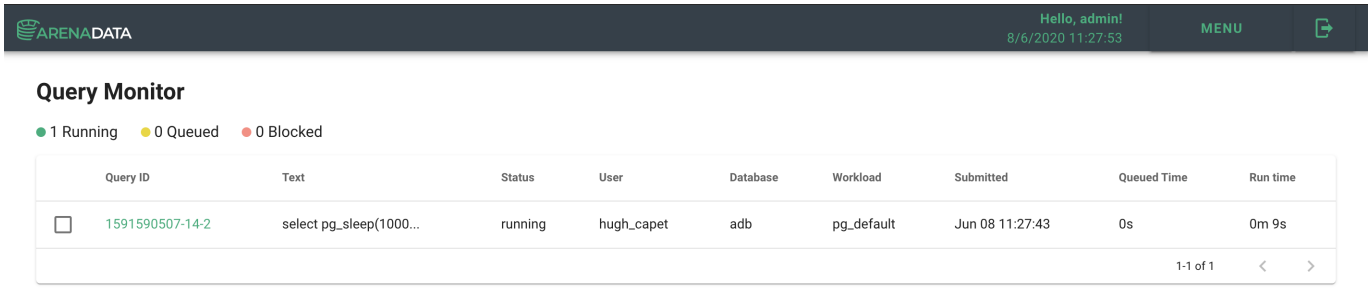


Рис.2.1.: Query Monitor

2.1.1 Информация о запросах

Для каждого запроса в списке указывается:

- **Query ID** – уникальный идентификатор запроса, например: `1590901567-16-2`. Значение является ссылкой на страницу с *расширенной информацией о запросе*. Идентификатор формируется из значений параметров *tmid*, *ssid* и *ccnt* для каждого запроса:
  - *tmid* – временная метка старта кластера;
  - *ssid* – id сессии;
  - *ccnt* – номер команды в рамках сессии.

- **Text** – первые 20 символов текста запроса. При наведении текст отображается в полном виде;
- **Status** – статус запроса, принимает одно из следующих значений:
  - *Queued* – запрос находится в очереди ожидания исполнения;
  - *Running* – запрос выполняется;
  - *Blocked* – выполнение запроса заблокировано ожиданием получения ресурсной очереди или слота ресурсной группы; блокировкой, вызванной синхронной репликацией или удерживаемой другой транзакцией на объекте базы данных (lock);
  - *Cancelling* – отправлен запрос на отмену выполнения.

  В случае, если статус *Blocked* вызван блокировкой на объекте базы данных, удерживаемой другой транзакцией, нажатие на пиктограмму справа от значения в поле *status* открывает окно с дополнительной информацией (Рис.2.2.). В окне содержится перечисление блокирующих запросов, для каждого из которых указывается:
  - *queryId* – уникальный идентификатор. Значение параметра *ssid* запроса, выполняющегося в режиме *utility*, равно *-1*. В некоторых случаях, когда невозможно определить номер команды в рамках сессии, параметр *ccnt* содержит значение *X*;
  - *target* – информация об объекте, на котором удерживается блокировка;
  - *segments* – идентификатор сегмента, на котором удерживается блокировка;
  - *user* – пользователь, выполнивший запрос;
  - *blocked for* – длительность блокировки.

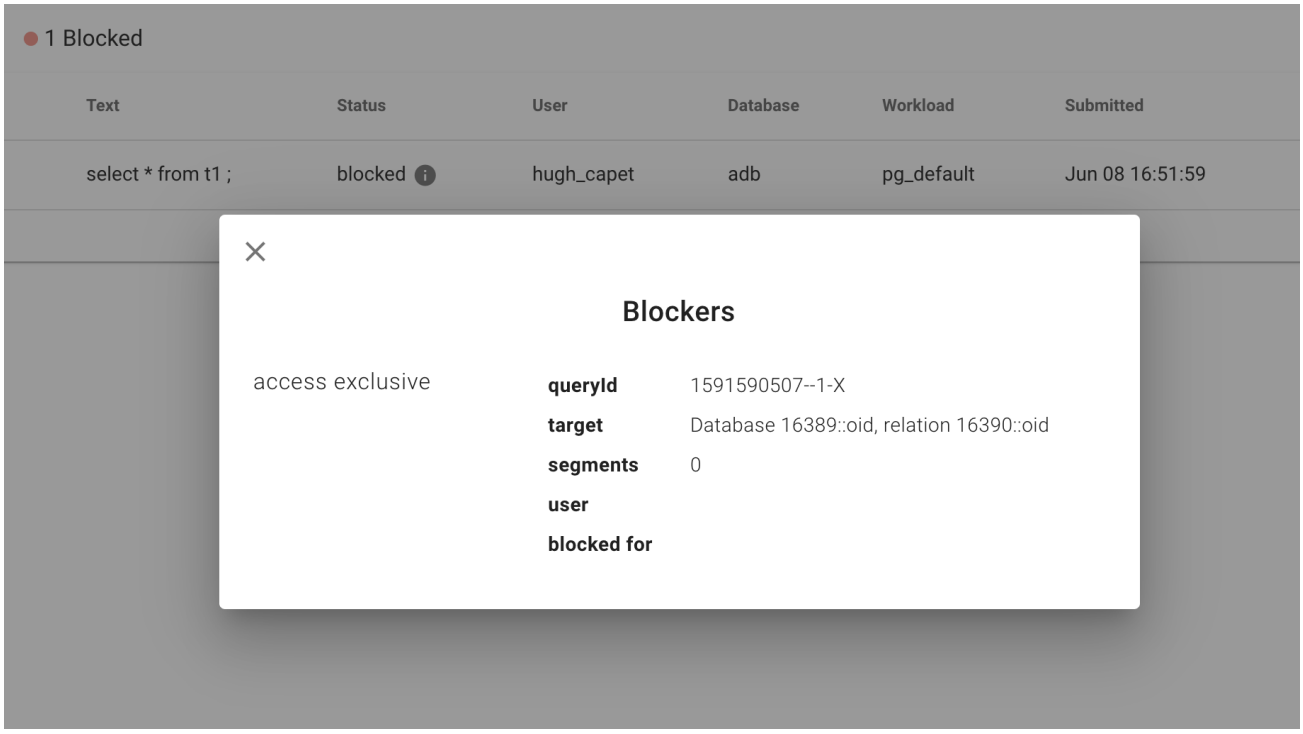


Рис.2.2.: Дополнительная информация о блокировке

Если выполнение запроса заблокировано по другим причинам, дополнительной информации не предоставляется.

- **User** – пользователь ADB, выполнивший запрос;
- **Database** – название базы данных, в которой был выполнен запрос;
- **Workload** – название ресурсной группы или ресурсной очереди, управляющей данным запросом;
- **Submitted** – время создания запроса;
- **Queued time** – длительность нахождения запроса в очереди ожидания исполнения;
- **Run time** – длительность исполнения запроса.

2.1.2 Отмена выполнения запроса

ADCC предоставляет возможность отмены запросов, выполняющихся в кластере **ADB**. Для отмены одного или нескольких запросов необходимо (Рис.2.3.):

1. Выбрать при помощи чекбоксов в начале строк таблицы все необходимые запросы (выбор доступен только для запросов, отмена которых разрешена, исходя из прав доступа, определенных для пользователю);
2. Выбрать способ отмены запроса и нажать соответствующую кнопку справа над таблицей:
   - *Cancel* – отмена выполнения запроса путем вызова функции `pg_cancel_backend`, отменяющей запрос в обслуживающем процессе;
   - *Terminate* – отмена выполнения запроса путем вызова функции `pg_terminate_backend`, завершающей обслуживающий процесс, в котором выполняется запрос.

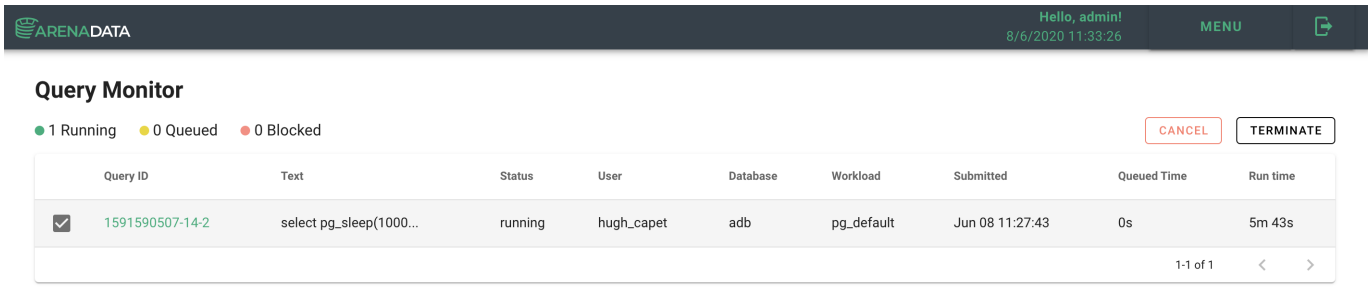


Рис.2.3.: Выбор запроса для отмены выполнения

2.2 History

На странице *History* можно получить информацию о запросах, выполнявшихся в кластере **ADB** в заданный период времени (Рис.2.4.).

