Arenadata DB

Arenadata DB (ADB) — это распределенная аналитическая СУБД, построенная на базе MPP-системы с открытым исходным кодом Greenplage DB. Она предназначена для хранения и обработки больших объемов структурированных и полуструктурированных данных (до десятков петабайт).

Версия CURRENT Язык: RU

Содержание

1. Концепции
   - Архитектура ADB
   - Модель данных

2. Подготовка окружения
   - Требования к оборудованию
   - Требования к сети
   - Программные требования

3. Начало работы
   - Установка
     - Online-установка
       - Установка ADCM
       - Подготовка хостов
       - Установка кластера ADB
         - Создание кластера
         - Добавление сервисов
         - Добавление хостов в кластер
         - Добавление компонентов
         - Настройка сервисов
         - Настройка кластера
         - Установка кластера
     - Offline-установка
       - Установка ADCM
       - Подготовка хостов
       - Установка кластера Enterprise Tools
         - Создание кластера
         - Добавление сервисов
         - Добавление хостов в кластер
         - Добавление компонентов
         - Установка кластера
       - Установка кластера ADB
         - Создание кластера
         - Добавление сервисов
         - Добавление хостов в кластер
         - Добавление компонентов
         - Настройка сервисов
         - Настройка кластера
         - Импорт настроек ET
         - Установка кластера

4. Управление доступом
   - Аутентификация
     - LDAP
     - PAM с LDAP
     - Kerberos
       - MIT Kerberos
   - Учетные записи в ADB

5. Администрирование СУБД
   - Управление кластером через ADCM
     - Кластерные действия
     - Сервисные действия
       - ADB
       - ADB Control
       - ADBM
       - PXF
       - Коннекторы
       - Chrony
   - Настройка СУБД с помощью GUC
   - Резервное копирование и восстановление
     - Использование плагина ADB DDBoost
   - Оптимизация данных
     - Инструментарий arenadata_toolkit
     - Управление spill-файлами
     - Удаление "брошенных" файлов

6. Работа с данными
   - Обзор объектов базы данных
     - Базы данных
     - Схемы
     - Таблицы
       - Обзор таблиц
       - Типы таблиц
       - Сжатие данных
       - Распределение данных
       - Партиционирование
     - Табличные пространства

7. Интеграция с внешними системами
   - Arenadata DB Control
     - Обзор
     - Настройка внешних БД
     - Подключение к ADB Control
     - Управление доступом
       - Аутентификация
         - Basic
         - LDAP
       - Авторизация
       - Сессии пользователей
       - Шифрование SSL
     - Конфигурирование
       - Политики запуска задач
       - Управление кластерами
       - Политики безопасности
       - Общие настройки
     - Dashboard
     - Мониторинг
       - Команды
       - Транзакции
       - Сессии
       - Работа с тегами запросов
     - Аудит
     - Анализ работы системы
       - Просмотр jobs
       - Просмотр информации по компонентам
   - Arenadata DB Backup Manager
     - Обзор
     - Настройка внешней БД
     - Подключение к ADBM
     - Управление конфигурациями
     - Работа с бэкапами
       - Запуск действий
       - Просмотр бэкапов
       - Просмотр восстановлений БД
     - Disaster recovery
     - Просмотр действий
     - Использование ADBM CLI
   - Коннекторы
     - ADB ClickHouse Connector
       - Обзор
       - Установка
       - Настройка
       - Примеры использования
     - ADB и Kafka
       - ADB to Kafka Connector
         - Установка
         - Настройка
         - Примеры использования
       - Kafka to ADB Connector
         - Установка
         - Настройка
         - Примеры использования
     - ADB to ADB Connector
       - Обзор
       - Установка
       - Настройка
       - Примеры использования

8. Справочные материалы
   - Конфигурационные параметры
   - Поддерживаемые расширения
   - Список команд SQL, не поддерживаемых в ADB

9. Релизы
   - ADB
     - ADB 7
       - Релизы
       - Известные проблемы
     - ADB 6
       - Релизы
       - Известные проблемы
     - ADB 5
   - ADB Control
     - Релизы
     - Известные проблемы
   - ADBM
     - Релизы
     - Известные проблемы

Архитектура ADB

Автор: Антон Власенко

Содержание

Назад к содержанию

- Координатор
- Резервный координатор
- Сегменты
  - Основные сегменты
  - Зеркальные сегменты
  - Групповое зеркалирование
  - Распределенное зеркалирование
- Интерконнект
- Использование ADB для ETL
- Интеграция с другими системами
- Дополнительные инструменты по управлению кластером

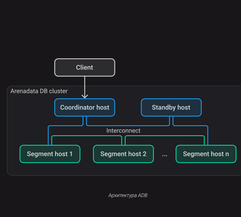
Arenadata DB (ADB) — это база данных, реализующая подход массово-параллельной архитектуры (Massively Parallel Processing, MPP), основанная на Greenplum DB.

В основе архитектуры MPP — кластер, состоящий из узлов, работающих одновременно. В ADB это реализовано как кластер отдельных экземпляров базы данных PostgreSQL.

ADB использует подход shared-nothing, при котором каждый сегмент содержит свою часть данных. Несколько сегментов могут работать на одном хосте, но обрабатывают свои данные независимо. Система масштабируется горизонтально — путем добавления новых физических хостов в кластер.

ADB состоит из трех ключевых компонентов:

- Координатор
- Резервный координатор
- Сегменты



Архитектура ADB

Координатор

Координатор (coordinator) (переженик название - мастер) — это точка входа в базу данных. Пользователи подключаются к координатору для отправки запросов.

В число функций координатора входит:

- Аутентификация клиентских подключений. Координатор управляет доступом к кластеру, используя стандартный файл *pg_hba.conf*. Сегменты принимают клиентские SQL-подключения только от координатора.
- Парсинг запросов. Координатор проверяет SQL-запрос на корректность синтаксиса. ADB использует синтаксис PostgreSQL с некоторыми изменениями, которые сделаны для поддержки распределенной архитектуры.
- Оптимизация. Оптимизатор определяет наиболее эффективный способ выполнения SQL-запроса, оценивая потенциальные планы выполнения запроса (query plan) и выбирая план с наименьшей расчетной стоимостью.
- Распределение нагрузки. Координатор распределяет рабочую нагрузку между обработчиками запроса на всех сегментах.
- Получение результатов. Координатор собирает результаты, сформированные сегментами (отфильтрованные строки, агрегированные значения или объединенные данные) и возвращает итоговый результат клиенту.
- Административные функции. Помимо собственно обработки запросов, координатор осуществляет управление и обслуживание баз данных. Например, он обновляет системные каталоги после получения DDL-команд (таких как `CREATE TABLE`).

Координатор не хранит пользовательские данные; они размещаются только на сегментах. Координатор хранит глобальный системный каталог (в виде таблиц, `pg_database`, `pg_attribute` и т.п.), который содержит метаданные о самой базе данных, их структуре и свойствах пользовательских данных. Когда пользователь создает таблицу, координатор сохраняет информацию о ее схеме, владельце и типе распределения, в фактическое содержимое таблицы хранится на сегментах.

Резервный координатор

Резервный координатор (standby coordinator) выполняет функции теплого резерва: он поддерживается в актуальном состоянии и готов к включению, но не обрабатывает запросы, пока не будет активирован. Хотя резервный координатор является опциональным, он крайне рекомендуется к использованию.

При отказе основного координатора резервный координатор не становится основным автоматически — действие по переключению должен выполнить администратор кластера. Вы можете использовать для этого утилиту `activate` (`gactivate`), доступную в веб-интерфейсе Arenadata Cluster Manager (ADCM).

Целостность и консистентность данных обеспечивается механизмом **write-ahead logging (WAL)**, унаследованным от PostgreSQL. WAL гарантирует, что все изменения данных сначала записываются в журнал WAL и сохраняются в постоянное хранилище до фактической модификации данных на диске. Это обеспечивает восстановление базы данных даже после сбоя путем повторного применения записей WAL.

Сегменты

Сегмент (segment) — это экземпляр PostgreSQL, хранящий и обрабатывающий свою часть данных. ADB распределяет пользовательские данные между сегментами на основе ключа распределения (distribution key), выбранного при создании таблицы. Равномерное распределение данных и рабочей нагрузки по сегментам — ключевое условие для достижения максимальной производительности ADB.

> **РЕКОМЕНДАЦИЯ**
>
> Рекомендуется, чтобы все хосты сегментов имели идентичные аппаратные ресурсы и конфигурацию. Это связано с тем, что координатор агрегирует результаты только после завершения работы всеми сегментами. Следовательно, производительность ADB будет определяться скоростью самого медленного хоста сегментов в кластере. Если один сегмент работает значительно медленнее остальных, это задержит выполнение всего запроса.

Основные сегменты

Основные сегменты (primary segment) — это активные сегменты, которые хранят пользовательские данные, получают планы запросов и реализуют их. Они выполняют локальные SQL-операции только со своей частью данных и отправляют результаты координатору.

Зеркальные сегменты

Зеркальные сегменты (mirror segment) обеспечивают отказоустойчивость для основных сегментов. При недоступности основного сегмента система автоматически переключается на его зеркальный сегмент. Зеркальный сегмент начинает функционировать как основной сегмент, а недоступный сегмент становится зеркальным сегментом. Администратор должен восстановить недоступный сегмент и вернуть сегменты в их исходные (**preferred**) роли.

Зеркальный сегмент всегда должен размещаться на отдельном от соответствующего основного сегмента хосте. Существует две конфигурации зеркалирования:

- Групповое зеркалирование
- Распределенное зеркалирование

Групповое зеркалирование

При **групповом зеркалировании (group mirroring)** все зеркальные сегменты основных сегментов одного хоста размещаются вместе на другом хосте. На рисунке, приведенном ниже, сегменты `Primary 0` , `Primary 1` располагаются на хосте `Segment host 1` , а их зеркальные сегменты `Mirror 0` , `Mirror 1` располагаются на хосте `Segment host 2`.

Основные свойства группового зеркалирования:

- Это наиболее дешевый вариант конфигурации, так как он может быть осуществлен с помощью всего двух хостов.
- В случае отказа такая конфигурация приводит к неравномерной нагрузке на хосты. Если один хост становится недоступным, его резервный хост получает двойную нагрузку, так как помимо собственных основных сегментов (`Primary 2` , `Primary 3` на рисунке ниже) его зеркальные сегменты (`Mirror 0` , `Mirror 1`) тоже начинают работать в режиме основных сегментов. Производительность хоста снижается на 50% и более.

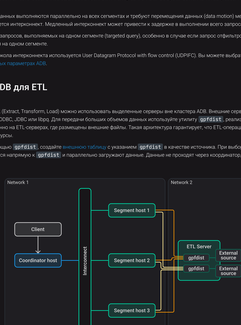
Групповое зеркалирование

Распределенное зеркалирование

При **распределенном зеркалировании (spread mirroring)** зеркальные сегменты для основных сегментов каждого хоста размещаются на разных резервных хостах — по количеству сегментов на хост. На рисунке ниже сегменты `Primary 0` , `Primary 1` находятся на хосте `Segment host 1` , а их зеркальные сегменты распределены по разным хостам: `Mirror 0` — `Segment host 2` , `Mirror 1` — `Segment host 3`.

Основные свойства распределенного зеркалирования:

- Повышенная отказоустойчивость. Сбой хоста приводит к активации только одного зеркального сегмента на каждом другом хосте, что снижает влияние на производительность. В примере на рисунке ниже если `Segment host 1` становится недоступным, на хосте `Segment host 2` активируется сегмент `Mirror 0` , а на хосте `Segment host 3` активируется сегмент `Mirror 1`. Нагрузка на каждый хост возрастает на 50%, при этом эта дополнительная нагрузка равномерно распределяется между хостами.
- Более высокая стоимость. Такая конфигурация требует, чтобы количество хостов было как минимум на один больше, чем количество сегментов на одном хосте.



Распределенное зеркалирование

Интерконнект

Интерконнект — это высокоскоростная сеть, обеспечивающая обмен данными между сегментами во время выполнения запросов. Рекомендуется использовать сеть 10 Гбит/с или быстрее.

Большинство запросов в базе данных выполняется параллельно на всех сегментах и требует перемещения данных (data motion) между сегментами. Для такого перемещения данных используется интерконнект. Медленный интерконнект может привести к задержке в выполнении всего запроса.

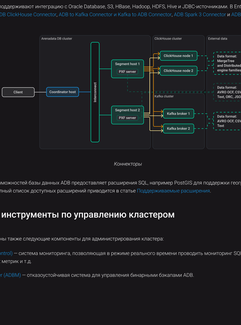
Интерконнект не требуется для запросов, выполняемых на одном сегменте (targeted query), особенно в случае если запрос отфильтрован по ключу распределения и все необходимые данные находятся на одном сегменте.

По умолчанию в качестве протокола интерконнекта используется User Datagram Protocol with flow control (UDPFC). Вы можете выбрать тип интерконнекта (`udpfc`, `tcp` или `proxy`) в конфигурационных параметрах ADB.

Использование ADB для ETL

Для выполнения процессов ETL (Extract, Transform, Load) можно использовать выделенные серверы вне кластера ADB. Внешние серверы могут подключаться к кластеру через стандартные протоколы ODBC, JDBC или libpq. Для передачи больших объемов данных используется утилита `gpfdist`, реализующая File Distribution Protocol. Ее можно запускать непосредственно на ETL-серверах, где размещены внешние файлы. Такая архитектура гарантирует, что ETL-операции и собственно SQL-запросы в кластере не конкурируют за ресурсы.

Чтобы загрузить данные с помощью `gpfdist`, создайте внешнюю таблицу с указанием `gpfdist` в качестве источника. При выборке данных с помощью такой таблицы сегменты подключаются напрямую к `gpfdist` и параллельно загружают данные. Данные не проходят через координатор, он только управляет процессом.



Загрузка данных из ETL

Интеграция с другими системами

Для интеграции с разнообразными источниками и форматами данных ADB использует Platform Extension Framework (PXF) и специализированные коннекторы.

PXF — это фреймворк доступа к данным, позволяющий выполнять запросы напрямую к внешним системам. Он позволяет выполнять федеративные запросы (federated query), при которых можно получать и объединять данные из разных источников в одном запросе. PXF также поддерживает технологию pushdown, позволяющую передать фильтрацию данных для условий `WHERE` внешним источникам. Такой подход снижает нагрузку, которая в итоге передается в ADB. PXF обеспечивает работу с такими форматами данных, как Avro, OCF, ORC, Text, ORC и JSON.

На основе PXF разрабатываются коннекторы для подключения и передачи данных между кластером и отдельными внешними системами (такими, как файловые хранилища, базы данных и потоковые платформы).

Встроенные в PXF коннекторы поддерживают интеграцию с Oracle Database, S3, HBase, Hadoop, HDFS, Hive и JDBC-источниками. В Enterprise версии ADB доступны дополнительные коннекторы: ADB Clickhouse Connector, ADB to Kafka Connector, Greenplum to ADB Connector, ADB Spark 3 Connector и ADB to ADB Connector.

Коннекторы

Кроме того, для дополнения возможностей базы данных ADB предоставляет расширения SQL, например PostGIS для поддержки географических объектов или plpgsql для поддержки языка PL/pgSQL. Полный список доступных расширений приводится в статье Поддерживаемые расширения.

Дополнительные инструменты по управлению кластером

В состав ADB Enterprise доступны также следующие компоненты для администрирования кластера:

- Arenadata DB Control (ADB Control) — система мониторинга, позволяющая в режиме реального времени проводить мониторинг SQL-операций и транзакций, сессий, ресурсных групп, системных метрик и т.д.
- Arenadata DB Backup Manager (ADBM) — отказоустойчивая система для управления бинарными бэкапами ADB.

Модель данных

Дарья Барышева

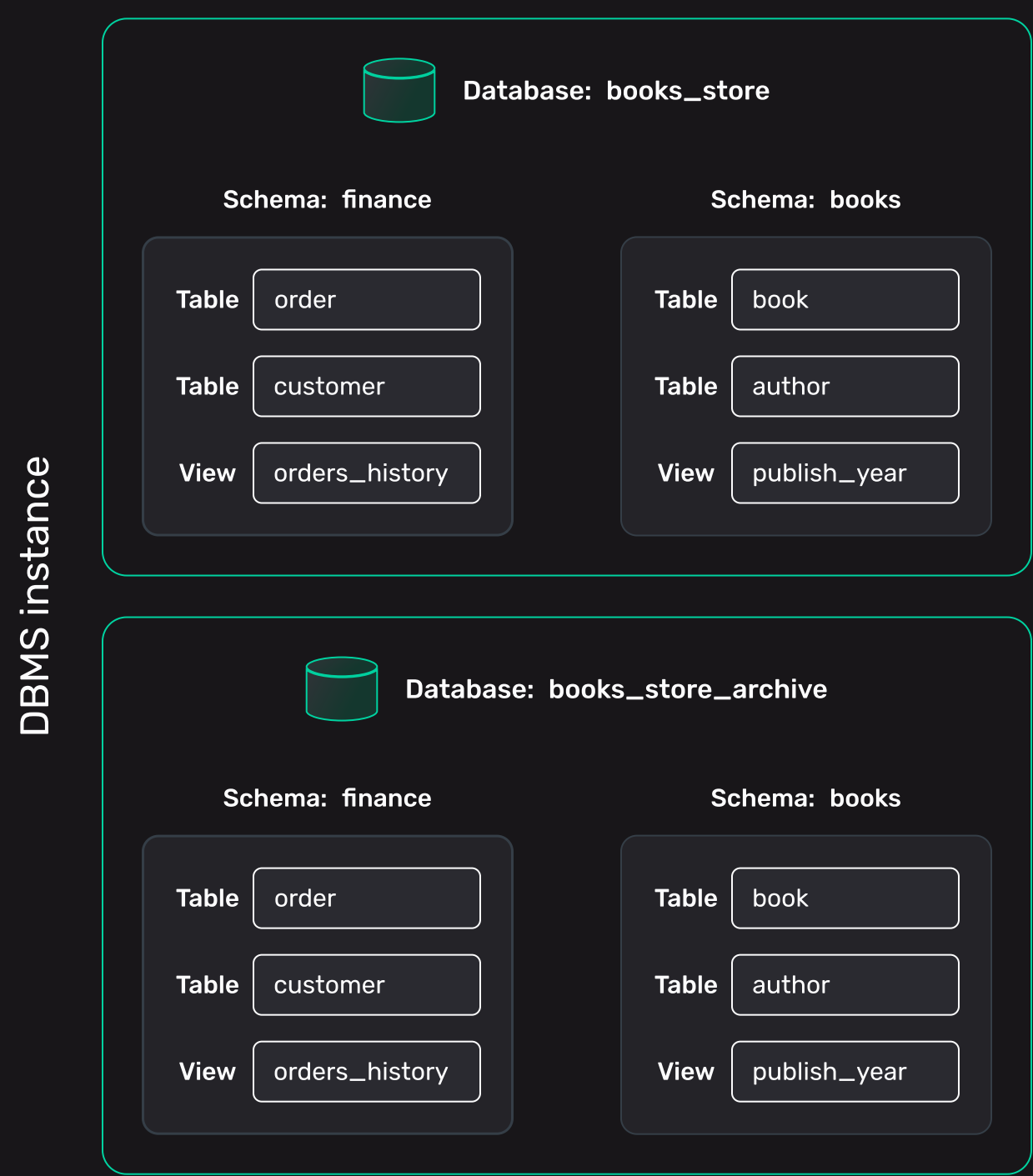
Назад к содержанию

ADB построена на базе MPP-системы с открытым исходным кодом Greengage DB, в основе которой, в свою очередь, лежит объектно-реляционная СУБД PostgreSQL. Поэтому большинство концепций, лежащих в основе модели данных PostgreSQL, применимо также и к ADB.

Основные компоненты модели данных ADB приведены ниже:

- База данных (БД, database). Это набор данных, физически хранимых вместе. Каждый ADB-кластер может содержать одну или более баз данных. В этом заключается отличие от некоторых других СУБД (например, Oracle), где инстансу СУБД соответствует одна база данных. Тем не менее, клиентские приложения могут обращаться не более чем к одной базе данных ADB в рамках одного соединения: кросс-запросы между различными базами данных невозможны по причине физического разделения БД. Возможность переноса данных между БД также отсутствует.
- Схема (schema). Это логическое объединение сущностей внутри одной базы данных. В отличие от БД, схемы разделены не так жестко: пользователи могут обращаться к нескольким схемам одновременно (при наличии соответствующих прав).
- **Объект (object)**. Это любая иная сущность, определенная внутри схемы БД. Например, таблица, функция, представление или материализованное представление и так далее. Для доступа к объектам можно использовать ANSI-стандарт: `database.schema.object`. Но форма `schema.object` предпочтительна (поскольку обращаться можно только к той БД, что указана в настройках текущего соединения). Обратите внимание, что одно и то же имя объекта может быть использовано в различных схемах.

Рисунок, приведенный ниже, иллюстрирует иерархию компонентов модели данных ADB на простом примере. Здесь представлены две базы данных `books_store` и `books_store_archive` с двумя схемами `finance` и `books` в каждой из них. Схемы, в свою очередь, содержат несколько таблиц и представлений. В этом примере допускается одновременное обращение к двум схемам любой БД (указанной в настройках текущего соединения). Однако одновременные запросы к обеим базам данных запрещены.



Пример модели данных

Требования к оборудованию

Дарья Барышева

Содержание

Назад к содержанию

ADB

ADB Control

ADBM

Arenadata Enterprise Tools

ВАЖНО

Системные требования, приведенные ниже, являются **минимальными**. Целевой сайзинг необходимо рассчитывать исходя из требований вашей организации.

ADB

Хост координатора

- Физический сервер.
- **CPU:** от 8 ядер.
- **RAM:** от 16 ГБ.
- **Диск:** отдельный физический неформатированный массив RAID 10 (от 5 ГБ).

Сегмент-хосты

- Физический сервер.
- **CPU:** от 16 ядер.
- **RAM:** от 32 ГБ.
- **Диск:** один или несколько физических неформатированных массивов RAID 10 (от 10 ГБ каждый).

ADB Control

ПРИМЕЧАНИЕ

Более подробную информацию об архитектуре ADB Control можно получить в статье Обзор Arenadata DB Control.

Хост с ADB Control

Ниже приведены минимальные требования к хосту, на котором устанавливается ADB Control.

Сервис	CPU в ядрах	RAM	Диск	Примечание
Backend Server	6	16 ГБ	—	Требуется дополнительный мониторинг внутренних очередей метрик
Scheduler	2	4 ГБ	—	Потребление ресурсов может зависеть от настроек системных задач — job (таких как тайм-ауты, размер батчей, число одновременно выполняющихся задач и так далее)
Metrics DB (ClickHouse)	4	16 ГБ	100 ГБ	Рекомендуется иметь объем RAM, равный сжатым данным в ClickHouse
Query DB (PostgreSQL)	4	8 ГБ	200 ГБ	Необходим тщательный мониторинг использования дискового пространства во избежание его исчерпания между запусками процедуры очистки данных (cleanup) в ADB Control
Другие компоненты: Web UI и UI Backend Server, Service Registry	4	20 ГБ	10 ГБ	—
Всего	20	64 ГБ	310 ГБ	—

Хосты ADB

Ниже приведены минимальные требования к хостам ADB, на каждом из которых устанавливается агент ADB Control:

- **CPU:** от 4 ядер.
- **RAM:** от 4 ГБ.

ADBM

ПРИМЕЧАНИЕ

Более подробную информацию об архитектуре ADBM можно получить в статье Обзор Arenadata DB Backup Manager.