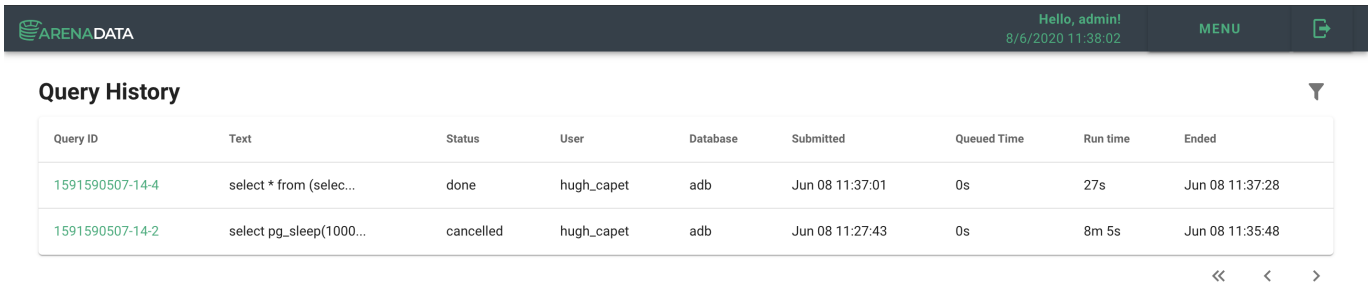


Рис.2.4.: Query History

Панель фильтров, позволяющая ограничивать вывод истории запросов, открывается при нажатии на иконку в правом верхнем углу экрана. Доступны следующие опции (Рис.2.5.):

- Ограничение периода времени, в который был создан запрос, по дате (Рис.2.6.) и времени (Рис.2.7.);
- Подстрока, которая должна содержаться в тексте запроса.

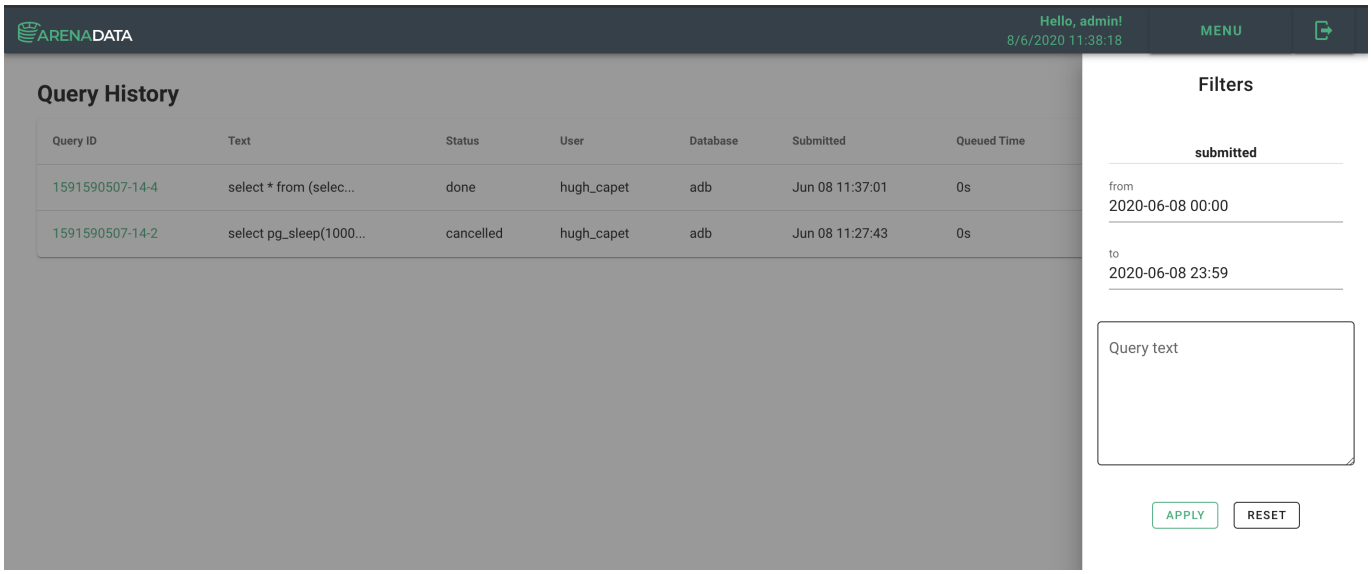


Рис.2.5.: Панель фильтров для истории запросов

Для каждого запроса в списке указывается:

- **Query ID** – уникальный идентификатор запроса, например: `1590901567-16-2`. Значение поля является ссылкой на страницу с *расширенной информацией о запросе*. Идентификатор формируется из значений параметров *tmid*, *ssid* и *ccnt* для каждого запроса:
  - *tmid* – временная метка старта кластера;
  - *ssid* – id сессии;
  - *ccnt* – номер команды в рамках сессии.
- **Text** – первые 20 символов текста запроса. При наведении текст отображается в полном виде;
- **Status** – итоговый статус запроса, принимает одно из следующих значений:
  - *Done* – запрос успешно выполнен;
  - *Cancelled* – выполнение запроса отменено;
  - *Error* – во время исполнения запроса произошла ошибка.
- **User** – пользователь ADB, выполнивший запрос;
- **Database** – название базы данных, в которой был выполнен запрос;
- **Submitted** – время создания запроса;
- **Queued time** – длительность нахождения запроса в очереди ожидания исполнения;
- **Run time** – длительность исполнения запроса;
- **Ended** – время завершения исполнения запроса: как в результате успешного выполнения, так и в следствие ошибки или отмены выполнения.

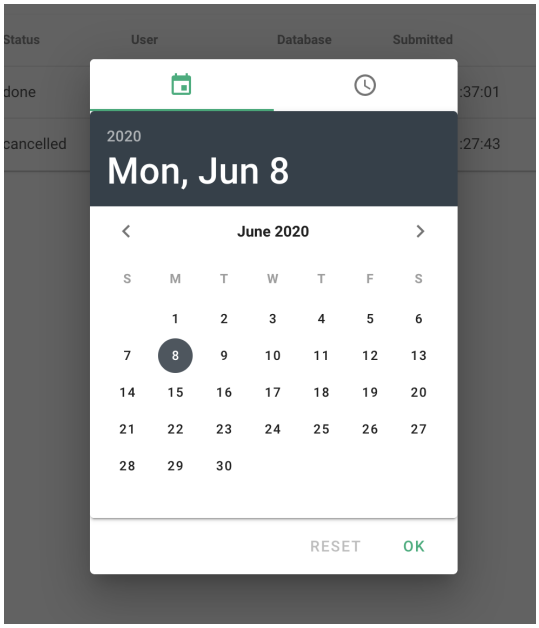


Рис.2.6.: Установка даты начала/конца периода

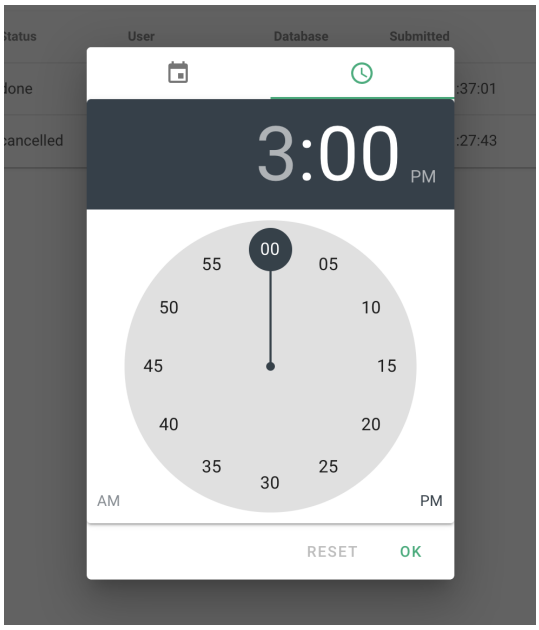


Рис.2.7.: Установка времени начала/конца периода

2.3 Query Details

Страница *Query Details* содержит расширенную информацию о выполняющемся запросе или об одном из запросов истории (Рис.2.8.).