Хост с ADBM

Arenadata DB Backup Manager может быть установлен на том же хосте, что и ADB Control, либо на отдельном хосте. Минимальные требования к хосту, на котором устанавливается ADBM (включая сервисы etcd, Backup Manager, Service Registry), приведены ниже:

- **CPU:** от 4 ядер.
- **RAM:** от 12 ГБ.
- **Диск:** от 10 ГБ.

Хосты ADB

На каждом сегмент-хосте ADB агент ADBM и инстанс pgbackrest используют порядка 30% CPU и 2 ГБ RAM (в случае однопоточного процесса бэкапирования).

Для работы агентов ADBM на сегмент-хостах требуется наличие сети со скоростью не ниже 1 Гбит/с. Однако рекомендуемое значение скорости составляет 10—25 Гбит/с (конкретная величина определяется требованиями к бэкапам).

Arenadata Enterprise Tools

Если вместе с вашим продуктом вы используете ET, примите во внимание следующие рекомендации:

- Виртуальный сервер.
- CPU: от 4 ядер.
- RAM: от 8 ГБ.
- Диск: от 50 ГБ.

Требования к сети

Дарья Барышнева

Содержание

Назад к содержанию

Обзор
Порты
- Порты ADB
- Порты ADB Control
- Порты ADBM
- Порты Arenadata Enterprise Tools

Обзор

Сетевые требования для установки Arenadata DB приведены ниже:

- Необходимо наличие одной или нескольких выделенных и обособленных одноранговых интернэкспертсетей под цели внутренней коммуникации кластера.
- Все хосты кластера должны быть подключены к интернэкспертсетям.
- Все хосты кластера должны иметь IP-адреса и возможность обмениваться данными друг с другом на заданных скоростях.
- Должна быть обеспечена доступность `ping` любого из хостов кластера (ICMP).
- В интернэкспертсетях должен быть установлен `MTU = 9000` (jumbo frame), чтобы пакеты, формируемые ADB (`gp_max_packet_size` + overhead) помещались в эти фреймы целиком.
- Необходимо обеспечить скорость Ethernet от 1 Гбит/с (стандартом является 10 Гбит/с).
- Рекомендуется отключение firewall и другого ПО, блокирующего или замедляющего сетевой трафик (`firewalld`, `iptables`, `ufw` и так далее). Если firewall не отключен, необходимо обеспечить возможность взаимодействия между хостами ADB-кластера. Дополнительную информацию можно получить в документации Greenplum.
- Снаружи кластер должен быть доступен на следующих портах:
  - `22` – SSH;
  - `5432` – PostgreSQL-интерфейс для входящих соединений.
- Для доступа к серверам точного времени (Network Time Protocol, NTP) хосту координатора должен быть доступен порт `123/udp`.
- Для отправки статусов компонентов кластера в Arenadata Cluster Manager (ADCM) к нему должно быть разрешено подключение хостов ADB (по умолчанию через порт `8000/tcp`). Информацию об установке ADCM можно получить в статье Установка ADCM.
- В случае установки ADB без доступа в Интернет (offline) требуется доступ к развернутому на выделенном сервере кластеру Arenadata Enterprise Tools (по умолчанию через порты `8172/tcp`, `443/tcp`). Установка Enterprise Tools описана в статье раздела Установка кластера Enterprise Tools.
- Все порты, перечисленные ниже, должны быть открыты на соответствующих хостах. Если используется firewall, необходимо добавить перечисленные внешние порты в список исключений.

Порты

Порты ADB

Coordinator/standby

Сервис	Порт	Протокол	Примечание
ADB	5432	TCP	Порт для клиентских SQL-соединений на хосте координатора ADB. Поддерживается клиенты, использующие PostgreSQL libpq API. Является настраиваемым
Standby coordinator replicator	1025-65535	TCP	Порты, используемые для репликации лога координатора на резервный координатор (standby)

Все хосты кластера

Сервис	Порт	Протокол	Примечание
Secure shell (SSH): `gpssh`, `gpssh-exkeys`, `gpdbg`	22	SSH	Многие утилиты Greenplum используют протоколы Secure Copy Protocol (SCP) и Secure Shell (SSH) для управления кластером и передачи файлов между хостами
Interconnect	10000-65535	UDP	Интерконнект перемещает кортежи базы данных между сегментами кластера в процессе выполнения запросов (точный перечень портов можно получить в системной таблице `gp_segment_configuration`)
Gpperfmon agents	8888	TCP	Порт для агентов `gpperfmon` (`gpmmon` и `gpsmon`), запущенных на хостах кластера. Настраивается путем установки конфигурационного параметра `gpperfmon_port` в файле postgresql.conf на координаторе и сегмент-хостах
PXF	5888	TCP	Сервис PXF Java запускается на порте `5888` на каждом сегмент-хосте
gpbackup	25 587	TCP	—

Хосты кластера/внешние хосты

Сервис	Порт	Протокол	Примечание
gpfdist	8080	HTTP	Обслуживающая файлы утилита `gpfdist`, может быть запущена на хостах ADB либо на внешнем хосте. Порт можно определить путем установки опций `-p` (начальный порт диапазона) и `-P` (конечный порт диапазона) при запуске сервера. Значение `8880` используется по умолчанию в качестве начального порта
gpload	8000-9000	HTTPS	Утилита `gpload` запускает один или более инстансов `gpfdist` на портах или диапазонах портов, указанных в конфигурационном файле. Значение `8000-9000` используется по умолчанию и может быть изменено

Хост ADQM/ClickHouse

Сервис	Порт	Протокол	Примечание
ADB ClickHouse Connector	8123	HTTP	Все ноды кластера ADB должны иметь доступ к HTTP-порту `8123` на всех нодах кластера ADQM/ClickHouse, через которые планируется производить загрузку данных

При использовании ADB to ADB Connector необходимо дополнительно убедиться, что TCP-порты, используемые для соединения, открыты на хостах удаленного (remote) кластера ADB: на координаторе, сегмент-хостах и резервном координаторе (standby). Список используемых портов можно получить с помощью следующей команды на координаторе remote-кластера ADB:

```
SELECT port FROM gp_segment_configuration;
```

Если используется firewall, добавьте порты, возвращаемые командой, в список исключений.

Порты ADB Control

ПРИМЕЧАНИЕ

Более подробную информацию об архитектуре ADB Control можно получить в статье Обзор Arenadata DB Control.

Хост с ADB Control

Сервис	Порт	Протокол	Примечание
Service Registry	6567	gRPC (HTTP 2.0)	gRPC endpoint для инстансов Service Registry
	8761	HTTP	—
UI Backend Server	6566	gRPC (HTTP 2.0)	—
	8890	HTTP	—
	81	HTTP	Вход в ADB Control UI
Backend Server	6565	gRPC (HTTP 2.0)	—
	8081	HTTP	—
Scheduler	6572	gRPC (HTTP 2.0)	—
Query DB (PostgreSQL)	5433	TCP	Внешний порт сопоставляется порту `5432`, внутри контейнера
Metrics DB (ClickHouse)	8123	HTTP	Порт HTTP API для HTTP-запросов. Используется JDBC, ODBC и web-интерфейсами
	9000	TCP	Порт для взаимодействия по протоколу Native (также именуемому протоколом ClickHouse TCP). Используется приложениями и процессами ClickHouse, такими как `clickhouse-server`, `clickhouse-client` и нативные утилиты ClickHouse, а также для распределенных запросов при межсерверном взаимодействии
Migration	8899	HTTP	—

Хосты ADB

Сервис	Порт	Протокол	Примечание
Agent	6581	TCP	Порт для utility-сервиса, который используется для управления параметрами GUC, изменяемыми из ADB Control
Agent	5701	TCP	Порт, являющийся частью адреса кластерной ноды Hazelcast (распределенного кеша). Порт должен быть открыт на каждом хосте кластера ADB (с агентом ADB Control), поскольку используется для взаимодействия кластерных нод Hazelcast друг с другом
Agent	От `36001` до `36001 + <число хостов ADB> - 1`	TCP	Диапазон разрешенных выходных портов для соединения с другими членами кластера Hazelcast. Порты должны быть открыты на каждом хосте кластера ADB с агентом ADB Control

Порты ADBM

ПРИМЕЧАНИЕ

Более подробную информацию об архитектуре ADBM можно получить в статье Обзор Arenadata DB Backup Manager.

Хост с ADBM

Сервис	Порт	Протокол	Примечание
Backup Manager	6570	TCP	—
	8870	TCP	—
	8988	TCP	Внешний порт сопоставляется порту `8843`, внутри контейнера
Service Registry	6567	TCP	—
	8761	TCP	—
PostgreSQL	5432	TCP	Сервис базы данных PostgreSQL доступен на localhost, портом по умолчанию является `5432`
etcd	2391 2392 2393	TCP	Внешние порты сопоставляются порту `2379`, внутри каждого инстанса etcd

Хосты ADB

Сервис	Порт	Протокол	Примечание
Agent	6571	TCP	Порт, через который работают gRPC-вызовы из ADBM к агентам

Порты Arenadata Enterprise Tools

Порт	Протокол	Примечание
443	TCP	Порт, используемый Docker Registry
81	TCP	Порт, используемый для доступа к репозиториям в случае offline-установки

Программные требования

Дарья Барышева

Содержание

Назад к содержанию

Обзор

Требования для Astra Linux и Ubuntu

Предварительная настройка ОС

Обзор

Программные требования для установки Arenadata DB приведены ниже:

1. ADB поддерживает следующие операционные системы:
   - Ubuntu 22.04. См. Требования для Astra Linux и Ubuntu ниже.
   - AltLinux 10.2 SP (для Enterprise-версии ADB).
   - РЕД ОС 7.3 Сертифицированная редакция (для Enterprise-версии ADB).
   - Astra Linux 1.7.6 "Орел" SE и Astra Linux 1.7.6 "Воронеж" (для Enterprise-версии ADB). См. Требования для Astra Linux и Ubuntu ниже.
2. Если ClickHouse или PostgreSQL, используются в качестве внешних систем хранения для ADB Control/ADBM (см. Metrics DB и Query DB в статье Обзор Arenadata DB Control), требования к их версиям следующие:
   - ClickHouse 22.8 — 25.9
   - PostgreSQL 12.20
3. ADB использует Docker в качестве платформы контейнеризации для части сервисов (например, ADB Control, ADBM) и для Enterprise Tools. ADB поддерживает версии Docker, поставляемые вендорами Linux для нужд своих операционных систем. Также есть возможность использования предустановленного Docker — путем установки флага ***Use existing docker*** во время конфигурирования кластера ADB.

ВАЖНО

Начиная с ADB 6.27.1.56 сервис ADB Control перезапускает настройки Docker в файле /etc/docker/daemon.json в целях повышения безопасности. Поэтому при необходимости внесения изменений в этот файл используйте опцию ***Use existing docker*** при настройке кластера ADB. Эта опция также позволяет запускать Docker в режиме rootless.

4. ADB использует Java 17 для некоторых сервисов (например, PXF, ADB Control, ADBM). Во всех операционных системах, кроме Astra Linux, дополнительная установка ПО не требуется. Java поставляется с основным репозиторием операционной системы. В Astra Linux перед установкой ADB необходимо установить Java на хостах кластера и прописать путь к ней в поле ***JAVA_HOME*** на этапе конфигурирования кластера ADB. Заполнение поля также требуется в других операционных системах в случае использования пути, отличного от дефолтного.

   Пути, используемые по умолчанию в поддерживаемых ОС, приведены ниже:

   Ubuntu

   /usr/lib/jvm/java-17-openjdk-amd64

   РЕД ОС

   /usr/lib/jvm/jre-17-openjdk

   AltLinux

   /usr/lib/jvm/jre-17

ВАЖНО

- Если вы планируете использовать собственную Java (например, от Axiom), для установки требуется Java версии 17.
- В Astra Linux заполнение поля ***JAVA_HOME*** обязательно начиная с версии ADB 6.27.1.56.