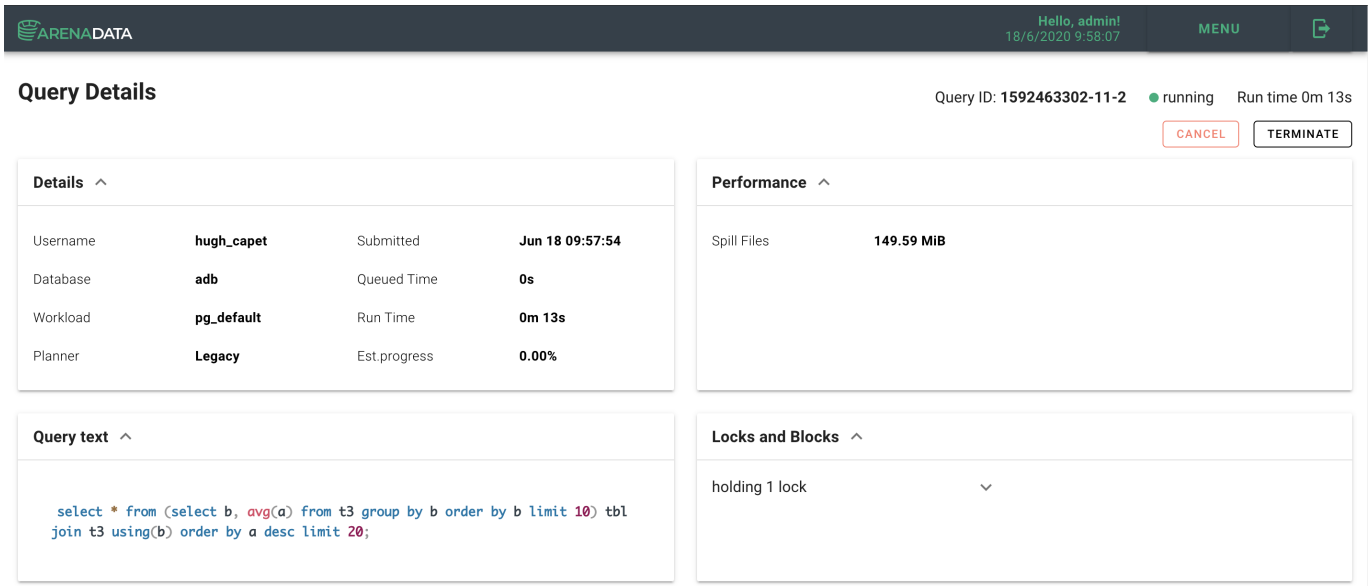


Рис.2.8.: Query Details - выполняющийся запрос

2.3.1 Информация о запросе

В верхней части страницы отображается текущий статус и длительность исполнения запроса, а также следующая информация:

Details

- **User** – пользователь ADB, выполнивший запрос;
- **Database** – название базы данных, в которой выполнен запрос;
- **Workload** – название ресурсной группы или ресурсной очереди, управляющей данным запросом;
- **Planner** – планировщик запросов, используемый для данного запроса: Legacy или GPORCA;
- **Queued Time** – длительность нахождения запроса в очереди ожидания исполнения;
- **Submitted** – время создания запроса;
- **Run Time** – длительность исполнения запроса;
- **Est. progress** – оценка прогресса выполнения запроса в процентах. Значение рассчитывается на основании оценок стоимости (коэффициента трудоемкости выполнения) и прогресса для каждого узла плана запроса:

$$NodeProgress = \frac{Actual\ Rows}{Estimated\ Rows}$$

Где

– *NodeProgress* – прогресс узла;

– *ActualRows* – реальное количество извлеченных строк (кортежей);

– *EstimatedRows* – предположение планировщика о количестве строк (кортежей), которое будет извлечено для данного узла.

Тогда общий прогресс, рассчитанный на основании суммы оценок для всех узлов плана:

$$OverallProgress = \frac{\Sigma(NodeProgress(x) * Est.Cost(x))}{\Sigma\, Est.Cost(x)}$$

Где *Est.Cost(x)* – оценка стоимости узла, сделанная планировщиком.

Оценка прогресса не является точной, так как при расчетах во внимание не принимается множество дополнительных факторов, влияющих на процесс исполнения запроса.

Performance

Метрики по производительности опрашиваются агентами на сегментах с периодичностью раз в 5 секунд, таким образом запросы, которые успевают завершиться быстрее этого времени, не получат метрик по производительности. В этом блоке отображаются как метрики по каждому сегменту, включая мастер-сегмент seg-1 (segment_id = -1), так и средние значения с “перекосами” по кластеру.

Объем spill-файлов

ADB создает spill-файлы, если доступной памяти ОС оказывается недостаточно. Эта метрика меняется в зависимости от статуса запроса.

Для исполняющегося запроса в блоке отражено единственное значение:

- **Spill Files** – объем spill-файлов в байтах, используемых в данный момент времени.

Для запроса из истории блок содержит следующие значения:

- **Spill Files (max)** – максимальный объем используемых spill-файлов в мебибайтах (MiB), зафиксированный за время выполнения запроса;
- **Spill Files (skew)** – паказатель в %, отражающий “перекос” в объеме созданных на сегментах spill-файлов для момента времени, в который наблюдалась величина Spill Files (max). Значение, отличное от нуля, говорит о том, что один из сегментов использовал больший объем spill-файлов, чем остальные. При расчете не учитывается мастер-сегмент (seg-1). Расчет значения производится следующим образом:

$$1 - \frac{avg(segment\ Spill\ Files)}{max(segment\ Spill\ Files)}$$

Потребление CPU

Блок содержит следующие значения:

- **CPU Avg** – средний объем потребляемого CPU для сегмента в %, зафиксированный в момент выполнения запроса (если запрос завершен, то на момент завершения запроса). При расчете не учитывается мастер-сегмент (seg-1). Потребление CPU в каждый конкретный момент рассчитывается на основе информации о процессорном времени из */proc/[pid]/stat* операционной системы, как дельта между текущим и предыдущим значением;
- **CPU Skew** – показатель в %, отражающий “перекос” в потреблении CPU на кластере. Значение, отличное от нуля, говорит о том, что один из сегментов использовал больший объем CPU, чем остальные. При расчете не учитывается мастер-сегмент (seg-1).

Потребление RAM

- **RAM Avg** – средний объем потребляемого RAM для сегмента в байтах, зафиксированный в момент выполнения запроса (если запрос завершен, то на момент завершения запроса). При расчете не учитывается мастер-сегмент (seg-1). Потребление RAM в каждый конкретный момент рассчитывается на основе метрики *rss (Resident Set Size)* из операционной системы;
- **RAM Skew** – показатель в %, отражающий "перекос" в потреблении CPU на кластере. Значение, отличное от нуля, говорит о том, что один из сегментов использовал больший объем RAM, чем остальные. При расчете не учитывается мастер-сегмент (seg-1).

Чтение с диска

- **Read Avg (per sec)** – средний объем чтения с диска для сегмента в байтах, зафиксированный в момент выполнения запроса (если запрос завершен, то суммарно на момент завершения запроса). При расчете не учитывается мастер-сегмент (seg-1). Чтение с диска в каждый конкретный момент рассчитывается на основе информации об IO Read из */proc/[pid]/stat* операционной системы, как дельта между текущим и предыдущим значением;
- **Read Skew** – показатель в %, отражающий "перекос" в чтении данных на кластере. Значение, отличное от нуля, говорит о том, что один из сегментов ичитывал больший объем с диска, чем остальные. При расчете не учитывается мастер-сегмент (seg-1).

Запись на диск

- **Write Avg (per sec)** – средний объем записи на диск для сегмента в байтах, зафиксированный в момент выполнения запроса (если запрос завершен, то суммарно на момент завершения запроса). При расчете не учитывается мастер-сегмент (seg-1). Запись на диск в каждый конкретный момент рассчитывается на основе информации об IO Write из */proc/[pid]/stat* операционной системы, как дельта между текущим и предыдущим значением;
- **Write Skew** – показатель в %, отражающий "перекос" в записи данных на кластере. Значение, отличное от нуля, говорит о том, что один из сегментов записывал на диск больший объем, чем остальные. При расчете не учитывается мастер-сегмент (seg-1).

Виртуальная память

- **Virtual Memory Avg** – средний объем потребляемой виртуальной памяти для сегмента в байтах, зафиксированный в момент выполнения запроса (если запрос завершен, то на момент завершения запроса). При расчете не учитывается мастер-сегмент (seg-1). Потребление виртуальной памяти в каждый конкретный момент рассчитывается на основе метрики *vsize (Virtual Memory Size)* из операционной системы;
- **Virtual Memory Skew** – показатель в %, отражающий "перекос" в потреблении виртуальной памяти на кластере. Значение, отличное от нуля, говорит о том, что один из сегментов использовал больший объем виртуальной памяти, чем остальные. При расчете не учитывается мастер-сегмент (seg-1).

Query Text

Текст запроса, отправленный на исполнение пользователем.

Plan&Progress

Блок содержит графическое представление дерева плана, отражающее динамику выполнения запроса. В каждом узле отображается тип (выполняемая операция) и прогресс выполнения. Значение прогресса, превышающее *100%*, свидетельствует о том, что планировщик сделал неправильное предположение о количестве кортежей, которое должно быть извлечено при выполнении данного узла.

При нажатии на узел открывается панель с дополнительной информацией (Рис.2.10.):

- *Cost* – коэффициент трудоемкости выполнения узла плана;

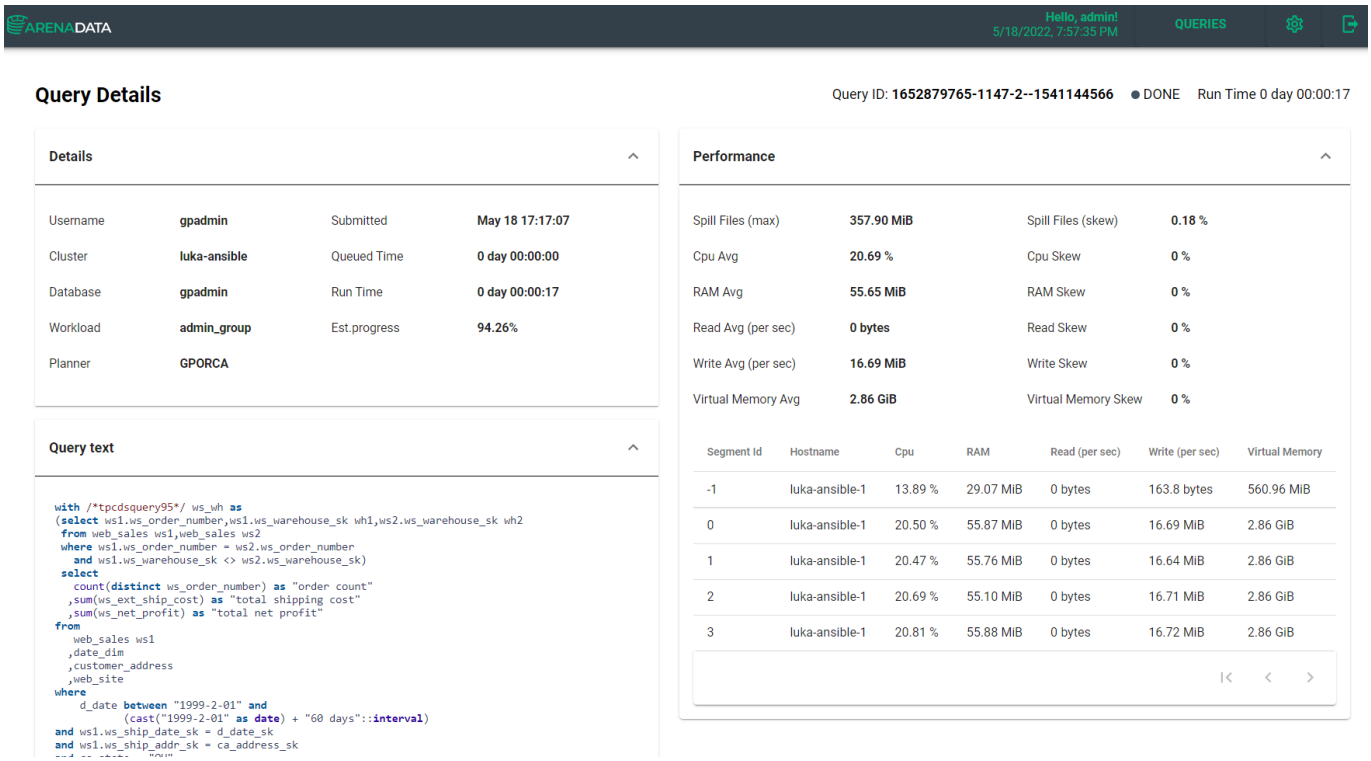


Рис.2.9.: Query Details - запрос из истории

- *Relation* – используемая таблица (опционально);
- *Row Skew* – перекос в данных. Разность между единицей и отношением среднего значения к максимальному количеству кортежей, полученных с сегментов;
- *Est. Tuples* – прогнозируемое количество кортежей;
- *Actual Tuples* – реальное количество извлеченных кортежей;
- *Operation Keys* – дополнительные условия, используемые при выполнении некоторых операций (например, *Hash Key*, *Join Filter*, *Merge Key*, *Filter*).

Цвет узла указывает на его состояние:

- Белый – проинициализирован, выполнение не начато;
- Темно-зеленый – выполняется;
- Светло-зеленый – выполнение закончено.

Textual plan

Текстовое представление плана запроса – результат выполнения команды *Explain*.

Кнопка *planchecker* в левом верхнем углу открывает текст запроса в сервисе https://planchecker.arenadata.io/, позволяющем получить детальную информацию о возможных проблемах производительности.

2.3.2 Отмена выполнения запроса

Исходя из прав доступа, определенных для пользователя, в правом верхнем углу страницы могут быть доступны кнопки, соответствующие двум способам отмены выполнения запроса (Рис.2.11.):



Рис.2.10.: Графическое представление дерева плана

- *Cancel* – отмена выполнения запроса путем вызова функции `pg_cancel_backend`, отменяющей запрос в обслуживающем процессе;
- *Terminate* – отмена выполнения запроса путем вызова функции `pg_terminate_backend`, завершающей обслуживающий процесс, в котором выполняется запрос.

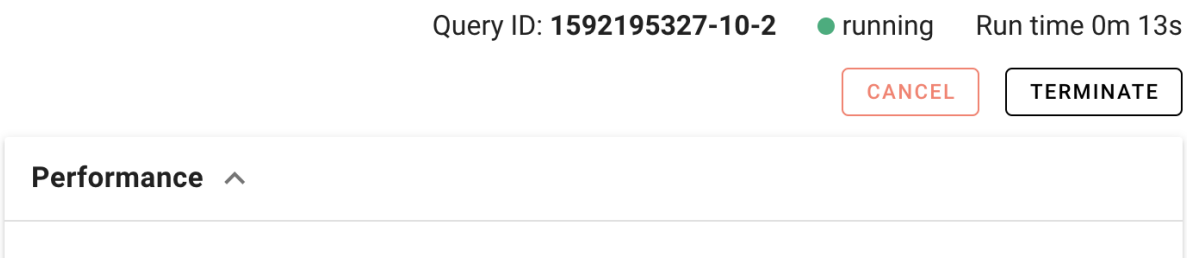


Рис.2.11.: Кнопки отмены выполнения запроса

Глава 3

Администрирование

- *Уровни доступа*
- *Аутентификация*
- *Управление пользователями*

3.1 Уровни доступа

В рамках **ADCC** предусмотрен механизм ролей, управляющий доступом пользователей к различной функциональности приложения. Каждая роль определяется присущими ей уровнями доступа:

- *manage users* – доступ к администрированию (создание, изменение, удаление пользователей, управление доступом для LDAP-групп);
- *view all queries* – возможность видеть все выполняемые в кластере ADB запросы;
- *kill all queries* – возможность отменять выполнение любого запроса в кластере ADB;
- *view self queries* – возможность видеть собственные запросы (запросы, выполненные ролью ADB с именем, идентичным логину текущего пользователя ADCC или названию LDAP-группы, в которую входит пользователь – см. *Аутентификация*);
- *kill self queries* – возможность отменять выполнение собственных запросов (запросов, выполненных ролью ADB с именем, идентичным логину текущего пользователя ADCC или названию LDAP-группы, в которую входит пользователь – см. *Аутентификация*).

Таким образом, авторизованному пользователю **ADCC** присваивается одна из следующих ролей:

уровень доступа \ роль	Owner	Administrator	Advanced User	Regular User
manage users	+			
view all queries	+	+	+	
kill all queries	+	+		
view self queries				+
kill self queries			+	+

3.2 Аутентификация

ADCC предоставляет два способа аутентификации пользователей:

- *Стандартная аутентификация*;

- *LDAP-аутентификация*.

3.2.1 Стандартная аутентификация

В данной схеме аутентификации логин пользователя ассоциируется с именем роли **ADB** на основании полного совпадения. Создание, модификация и удаление стандартных пользователей доступно пользователям, которым присвоена роль *Owner*.

3.2.2 LDAP-аутентификация

В приложении **ADCC** доступна аутентификация средствами LDAP. В данном методе проверка корректности логина и пароля пользователя происходит на заданном сервере LDAP.