5. Минимальная версия ADCM — 2.6.0.
6. Если вы используете ADBM, то его минимальная версия для ADB 7 — 2.8.0.
7. Если вы используете ADB Control, то его минимальная версия для ADB 7 — 4.15.0.
8. ADB требует наличия локали ОС `en_US.UTF-8` на всех хостах кластера, при этом она необязательно должна быть выбрана по умолчанию. Эта локаль по умолчанию присутствует во всех поддерживаемых операционных системах, кроме Astra Linux. В Astra Linux локаль автоматически добавляется при установке ADB. Проверить доступные локали можно следующей командой:

```
$ localectl list-locales
```

9. Рекомендуется отключать антивирусные программы перед установкой и в течение процесса эксплуатации ADB. Если антивирус все же используется — пожалуйста, уточните настройки, необходимые для использования ADB, у поставщика вашего антивирусного ПО.
10. При выборе имен объектов баз данных ADB следует использовать только латинские буквы и цифры. Не следует использовать следующие зарезервированные символы: `@`, `&`, `.`, `,` (пробел), `{`, `}`, `[`, `]`, `<`, `>`, `\`, `:`, `|`. В случае наличия таких символов Arenadata DB Backup Manager не сможет снять бэкап/восстановить базы данных.
11. При установке ADB бандл устанавливает необходимые права для конкретных файлов и директорий, создаваемых им, независимо от значения `umask` на хостах кластера.
12. В процессе установки вносятся следующие изменения на уровне операционной системы. Не рекомендуется изменять перечисленные ниже объекты вручную.
    - Изменяются системные файлы:
      - */etc/hosts* (опционально)
      - */etc/hostname* (опционально)
      - */etc/chrony.conf* (при установке Chrony)
      - */etc/ssh/sshd_config* (всегда)
      - */etc/security/limits.conf* (всегда)
    - Создается файл сервиса настройки cgroup v2 для ADB — */etc/systemd/system/cgroup_v2_gpdb.service*;
    - Создаются новые файлы в директории */sys/fs/cgroup*.
13. В процессе установки ADB создается новый файл в */var/spool/cron/* для системного пользователя ADB (по умолчанию `gpadmin`). Доступ к изменению этого файла `crontab -e -u gpadmin` должен быть также у всех технических учетных записей. ADB добавляет в crop задачи по запуску скриптов для обслуживания кластера – изменять их расписание и аргументы нужно с помощью соответствующего раздела в конфигурационных параметрах ADB.
14. В случае online-установки ADB (с доступом в Интернет) все хосты должны иметь доступ к репозиториям Arenadata, перечисленным на вкладке Configuration в меню кластера ADB web-интерфейса ADCM (см. поля ***Arenadata repo url*** и ***Monitoring repo url***). Для offline-установки ADB необходимо обеспечить доступ к хосту, на котором развернут кластер Enterprise Tools.
15. Для обоих видов установки (online и offline) необходимо обеспечить доступ к официальным репозиториям соответствующей операционной системы со всех хостов кластера (или создать локальные зеркала):
    - AltLinux: ftp.altlinux.org
    - Astra Linux: download.astralinux.ru
    - РЕД ОС: repo.red-soft.ru и repo1.red-soft.ru
    - Ubuntu: archive.ubuntu.com/ubuntu

Требования для Astra Linux и Ubuntu

1. ADB, подобно иным аналитическим СУБД, оптимизирована для работы с generic-ядрами. Использование ядер kwialsency может не обеспечить оптимальную производительность для задач, характерных для аналитической обработки данных. По этой причине для корректной работы ADB рекомендуется использовать исключительно generic-ядра.
2. Необходимо учитывать совместимость версий.

ПРИМЕЧАНИЕ

ADB 6 была основана на механизме cgroup v1. Начиная с версии 7 ADB основана на cgroup v2, который используется по умолчанию во всех поддерживаемых операционных системах.

Предварительная настройка ОС

Ниже приведен список настроек операционной системы, которые необходимо выполнить для корректной работы ADB. Вы можете подготовить окружение вручную на всех хостах ADB или выбрать соответствующие конфигурационные параметры сервиса ADB, чтобы автоматически применить указанные настройки во время установки ADB.

Перед установкой ADB запустите действие Precheck, которое в том числе проверяет параметры операционной системы хостов.

Требование	Название параметра в сервисе ADB

Файл /etc/hosts содержит доменные имена всех хостов кластера: мастера, резервного мастера и хостов и всех сегментов. Например:
`192.168.1.10 mdw.example.com`
`192.168.1.20 smdw.example.com`
`192.168.1.30 sdw1.example.com`
`192.168.1.40 sdw2.example.com`
Для сопоставления доменных имен также можно использовать DNS
	Edit /etc/hosts

Отключена система Security Enhanced Linux (SELinux)	***Disable linux security module*** (изменение параметра требует перезапуска хостов)
IPv6 отключен	***Disable IPv6***
Значение параметра `MaxStartups` в sshd равно `1000:30:1022`	***Set up MaxStartups for sshd***
Отключена функция Transparent Huge Pages в ядре Linux	***Set up GRUB for DBMS*** (изменение параметра требует перезапуска хостов)
В файле /etc/security/limits.conf для пользователя базы данных установлены следующие лимиты: - Мягкое и жесткое ограничение на количество открытых файловых дескрипторов равно `524288`. - Мягкое и жесткое ограничение на количество процессов равно `150000`.	***Set up ulimits for DBMS user***
Firewall отключен на хостах ADB	***Disable firewall***

В ядре применены следующие настройки (просмотреть все настройки ядра можно командой `sysctl -a`):
`kernel.core_pipe_limit = 0`
`kernel.core_uses_pid = 1`
`kernel.msgmax = 65536`
`kernel.msgmnb = 65536`
`kernel.msgmni = 2048`
`kernel.sem = 250 2048000 200 8192`
`kernel.shmmni = 4096`
`kernel.sysrq = 1`
`net.core.netdev_max_backlog = 10000`
`net.core.rmem_max = 2097152`
`net.core.wmem_max = 2097152`
`net.ipv4.conf.all.arp_filter = 1`
`net.ipv4.conf.default.accept_source_route = 0`
`net.ipv4.ip_local_port_range = 12000 65535`
`net.ipv4.ipfrag_high_thresh = 41943040`
`net.ipv4.ipfrag_low_thresh = 31457280`
`net.ipv4.ipfrag_time = 60`
`net.ipv4.tcp_max_syn_backlog = 4096`
`net.ipv4.tcp_syncookies = 1`
`vm.dirty_background_bytes = 1610612736`
`vm.dirty_background_ratio = 0`
`vm.dirty_bytes = 4294967296`
`vm.dirty_expire_centisecs = 500`
`vm.dirty_ratio = 0`
`vm.dirty_writeback_centisecs = 100`
`vm.overcommit_memory = 2`
`vm.overcommit_ratio = 95`
`vm.swappiness = 10`
`vm.zone_reclaim_mode = 0`
	Sysctl parameters

ВАЖНО

При изменении параметров ***Disable linux security module*** и ***Set up GRUB for DBMS*** требуется перезагрузка хостов. Поэтому:

- Для применения настроек на хостах кластера во время установки — при выполнении кластерного действия ***Install*** или ***Reinstall*** установите флаг ***Reboot cluster servers after installation***.
- Для применения настроек на хостах, добавляемых в сервис ADB:
  - При выполнении кластерного действия ***Move Host Segment*** установите флаг ***Reboot new segment host after installation***.
  - При выполнении кластерных действий ***Expand*** или ***Init Standby Master*** установите флаг ***Reboot new servers after installation***.

В противном случае эти настройки не применятся к операционной системе хостов. Если по какой-то причине флаг ***Reboot cluster servers after installation*** не был установлен, рекомендуется остановить кластер и перезапустить хосты вручную, чтобы применились указанные настройки.

Установка

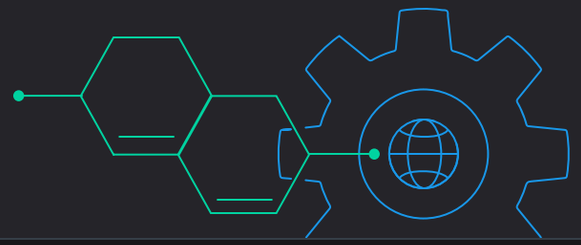
Назад к содержанию

Этот раздел содержит описание установки кластера Arenadata DB (ADB). Для установки рекомендуется использовать Arenadata Cluster Manager (ADCM) — отдельный продукт Arenadata, предназначенный для простого, удобного и быстрого развертывания и дальнейшей эксплуатации программного обеспечения.

Возможны два способа установки ADB через ADCM.

Online-установка

на хостах с доступом к Интернету



Offline-установка

на хостах без доступа к Интернету

Online-установка

Назад к содержанию

Этот раздел посвящен **online**-установке ADB. Данный способ установки подходит для хостов с доступом к Интернету.

Процесс online-установки через ADCM включает следующие шаги.

Шаг 1.

Установка ADCM

Шаг 2.

Подготовка хостов

Шаг 3.

Установка кластера ADB

Установка АСПК

Язык: Русский

Меню / содержание

Содержание

Шаг 1. Предварительная подготовка
Выбор версии и обзор системных требований
Шаг 2. Установка АСПК
Установка Docker
Установка АСПК
Шаг 3. Запуск АСПК
Первичная настройка после развертывания
Проверка доступности и работоспособности сервисов
Проверка доступности и работоспособности сервисов
Проверка веб-интерфейсов АСПК
Шаг 4. Первичная настройка
Первичная настройка Keycloak и интеграция с АСПК
Первичная настройка Keycloak и интеграция с АСПК
Шаг 5. Обновление АСПК
Online обновление
Offline обновление
Offline обновление

Шаг 1. Предварительная подготовка

Выбор версии и обзор системных требований

Для корректной работы платформы АСПК необходимо выбрать версию и развернуть соответствующий ей набор сервисов.

Актуальная версия — 1.0.0.

Для выбора версии и ознакомления с системными требованиями:

1. Перейдите в раздел Релизы.
2. Выберите версию, которая соответствует вашему сценарию развертывания.
3. Ознакомьтесь с системными требованиями выбранной версии.

Docker

Docker — платформа для разработки, развертывания и запуска приложений в изолированных контейнерах. Все компоненты АСПК развертываются как контейнеры Docker, поэтому перед установкой платформы необходимо установить Docker на сервер.

Docker Desktop — графическая версия Docker для Windows и macOS, включающая Docker Engine, Docker CLI и Docker Compose. Подходит для разработчиков и тестовых стендов.

Docker Engine — серверная версия Docker для Linux. Используется на production-серверах.

Docker CLI — командная строка Docker. Позволяет управлять контейнерами, образами и сетями из терминала.

Docker Compose — инструмент для описания и запуска многоконтейнерных приложений. Все сервисы АСПК описываются в файлах docker-compose.yaml и запускаются одной командой.

Docker Hub

Docker Hub — облачный реестр образов Docker. Все образы АСПК хранятся в реестре hub.docker.com и загружаются на сервер при развертывании.