В **ADCC** реализована поддержка следующих LDAP-реализаций:

- MSAD (Microsoft Active Directory);
- FreeIPA.

Параметры для подключения к серверу LDAP становятся доступны в конфигурации сервиса **ADCC** кластера **ADB** в **ADCM** после включения метода аутентификации LDAP (через переключатель слева от надписи “LDAP”):

- Type (по умолчанмю “MSAD”);
- URI;
- Users baseDN;
- Groups baseDN;
- Login;
- Password;
- Size limit (по умолчанию равно 1000);
- Lowercase login (по умолчанию чекбокс снят).

Пример конфигурации для Microsoft Active Directory:

```
* Type: MSAD
* URI: ldap://ad.example.com:389
* Users baseDN: ou=Peoples,dc=ad,dc=adcc-user
* Groups baseDN: ou=Groups,dc=ad,dc=adcc-group
* Login: cn=admin,dc=ad,dc=adcc-user
* Password: yourpassword
* Size limit (по умолчанию равно 1000);
```

Для уточнения параметров конкретной реализации LDAP в вашем случае обратитесь к администратору LDAP-каталога.

Important:

- Для логина в ADCC web-интерфейс с помощью LDAP-аутентификации от MSAD используйте upn-формат DN:

  `sAMAccountName@domain`

- Для входа в ADCC web-интерфейс с помощью LDAP-аутентификации от FreeIPA достаточно DN в обычном формате:

`sAMAccountName`

3.3 Управление пользователями

Для доступа к панели управления пользователями необходимо выбрать раздел *ADMIN* в меню в правом верхнем углу главной панели приложения. При этом открывается экранная форма *Admin* (Рис.3.1.).

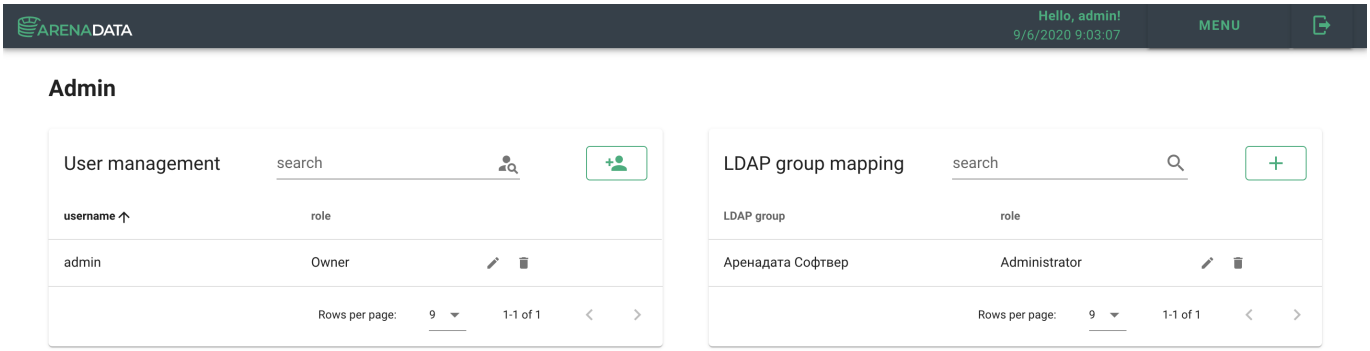


Рис.3.1.: *Окно управления пользователями*

3.3.1 Стандартные пользователи

Раздел *User Management* предоставляет следующие возможности управления стандартными пользователями **ADCC**:

- Для **создания** нового пользователя (Рис.3.2.):
  1. Нажать кнопку справа над списком;
  2. В открывшемся окне заполнить все поля:
  - Логин;
  - Роль;
  - Пароль и подтверждение пароля.
- Для **модификации** пользователя:
  1. Нажать иконку карандаша в соответствующей строке списка;
  2. В открывшемся окне изменить необходимые атрибуты пользователя:
  - Логин;
  - Роль;
  - Пароль (доступно после выбора чекбокса “Change password”).
- Для **удаления** пользователя нажать иконку мусорной корзины в соответствующей строке списка.

3.3.2 Управление доступом для LDAP-групп

В разделе *LDAP group mapping* доступно определение соответствий LDAP-групп заданного LDAP-сервера их ролям в рамках **ADCC**. Соответствие подразумевает, что каждому пользователю, входящему в данную

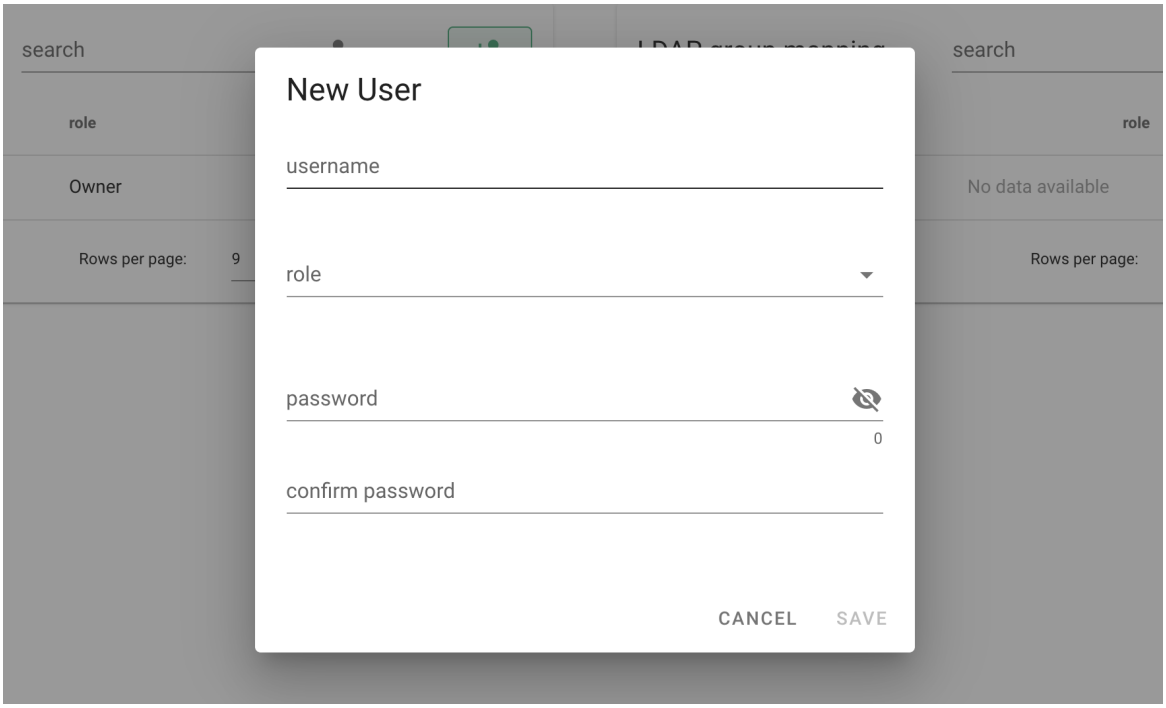


Рис.3.2.: *Создание нового пользователя*

группу, при авторизации в **ADCC** присваивается указанная роль. Каждой LDAP-группе может соответствовать только одна роль.

- Для **создания** нового соответствия (Рис.3.3.):
  1. Нажать кнопку *"+"* справа над списком;
  2. В выпадающем списке *group* открывшегося окна выбрать необходимую LDAP-группу;
  3. В выпадающем списке *role* выбрать необходимую роль;
  4. Нажать кнопку *save*.
- Для **изменения** роли, определенной для группы, нажать иконку карандаша в соответствующей строке списка;
- Для **удаления** соответствия нажать иконку мусорной корзины в соответствующей строке списка.

3.3.3 Настройка SSL в ADCC

Для включения SSL при работе с ADCC требуется в ADCM в разделе **ADCC Configuration** включить *HTTPS parameters enable* и настроить следующие поля:

Здесь нужно указать порт (8443 по умолчанию) и путь к SSL сертификату и его ключу.

При этом файлы *.crt* и *.key* для самоподписанного сертификата уже есть по умолчанию на машине где развернут ADCC в папке `/opt/adcc/ssl`, их можно указать например для целей тестирования:

- Full ssl certificate path = `/opt/adcc/ssl/adcc_default.crt`
- Full ssl key path = `/opt/adcc/ssl/adcc_default.key`

```
$ pwd
/opt/adcc/ssl
```

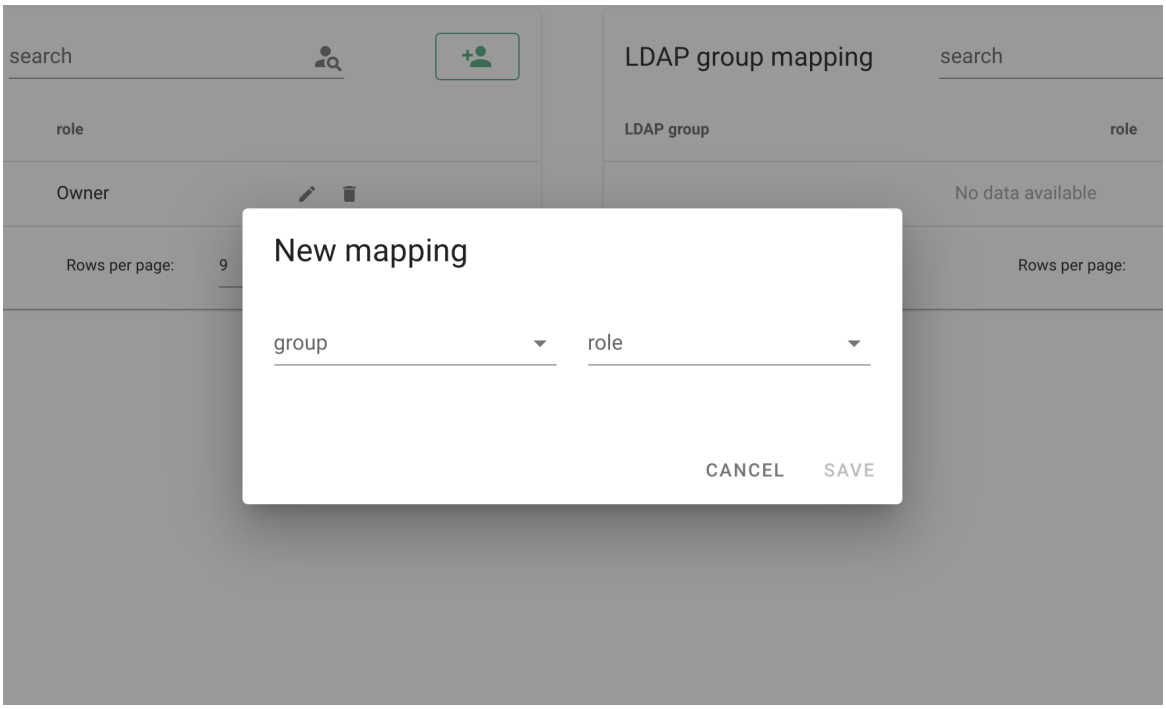


Рис.3.3.: *Создание нового соответствия*

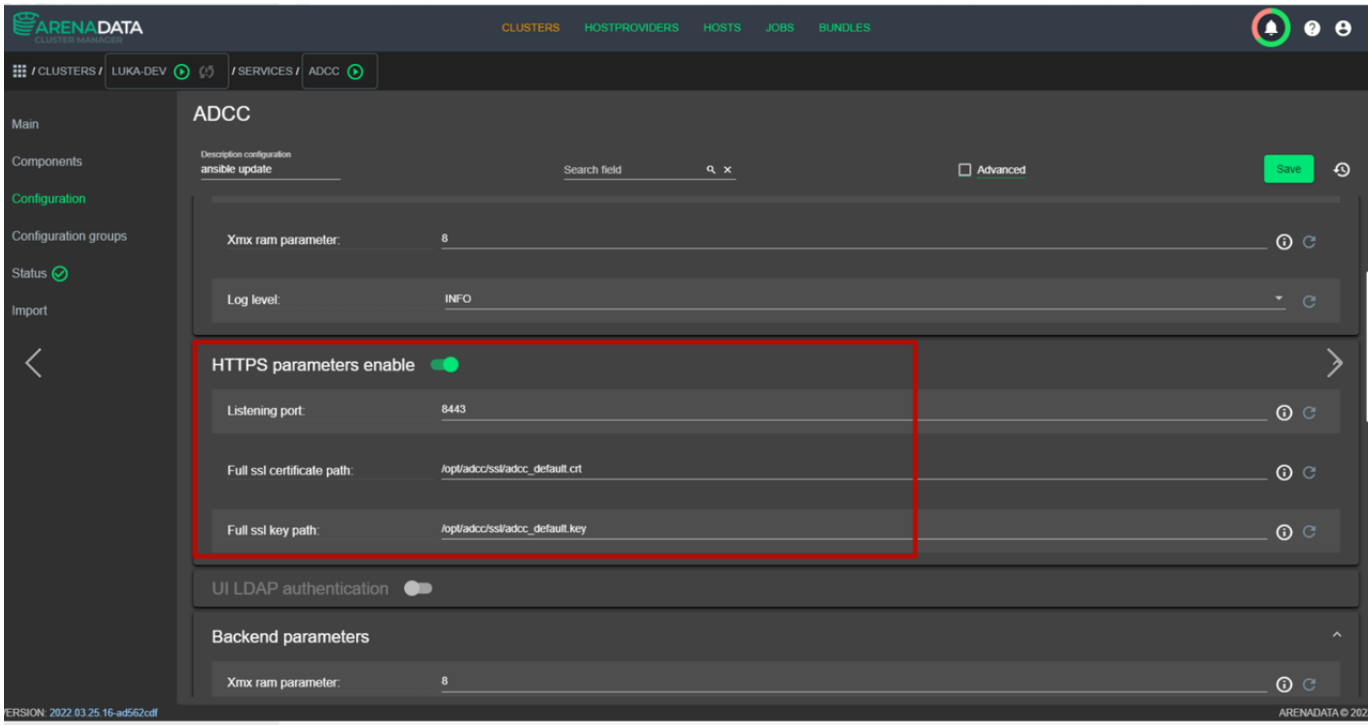


Рис.3.4.: *Окно HTTPS parameters enable в ADCM*

```
$ ls
adcc_default.crt  adcc_default.key  openssl.cnf
```

Чтоб создать реальный SSL сертификат и зарегистрировать его в центре сертификации используйте `openssl`.

Далее для применения настроек надо в ADCM сделать *Save* и *ADCC - > Reconfigure & Restart*

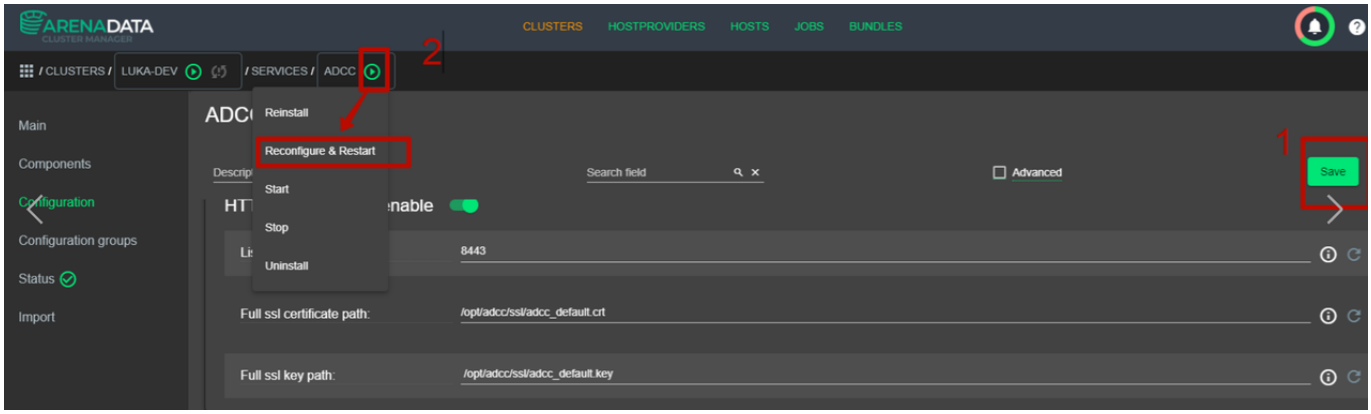


Рис.3.5.: *Save и Reconfigure & Restart в ADCM*

После этого в файле `/opt/adcc/nginx` появится запись (порт 8443):

```
server {
   listen 8443 ssl;
   server_name <your_server>;
   ssl_certificate      /etc/nginx/conf.d/adcc_ssl.crt;
   ssl_certificate_key  /etc/nginx/conf.d/adcc_ssl.key;
   ssl_prefer_server_ciphers   on;
```

После этого вход в ADCC без SSL на `http://<your_server>:81` автоматически будет перенаправлять на `https://<your_server>:8443` (Будет сообщение *Not secure* для https, так как используется самоподписанный сертификат)