Для корректной работы платформы необходимо иметь доступ к следующим ресурсам:

- hub.docker.com — для загрузки образов АСПК
- registry-1.docker.io — для загрузки слоев образов
- auth.docker.io — для аутентификации при загрузке образов

Системные требования

Минимальные требования:

- 4 vCPU
- 8 GB RAM
- 50 GB SSD
- Ubuntu 22.04 LTS

Рекомендуемые требования:

- 8 vCPU
- 16 GB RAM
- 100 GB SSD
- Ubuntu 22.04 LTS

Требования к ПО:

- Docker Engine 24.0 и выше
- Docker Compose v2.20 и выше
- curl
- wget
- tar
- unzip

Требования к сетевому доступу:

- Доступ к hub.docker.com по HTTPS (443)
- Доступ к registry-1.docker.io по HTTPS (443)
- Доступ к auth.docker.io по HTTPS (443)
- Доступ к репозиторию дистрибутивов АСПК

Требования к файловой системе:

- /opt/ — рекомендуется выделить отдельный раздел или диск, минимум 50 GB свободного места
- /var/lib/docker/ — рекомендуется выделить отдельный раздел или диск, минимум 50 GB свободного места

Проверка выполнения требований:

```
# Проверка версии ядра
uname -r

# Проверка свободного места на диске
df -h /opt /var/lib/docker

# Проверка доступности hub.docker.com
curl -I https://hub.docker.com

# Проверка доступности registry-1.docker.io
curl -I https://registry-1.docker.io

# Проверка доступности auth.docker.io
curl -I https://auth.docker.io
```

Шаг 2. Установка АСПК

Установка Docker

Docker Engine

```
# Удаление старых версий
sudo apt-get remove docker docker-engine docker.io containerd runc

# Обновление индекса пакетов
sudo apt-get update

# Установка зависимостей
sudo apt-get install ca-certificates curl gnupg

# Добавление ключа репозитория Docker
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Добавление репозитория Docker
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list

# Обновление индекса пакетов
sudo apt-get update

# Установка Docker Engine, CLI, containerd и Compose plugin
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

# Проверка версии
docker --version
docker compose version
```

Docker CLI

```
# Проверка версии
docker --version

# Проверка статуса сервиса
sudo systemctl status docker

# Запуск сервиса
sudo systemctl start docker

# Автозапуск при перезагрузке
sudo systemctl enable docker

# Проверка работоспособности
docker info
```

Docker Compose

```
# Проверка версии
docker compose version

# Если plugin не установлен — установка вручную
sudo apt-get install docker-compose-plugin
```

Установка АСПК

Клонирование репозитория

```
# Переход в рабочий каталог
cd /opt

# Клонирование репозитория
sudo git clone https://gitlab.aspk.ru/aspk/deployment.git

# Переход в каталог развертывания
cd /opt/deployment

# Переключение на ветку релиза
sudo git checkout release/1.0.0

# Проверка ветки
sudo git branch -a
```

Настройка переменных

```
# Копирование шаблона переменных
sudo cp .env.example .env

# Редактирование переменных
sudo nano .env
```

Минимальный набор переменных для первичного развертывания:

```
# Версия релиза
RELEASE_VERSION=1.0.0

# Домен платформы
DOMAIN=aspk.local

# Пароль администратора Keycloak
KEYCLOAK_ADMIN_PASSWORD=<password>

# Пароль базы данных Keycloak
KEYCLOAK_DB_PASSWORD=<password>

# Пароль базы данных Grafana
GRAFANA_DB_PASSWORD=<password>

# Пароль базы данных Prometheus
PROMETHEUS_DB_PASSWORD=<password>

# Пароль базы данных Loki
LOKI_DB_PASSWORD=<password>

# Пароль базы данных Tempo
TEMPO_DB_PASSWORD=<password>

# Пароль базы данных Alertmanager
ALERTMANAGER_DB_PASSWORD=<password>

# Пароль базы данных Mimir
MIMIR_DB_PASSWORD=<password>

# Пароль базы данных Pyroscope
PYROSCOPE_DB_PASSWORD=<password>

# Пароль базы данных Grafana Alloy
ALLOY_DB_PASSWORD=<password>

# Пароль базы данных Grafana Image Renderer
IMAGE_RENDERER_DB_PASSWORD=<password>

# Пароль базы данных Grafana OnCall
ONCALL_DB_PASSWORD=<password>

# Пароль базы данных Grafana Incident
INCIDENT_DB_PASSWORD=<password>

# Пароль базы данных Grafana Synthetic Monitoring
SYNTHETIC_MONITORING_DB_PASSWORD=<password>

# Пароль базы данных Grafana Sift
SIFT_DB_PASSWORD=<password>

# Пароль базы данных Grafana Assistant
ASSISTANT_DB_PASSWORD=<password>

# Пароль базы данных Grafana Frontend Observability
FRONTEND_OBSERVABILITY_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Migration Assistant
CLOUD_MIGRATION_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Portal
CLOUD_PORTAL_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Provisioning
CLOUD_PROVISIONING_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Insights
CLOUD_USAGE_INSIGHTS_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Billing
CLOUD_USAGE_BILLING_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Groups
CLOUD_USAGE_GROUPS_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Alerting
CLOUD_USAGE_ALERTING_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Logs
CLOUD_USAGE_LOGS_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Metrics
CLOUD_USAGE_METRICS_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Traces
CLOUD_USAGE_TRACES_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Profiles
CLOUD_USAGE_PROFILES_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Frontend Observability
CLOUD_USAGE_FRONTEND_OBSERVABILITY_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Synthetic Monitoring
CLOUD_USAGE_SYNTHETIC_MONITORING_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Sift
CLOUD_USAGE_SIFT_DB_PASSWORD=<password>

# Пароль базы данных Grafana Cloud Usage Assistant
CLOUD_USAGE_ASSISTANT
```

Подготовка хостов

Назад к содержанию

Перед установкой нового кластера необходимо добавить информацию о его будущих хостах в ADCM. Это задача **хостпровайдеров** — специальных плагинов, отвечающих за взаимодействие между хостами и ADCM.

ADCM поддерживает несколько типов хостпровайдеров. Для получения подробной информации о том, как их настроить и использовать для создания хостов, обратитесь к документации:

- SSH
- Yandex Cloud
- VCloud (VMware)
- Cloud.ru Advanced
- CROC Cloud
- ADB Cloud Bundle

ВНИМАНИЕ

Начиная с версии 6.23.3.44 ADB поддерживает как FQDN, так и короткие имена хостов (без доменов). В предыдущих версиях ADB использование FQDN не рекомендуется (поскольку это приводит к ошибкам в ходе выполнения действия ***Expand*** и требует применения обходного решения).

Создание кластера

Дарья Барышева

Содержание

Назад к содержанию

Шаг 1. Загрузка бандла

Шаг 2. Загрузка бандла в ADCM

Шаг 3. Создание кластера на базе загруженного бандла

В ADCM **кластер** означает набор сервисов, запущенных на группе хостов.

Дистрибутивы кластеров для ADCM поставляются в виде бандлов. В общем случае бандл представляет обычный архив, описывающий логику развертывания кластера.

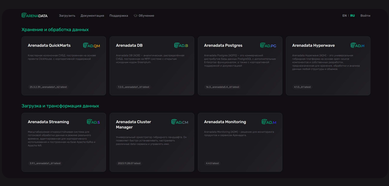
Шаги для добавления нового кластера приведены ниже.

ПРИМЕЧАНИЕ

При использовании Enterprise-версии продукта следует пропустить шаг 1, получить бандл у команды поддержки Arenadata и перейти к шагу 2.

Шаг 1. Загрузка бандла

1. Перейдите на сайт https://network.arenadata.io/ и выберите продукт ***Arenadata DB***.



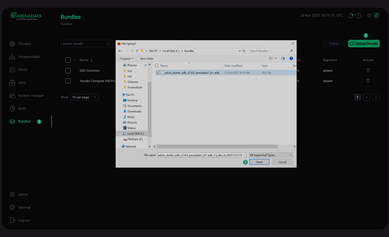
Переход к загрузке дистрибутивов

2. На следующей странице загрузите бандл для нужной версии ADB.

Шаг 2. Загрузка бандла в ADCM

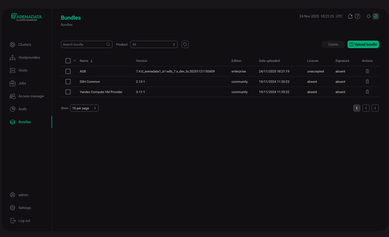
Чтобы загрузить бандл в ADCM:

1. Выберите пункт ***Bundles*** в левом навигационном меню и нажмите ***Upload bundle***.
2. Выберите бандл в открывшемся диалоговом окне.



Переход к загрузке бандла

3. После успешной загрузки бандл отображается на странице ***Bundles***.



Результат успешной загрузки бандла

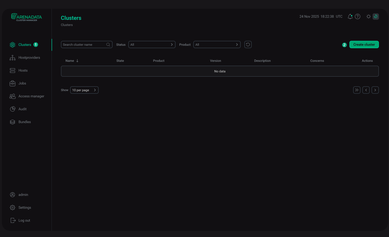
Шаг 3. Создание кластера на базе загруженного бандла

ВАЖНО

На этапе создания кластера через web-интерфейс ADCM генерируется только сам кластер. Это означает добавление записи о кластере в ADCM — сервисы на этом шаге еще не устанавливаются.

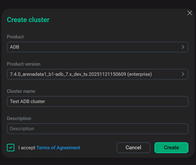
Для создания кластера необходимо выполнить шаги:

1. Выберите пункт ***Clusters*** в левом навигационном меню и нажмите ***Create cluster***.



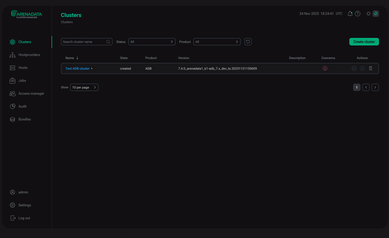
Переход к созданию кластера

2. В открывшемся окне требуется описать новый кластер:
   - Выберите загруженный бандл в поле ***Product***.
   - Выберите версию бандла в поле ***Product version***. Несколько версий становятся доступны после загрузки нескольких бандлов для одного типа продуктов.
   - Введите имя кластера в поле ***Cluster name***.
   - При необходимости добавьте описание кластера в поле ***Description***.
   - Установите флаг ***I accept Terms of Agreement*** для подписания лицензионного соглашения. Флаг отображается только для бандлов версии Enterprise. Чтобы ознакомиться с текстом соглашения, перейдите по ссылке ***Terms of Agreement***. Подписать соглашение также возможно на странице ***Bundles***, нажав на соответствующий продуктовый бандл.
   - Нажмите ***Create***.



Создание кластера

3. После успешного добавления кластер отображается на странице ***Clusters***.



Результат успешного добавления кластера

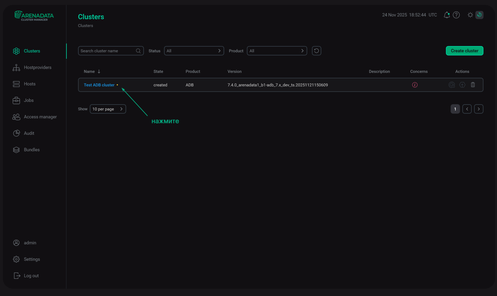
Добавление сервисов

Дарья Барышева

Назад к содержанию

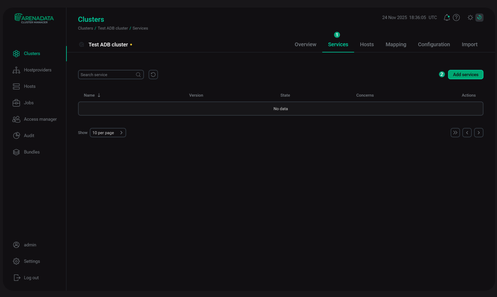
В ADCM **сервис** означает программное обеспечение, выполняющее некоторую функцию. Примеры сервисов в кластерах ADB: ADB, ADB Control, Chrony и так далее. Для добавления сервисов в кластер:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



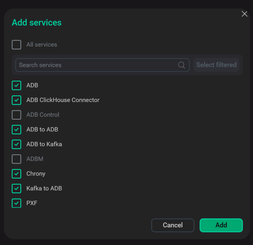
Выбор кластера

2. Откройте вкладку ***Services*** на странице кластера и нажмите ***Add service***.



Переход к добавлению сервисов

3. В открывшемся окне выберите сервисы для добавления в кластер и нажмите ***Add***.