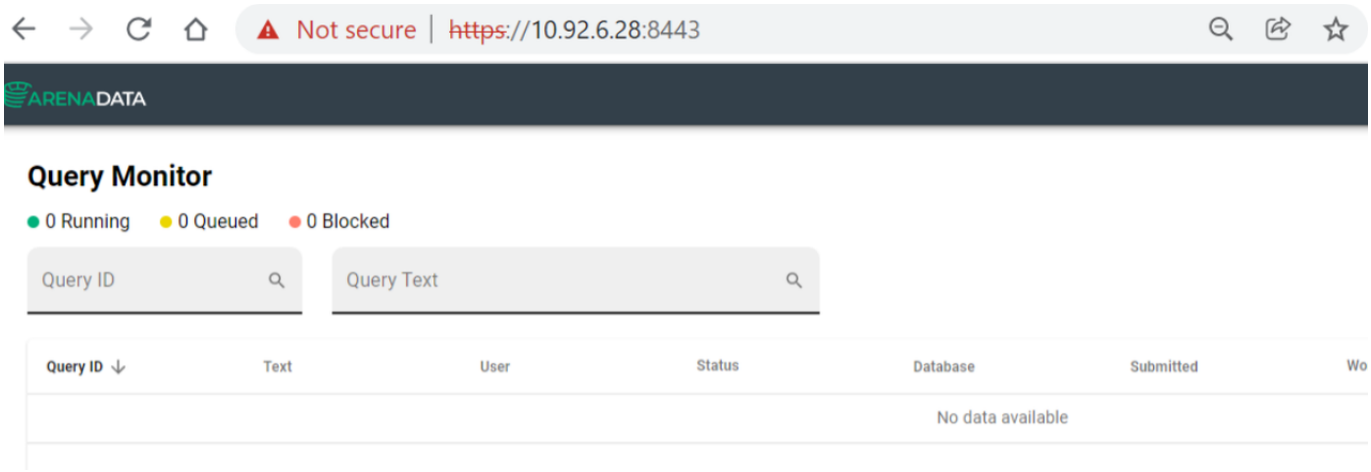


Рис.3.6.: *Редирект в ADCC на https*

3.4 Подключение внешнего ADCC и мониторинг нескольких кластеров

В ADCC, начиная с версии *6.18.0_arenadata28*, можно настроить мониторинг из одного ADCC Web Server несколько ADB кластеров. В примере ниже есть целевой кластер ADB1, где мы используем UI ADCC для мониторинга, и кластер-донор ADB2, который будет посылать свои метрики в целевой кластер.

Шаг 1. Настройка соединения в ADCM

Пропишите в ADCM для кластера ADB2, который хотите подключить, ip-адрес ADCC сервера кластера ADB1

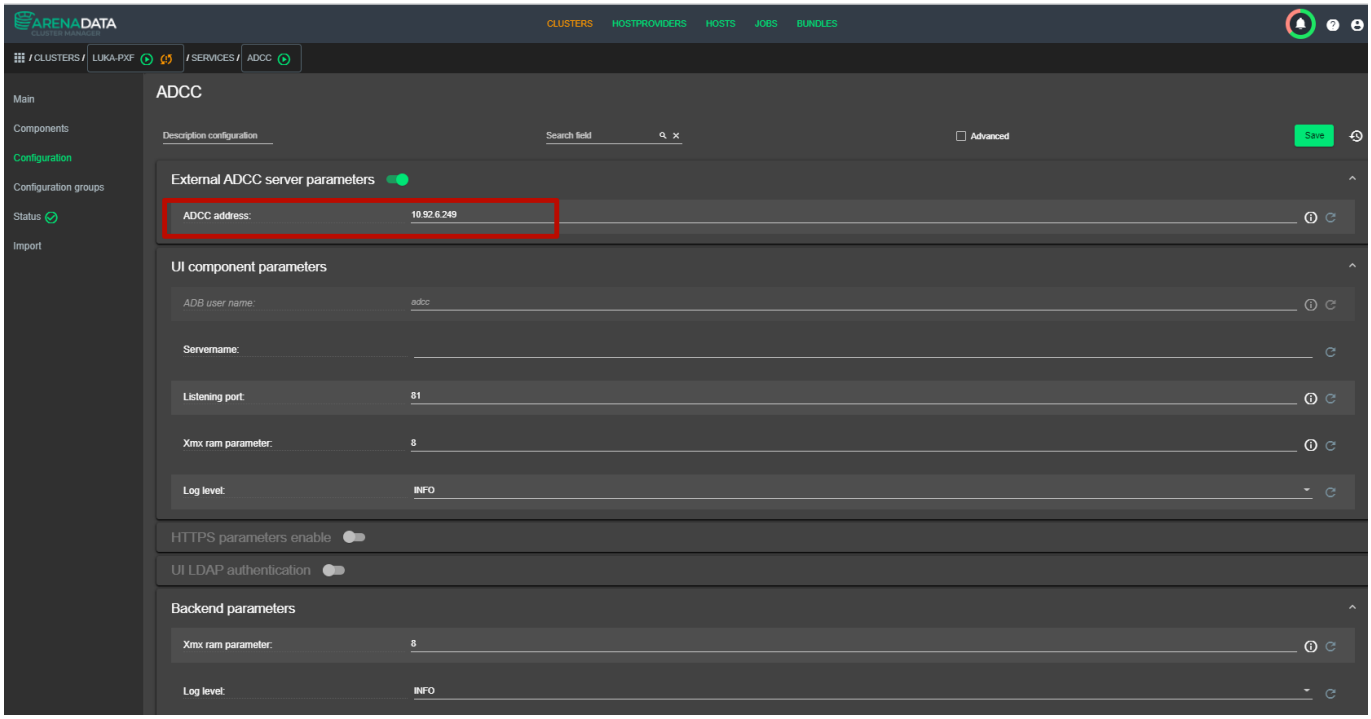


Рис.3.7.: *Настройка соединения в ADCM*

Запустите *ADCC Action Reconfigure & Restart* чтоб изменения вошли в силу.

Шаг 2. Настройка соединения в ADCC

Подключите в ADCC кластера ADB1, где хотите увидеть данные из кластера ADB2.

Important: Имя кластера при подключении в ADCC должно совпадать с его именем в ADCM. В противном случае соединение хоть и установится, но данные в ADCC приходить с этого кластера не будут. При этом кластер-донор и целевой кластер с ADCC не обязательно должны находиться в одном ADCM-контуре, соединение пройдет и между различными ADCM.

Формат JDBC (пример):

```
jdbc:postgresql://10.92.6.210:5432/postgres
```

Укажите пользователя ADB (по умолчанию это *gpadmin* без пароля)

Нажмите *Fetch databases* и отметьте базы данных которые хотите увидеть (если возникла ошибка то значит в `pg_hba.conf` кластера ADB2 не разрешен вход для нашего хоста).