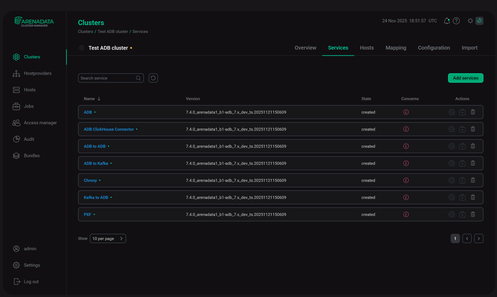
Выбор сервисов

Краткое описание доступных сервисов приведено ниже. `CE` означает Common-версию ADB (Common Edition), `EE` — Enterprise-версию ADB (Enterprise Edition).

Сервисы, которые могут быть добавлены в кластер ADB

Сервис	Обязательность	Версия ADB	Назначение
ADB	Да	CE/EE	Основной сервис, соответствующий MPP-системе на базе Greengage DB
Chrony	Нет	CE/EE	Сервис времени. Обеспечивает синхронизацию часов на хостах кластера, что необходимо для его корректной работы. Установка Chrony не требуется, если на хостах уже используется аналогичный сервис
PXF	Нет	CE/EE	Сервис, обеспечивающий взаимодействие ADB с внешними источниками данных
ADB ClickHouse Connector	Нет	EE	ADB ClickHouse Connector — сервис, обеспечивающий двустороннее взаимодействие ADB с ClickHouse. Требует установки PXF
ADB to Kafka	Нет	EE	ADB to Kafka Connector — сервис, обеспечивающий одностороннее взаимодействие ADB с Kafka. Требует установки PXF
Kafka to ADB	Нет	EE	Kafka to ADB Connector — сервис, обеспечивающий одностороннее взаимодействие Kafka с ADB
ADB to ADB	Нет	EE	ADB to ADB Connector — сервис, обеспечивающий двустороннее взаимодействие между кластерами ADB
ADBM	Нет	EE	Arenadata DB Backup Manager — сервис для управления бинарными бэкапами ADB. Может быть установлен только offline
ADB Control	Нет	EE	Arenadata DB Control (ADB Control) — сервис для мониторинга, диагностики и управления запросами, запущенными в кластере ADB. Может быть установлен только offline

4. В результате успешно добавленные сервисы отображаются на вкладке ***Services***.



Результат успешного добавления сервисов в кластер

ПРИМЕЧАНИЕ

Опциональные сервисы могут быть добавлены в кластер позднее. Процесс добавления сервисов в уже развернутый кластер не отличается от описанного выше.

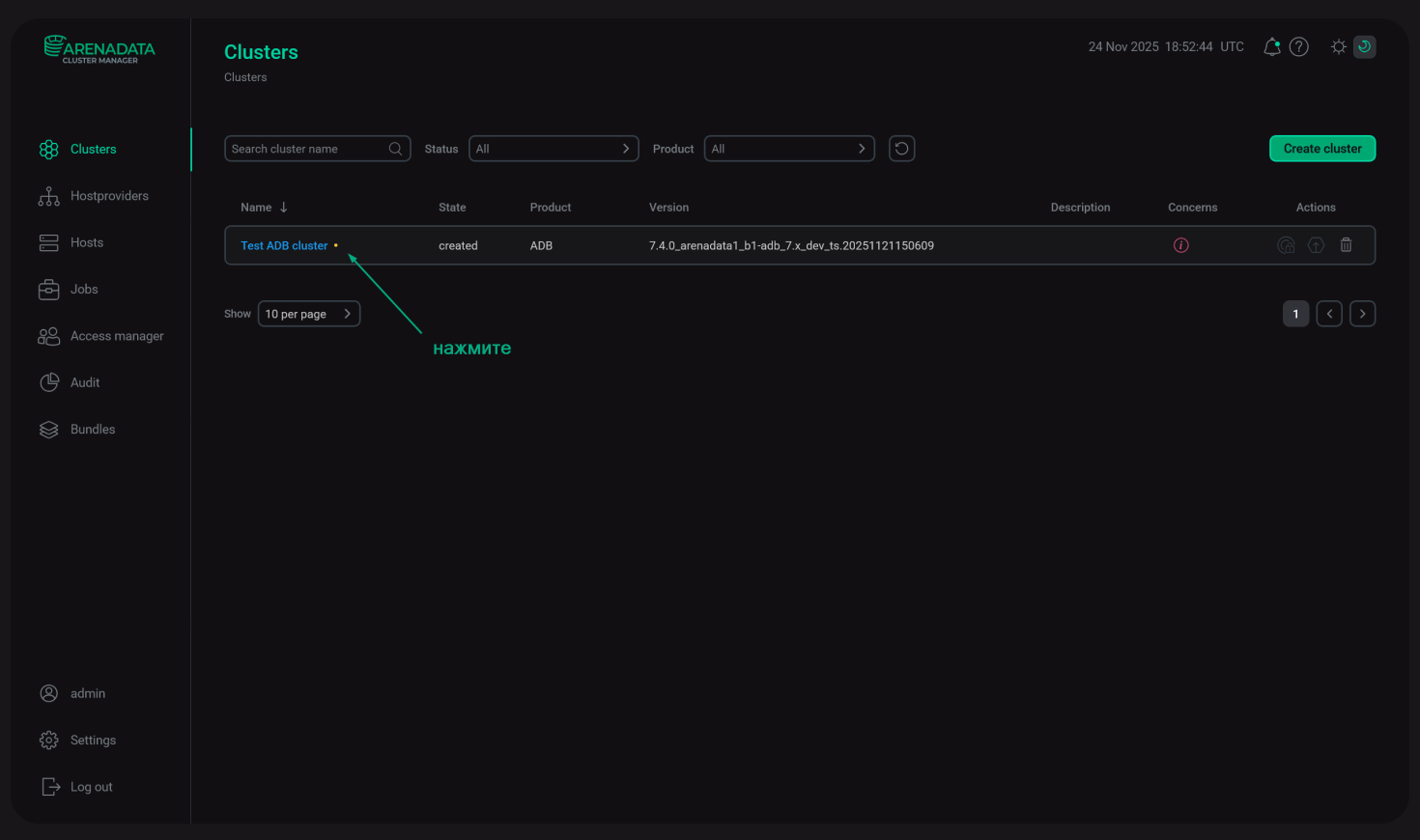
Добавление хостов в кластер

Дарья Барышева

Назад к содержанию

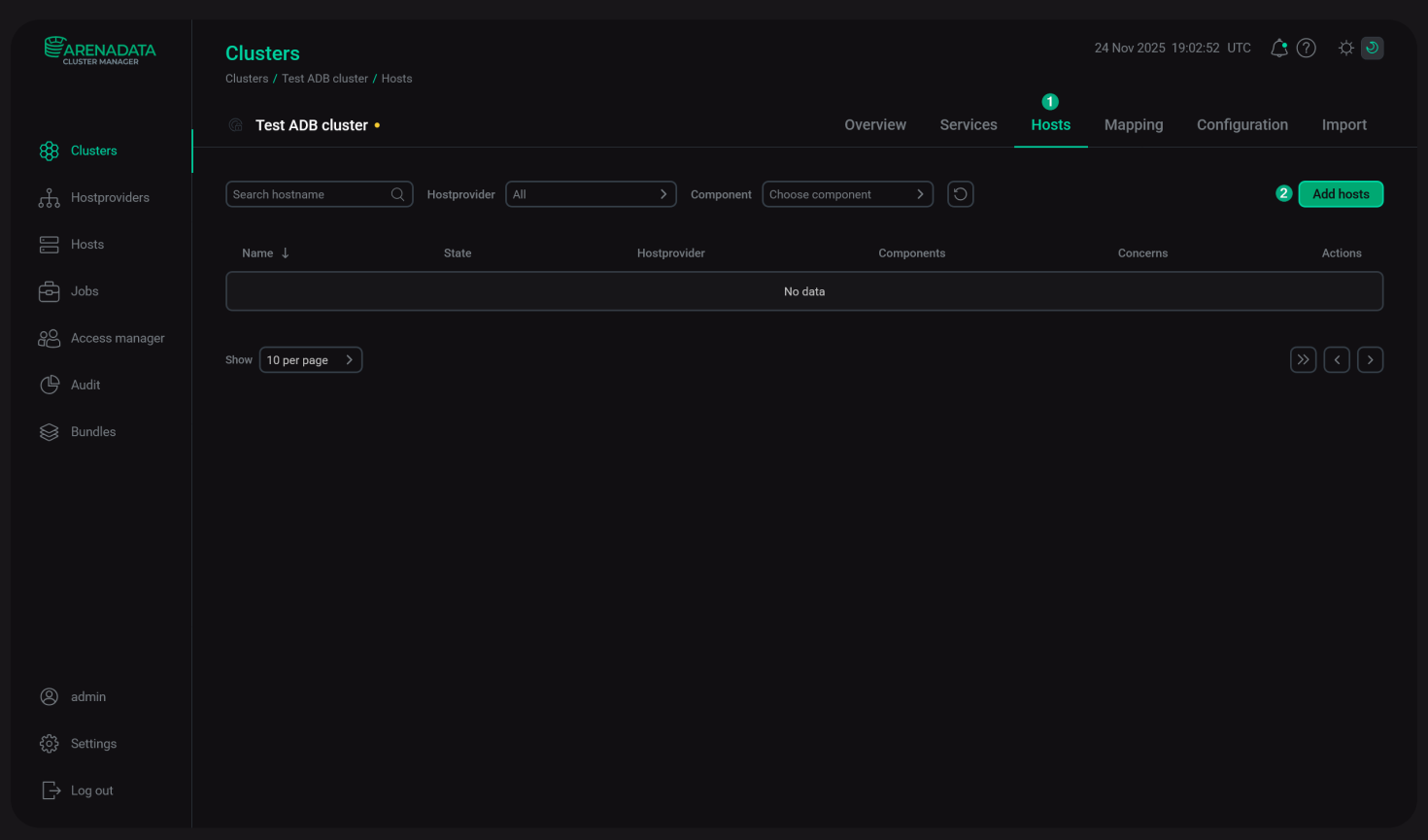
Для добавления хостов в кластер:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



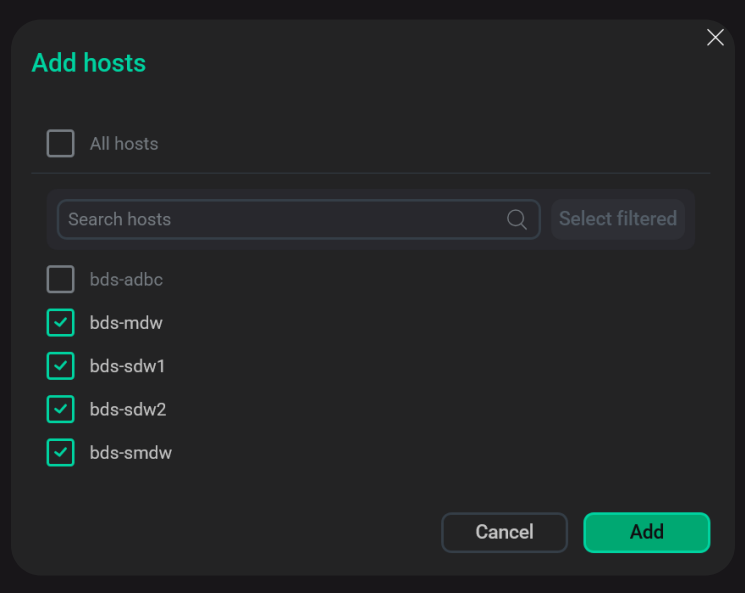
Выбор кластера

2. Откройте вкладку ***Hosts*** на странице кластера и нажмите ***Add hosts***.