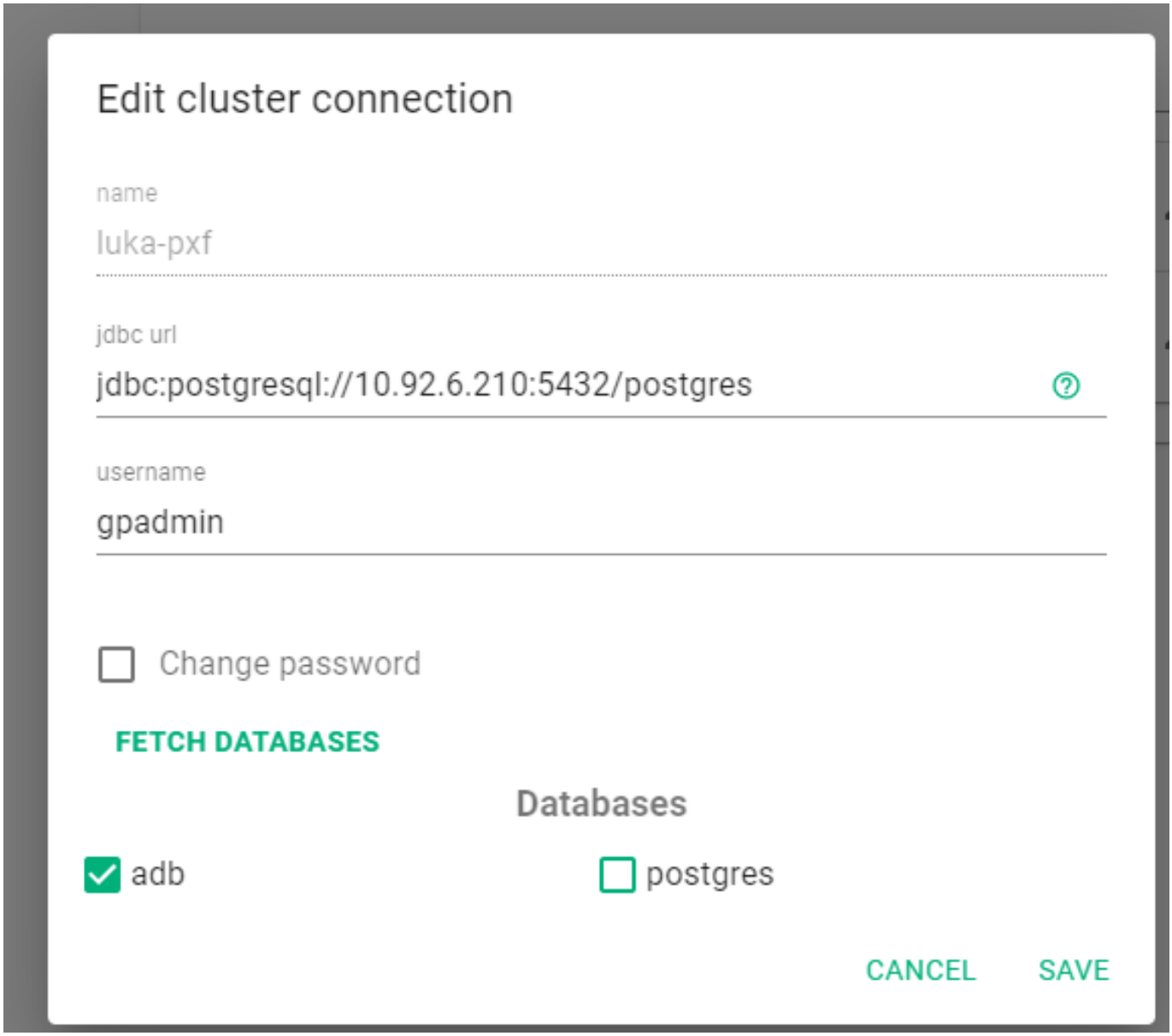


Рис.3.8.: *Редактирование соединения в ADCC*

В итоге имеем 2 кластера ADB в Configuration ADCC:

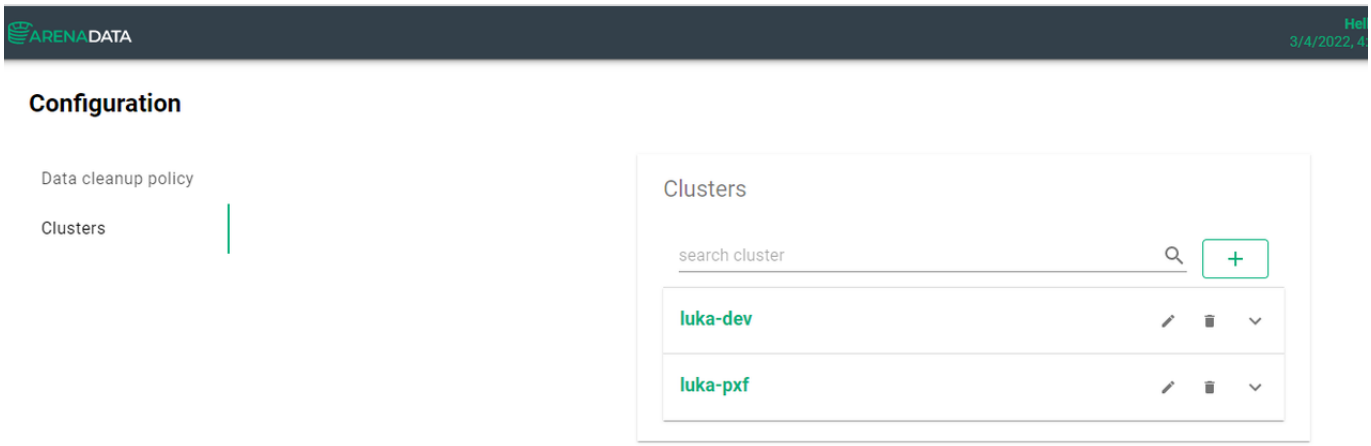


Рис.3.9.: *Окно настройки конфигурации в ADCC*

Шаг 3. Проверка подключения (опционально)

На кластере ADB2 выполняем команды :

```
$ psql adb
adb=# select version();
adb=# select pg_sleep(60);
```

В ADCC кластера ADB1 мониторим в *Query Monitor*, что они пришли на UI:

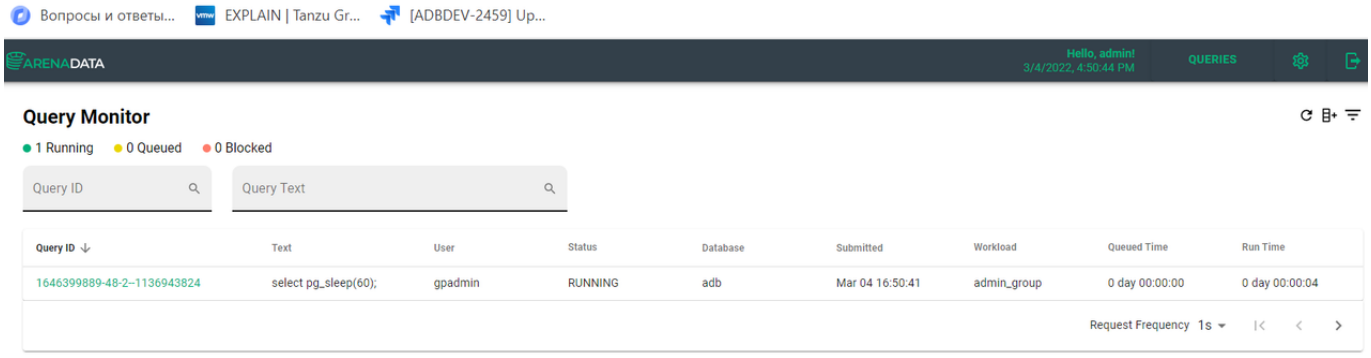


Рис.3.10.: *Окно Query Monitor в ADCC*

В *Query History* можно добавить колонку Cluster Name и видеть запросы к с обоих кластеров в одном UI:

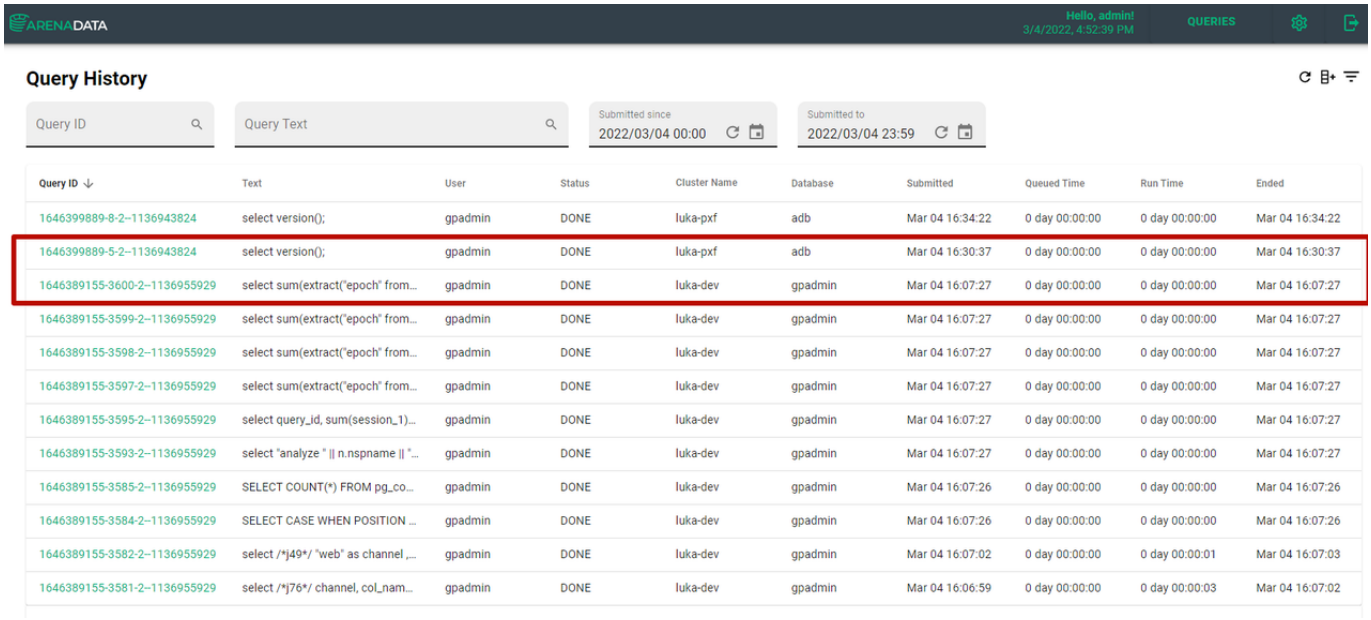


Рис.3.11.: *Окно Query History в ADCC*