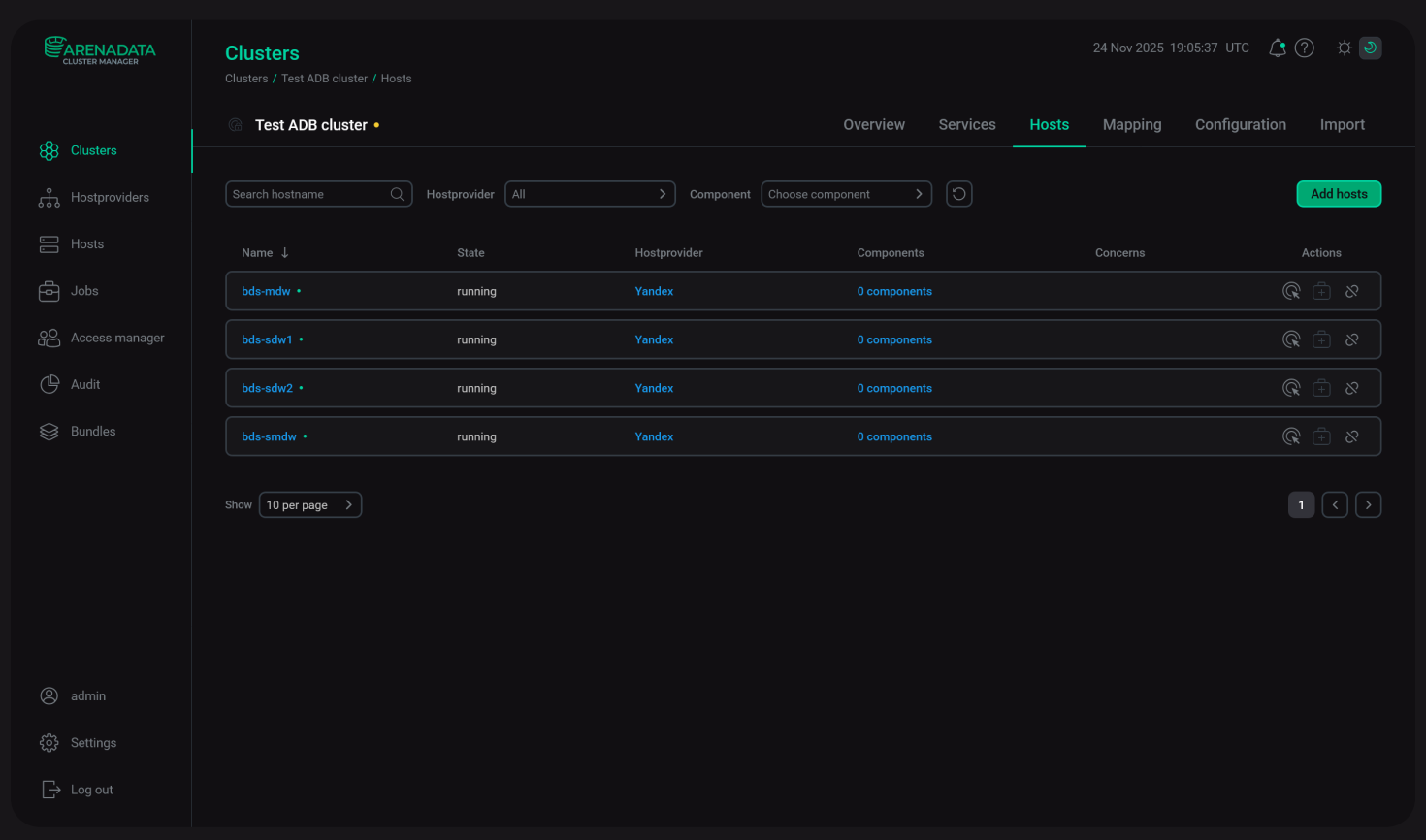
Переход к добавлению хостов

3. В открывшемся окне выберите хосты для добавления в кластер и нажмите ***Add***.



Выбор хостов

4. В результате успешно добавленные хосты отображаются на вкладке ***Hosts***.



Результат успешного добавления хостов в кластер

Добавление компонентов

Дарья Барышева

Назад к содержанию

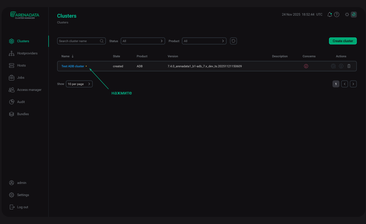
В ADCM **компонент** означает часть сервиса, которая должна быть развернута на одном или нескольких хостах кластера. Например, сервис ADB состоит из таких компонентов, как ADB Master, ADB Segment и ADB Standby.

ВАЖНО

Изначально на хостах кластера нет ни одного компонента — распределение компонентов между хостами должно быть проведено вручную.

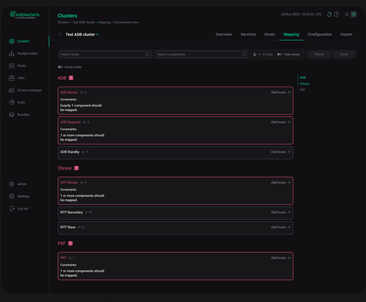
Для добавления компонентов на хосты кластера:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



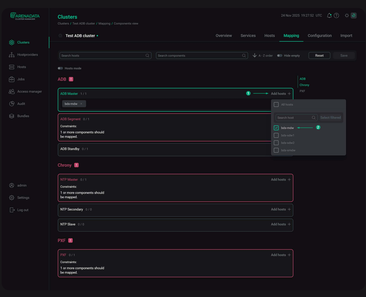
Выбор кластера

2. Откройте вкладку ***Mapping*** на странице кластера.



Переход к распределению сервисных компонентов

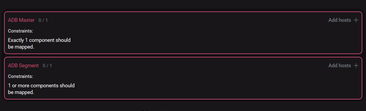
3. Распределите все компоненты между хостами кластера. Для назначения хостов компоненту нажмите ***Add hosts*** и выберите хосты в открывшемся окне.



Выбор хостов для компонента

Обязательные компоненты

Каждый сервис может иметь обязательные и опциональные компоненты. Обязательные подсвечиваются красным цветом. Также в интерфейсе ADCM для них выводится информация о том, в каком количестве необходимо установить каждый компонент.



Обязательные компоненты

ВНИМАНИЕ

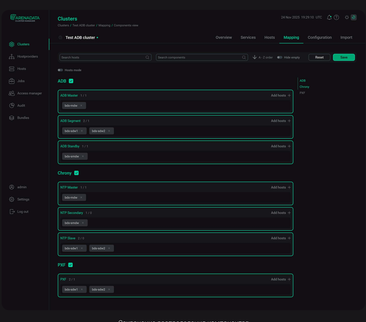
Без назначения нужного числа хостов обязательным компонентам сохранение общей схемы распределения невозможно.

Типовая схема распределения компонентов в кластере ADB приведена в таблице ниже. Она основана на Enterprise-версии ADB.

Распределение компонентов ADB

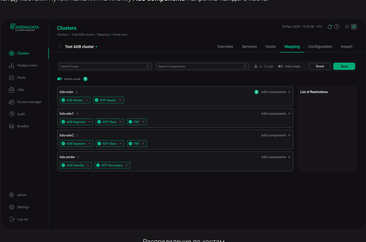
Хост	Сервис	Компонент	Примечание
Хост координатора	ADB	ADB Master	Необходимо установить на координатор-хосте
	Chrony	NTP Master	Необходимо установить на хосте координатор
Хост резервного координатора (standby)	ADB	ADB Standby	Необходимо установить на хосте резервного координатора (standby)
	Chrony	NTP Secondary	Необходимо установить на хосте резервного координатора (standby)
Сегмент-хост (1..N)	ADB	ADB Segment	Необходимо установить на каждом сегмент-хосте
	Chrony	NTP Slave	Необходимо установить на каждом сегмент-хосте
	PXF	PXF	Необходимо установить на сегмент-хостах и — опционально — на координатор-хосте

4. После завершения распределения компонентов нажмите ***Save***.



Сохранение распределения компонентов

Начиная с версии 2.2.0 ADCM поддерживает альтернативный способ сопоставления — по хостам. Для этого переведите в активное положение переключатель ***Hosts mode*** и распределите компоненты между хостами путем нажатия на кнопку ***Add components*** напротив каждого хоста.



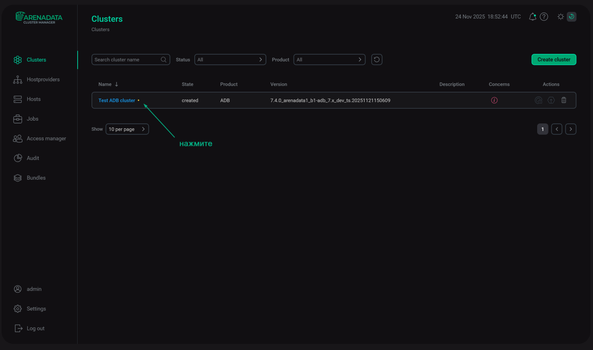
Распределение по хостам

Настройка сервисов

Дарья Барышева

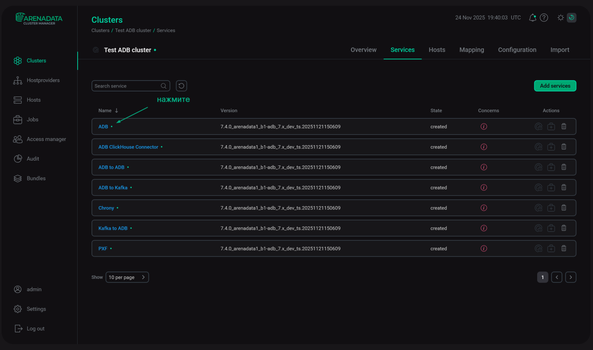
Назад к содержанию

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



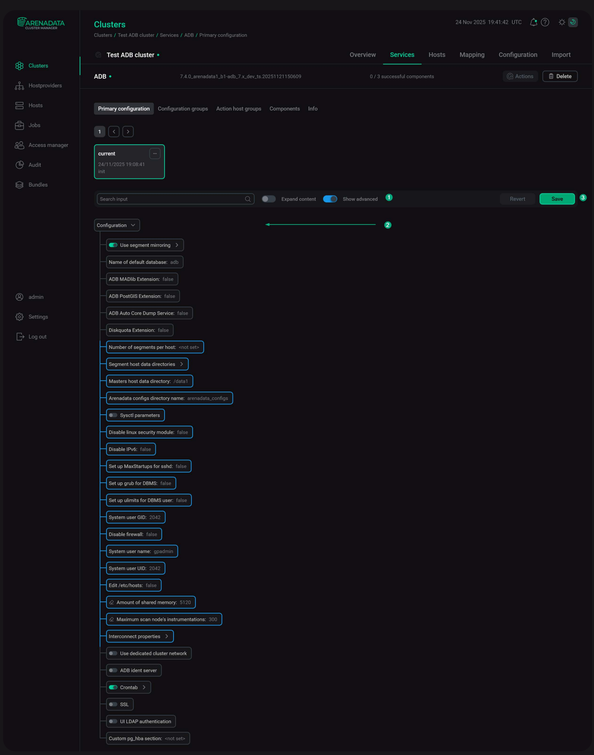
Выбор кластера

2. Откройте вкладку ***Services*** на странице кластера. Нажмите на имя сервиса в столбце ***Name***.



Переход к настройке сервиса

3. В открывшемся окне:
   - Откройте вкладку ***Primary Configuration***.
   - Переведите в активное состояние переключатель ***Show advanced***.
   - Заполните конфигурационные параметры выбранного сервиса. Поля, подсвеченные красным, обязательны для заполнения.
   - Нажмите ***Save***.



Настройка сервиса

Обратите внимание на следующие параметры:

- ***ADB → Edit /etc/hosts***. Установите значение `true`, чтобы сервис ADB мог добавить имена хостов кластера в файл */etc/hosts*. Если значение — `false`, требуется вручную внести изменения в этот файл или настроить DNS-сервер. Через настройки сервиса ADB вы можете изменить и другие параметры хостов, чтобы привести их к рекомендуемым значениям.
- ***ADB → Disable linux security module*** и ***ADB → Set up GRUB for DBMS***. Изменение этих параметров требует перезапуска хостов, поэтому рекомендуется установить для них значение `true` до установки кластера, а затем установить значение `true` для параметра ***Reboot cluster servers after installation*** в кластерном действии Install, чтобы хосты перезапустились автоматически. Если менять их значения после установки, потребуется остановить кластер и перезапустить хосты вручную.
- ***Chrony → NTP servers***. Хотя этот параметр необязателен, мы рекомендуем установить в нем адрес ближайшего NTP-сервера (например, используемого в вашей компании).

ПРИМЕЧАНИЕ

Более подробную информацию по всем конфигурационным параметрам можно получить в статье Конфигурационные параметры.

Содержание

Перед содержанием

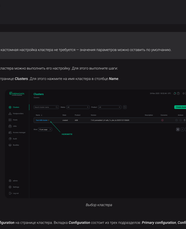
Primary configuration
Configuration groups
Ansible settings

ВАЖНО

В большинстве случаев настроек кластера по умолчанию — значения параметров можно оставить по умолчанию.

После добавления нового кластерного менеджера выполните его настройку. Для этого выполните шаги:

1. Выберите кластер на странице **Clusters**. Для этого нажмите на имя кластера в списке **Name**.



Выбор кластера

2. Откройте вкладку **Configuration** на странице кластера. Вкладка **Configuration** состоит из трех подразделов: **Primary configuration**, **Configuration groups**, **Ansible settings**.
3. Заполните необходимые конфигурационные параметры на одной вкладке и нажмите **Save**.

Primary configuration

Вкладка **Configuration** → **Primary configuration** содержит основные конфигурационные параметры кластера.

ВНИМАНИЕ!

Используйте кастомные инстанции *Use existing docker*, *Use custom repmgr*, *Use custom JRA_HOME for cluster* с осторожностью.

Доступные для редактирования параметры:

- *Cluster UUID* — уникальный идентификатор кластера ADB, формируемый автоматически при его установке.
- *Timeout in which package manager* — параметр этой опции предназначен для настройки тайм-аута ожидания разблокировки пакетного менеджера (RF: 8 Ubuntu после перезагрузки в фоновом обновлении, по умолчанию 2 Ожидание (RF

Параметр	Описание	По умолчанию
retries	Максимальное число повторных попыток	3
delay	Тайм-аут между повторными попытками (в секундах)	10

- Пулы репозиториев, которые будут использованы в процессе установки в зависимости от операционной системы хостов (RE.Linux, Astra Linux, РЕД ОС, Ubuntu), на основе которых регистрируются только стандартные репозитории

Altlinux 10 repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

Astra linux repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

redos 7 repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

Ubuntu 22.04 repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

- *Use existing docker* — флаг, указывающий на необходимость использования существующей инстанции Docker в процессе установки пакетов ADB (репрезиз, ADB Control, ADBM). Отображается, если установлен флаг **Use advanced** в верхней части формы. Если выражение перечислено в списке для исключения поля **JRA_HOME**, то все требуемые ресурсы будут использованы. Для выражения по умолчанию (если выражение не перечислено) в бинарном файлом (exe, exe.config) скопируются новый файл. При отсутствии файла по умолчанию или отсутствии совместимости с ошибкой. Дополнительная информация о требованиях к для установки Docker см. в разделе «Требования к Docker» настоящего руководства.
  При пустом значении поля **JRA_HOME** для поиска Java используется по умолчанию один из стандартных путей в зависимости от операционной системы.

Ubuntu
/usr/lib/jvm/java-17-openjdk-amd64/
РЕД ОС
/usr/lib/jvm/jre-17-openjdk/
Astra linux
/usr/lib/jvm/jre-17/

ВАЖНО!

- Если вы планируете использовать собственную Java (например, от Azul), для установки требуется Java версии 17.
- В Astra Linux запоминание поля **JRA_HOME** обнаружено начиная с версии 1.0.1.1.1

Установка кластера

Время прочтения: 15 мин

Содержание

Назад к содержанию

Шаг 1. Предварительная проверка (precheck)

Шаг 2. Запуск установки

- Установка всего кластера
- Установка отдельных сервисов

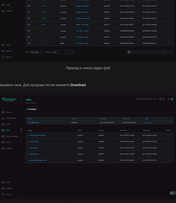
Шаг 3. Проверка статуса кластера после установки

Шаг 4. Проверка создания базы данных по умолчанию

Шаг 1. Предварительная проверка (precheck)

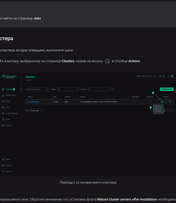
Перед установкой нового кластера рекомендуется выполнить его предварительную проверку — **Precheck**. Проверка позволит убедиться в том, что все необходимые репозитории доступны, в кластере нет хостов с одинаковыми IP-адресами и все прочие требования соблюдены. Для запуска проверки:

1. Проверьте доступ при **Precheck** к кластеру, выбранному на странице **Clusters**, нажав на иконку в столбце **Actions**.



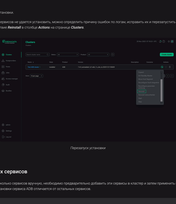
Запуск предварительной проверки кластера prechecks

2. Подтвердите действие в открывшемся окне.
3. Откройте страницу **Jobs** и дождитесь завершения проверки. После этого нажмите по первой строке в списке задач (job).



Переход к списку задач (job)

4. Просмотрите логи в открывшемся окне. Для загрузки логов нажмите **Download**.



Просмотр логов задачи

Если в логе содержится ошибка, связанная с конфигурацией кластера (например, отмечаются в IP4 или параметрах sysctl), убедитесь, что была выполнена предварительная подготовка серверов.

Шаг 2. Запуск установки

Существует два способа установки:

1. **Установка всего кластера**. В этом случае все сервисы кластера устанавливаются автоматически друг за другом после применения действия **Install** к кластеру.
2. **Установка отдельных сервисов**. В этом методе каждый сервис устанавливается вручную путем применения к нему одного или нескольких действий. Этот способ также может использовать для установки новых сервисов в уже развернутый кластер.

В обоих методах каждый сервис запускается автоматическим после установки.

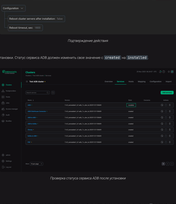
ПРИМЕЧАНИЕ

В случае ошибок логи можно найти на странице **Jobs**.

Установка всего кластера

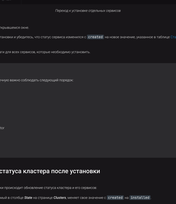
Чтобы установить все сервисы кластера за одну операцию, выполните шаги:

1. Проверьте доступ при **Install** к кластеру, выбранному на странице **Clusters**, нажав на иконку в столбце **Actions**.



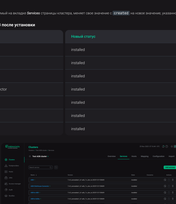
Переход к установке всего кластера

2. Подтвердите действие в открывшемся окне. Обратите внимание, что установка для **Robot cluster servers after installation** необходима для гарантируется системы после завершения инсталляции. Для этого сервера после установки потребуется выполнить вручную.



Подтверждение действия

3. Дождитесь завершения установки.
4. Если при или нескольких сервисов не удается установить, можно определить причину ошибки по логам, исправить их и перезапустить установку. Для этого необходимо выбрать действие **Install** в столбце **Actions** на странице **Clusters**.



Перезапуск установки

Установка отдельных сервисов

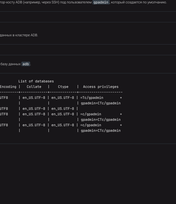
Чтобы установить один или несколько сервисов вручную, необходимо предварительно добавить его сервисы в кластер и затем применить к ним действия, описанные в подразделе ниже. Процесс установки сервиса ADB отличается от остальных сервисов.

ВНИМАНИЕ

Обратите внимание, что сервисы устанавливаются по очереди друг за другом. Не пытайтесь, установить следующий сервис до окончания установки предыдущего.

Сервис ADB

1. Откройте вкладку **Services** на странице кластера. В строке соответствующей сервису ADB, нажмите на иконку в столбце **Actions** и выберите действие **Install ADB** из выпадающего списка.



Переход к установке ADB

2. Подтвердите действие в открывшемся окне.

Подтверждение действия

3. Дождитесь завершения установки. Статус сервиса ADB должен изменить свое значение с `Installed` → `Installed`.

Проверка статуса сервиса ADB после установки

Другие сервисы

1. Откройте вкладку **Services** на странице кластера. Для сервиса, который должен быть установлен, нажмите на иконку в столбце **Actions** и выберите действие **Install** из выпадающего списка.

Переход к установке отдельных сервисов

2. Подтвердите действие в открывшемся окне.
3. Дождитесь завершения установки. Статус сервиса изменится с `Installed` на новое значение, указанное в таблице статусы сервисов ADB после установки.
4. Повторите предыдущие шаги для всех сервисов, которые необходимо установить.

ВАЖНО

При установке сервисов вручную важно соблюдать следующий порядок:

1. ADB
2. Chrony
3. KDB to ADB
4. PXF
5. ADB Clickhouse Connector
6. KDB to Kafka
7. Kafka to ADB

Шаг 3. Проверка статуса кластера после установки

В результате успешной установки произходит обновление статуса кластера и его сервисов:

- Статус кластера отображенный в столбце **State** на странице **Clusters**, меняет свое значение с `Installed` → `Installed`.

Статус кластера после успешной установки

- Статус сервисов, отображенный на вкладке **Services** страницы кластера, меняет свое значение с `Installed` на новое значение, указанное в таблице ниже:

Статусы сервисов ADB после установки

Сервис	Новый статус
ADB	installed
Chrony	installed
PXF	installed
ADB Clickhouse Connector	installed
ADB to ADB	installed
ADB to Kafka	installed
Kafka to ADB	installed

Статус сервисов кластера после успешной установки

Шаг 4. Проверка создания базы данных по умолчанию

Независимо от выбранного способа установки, в результате установки сервиса ADB автоматически создается base database с именем, указанным в конфигурационном параметре **Name of default database** сервиса ADB перед установкой (по умолчанию - **adb**). Впоследствии на нее будет использоваться в качестве базы данных по умолчанию для клиентских подключений (например, при установке расширений с помощью create ext_name from ext).

Чтобы убедиться в создании базы данных, выполните шаги:

1. Подключитесь к координатору кластера ADB (например, через SSH) под пользователем **gpadmin**, который создается по умолчанию.

```
$ ssh -ka -- gpadmin
```

2. Получите список всех баз данных в кластере ADB:

```
psql -l
```

Вывод команды содержит базу данных **adb**:

```
                                  List of databases
   Name    | Owner  | Encoding |  Collate   |  Ctype   | Access privileges
-----------+--------+----------+------------+----------+-------------------
 adb       | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =Tc/gpadmin       +
           |        |          |            |          |  gpadmin=CTc/gpadmin
 postgres  | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =Tc/gpadmin       +
 template0 | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/gpadmin        +
           |        |          |            |          |  gpadmin=CTc/gpadmin
 template1 | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/gpadmin        +
           |        |          |            |          |  gpadmin=CTc/gpadmin
(6 rows)
```

Offline-установка

Назад к содержанию

Этот раздел посвящен **offline**-установке ADB. Данный способ установки подходит для хостов без доступа к Интернету.

Для offline-установки необходимо установить и настроить Arenadata Enterprise Tools (ET). ET — это решение, предоставляющее инфраструктуру для развертывания продуктов Аренадата в среде с ограниченным доступом в интернет. Решение является частью платформы данных Аренадата и предоставляет репозитории, необходимые для установки и обновления компонентов платформы.

Процесс offline-установки через ADCM включает следующие шаги.

ПРИМЕЧАНИЕ

- Эта функция доступна только в Enterprise-версии ADB.
- Это единственный доступный способ установки сервисов:
  - Arenadata DB Control (ADB Control) — используется в кластерах ADB для мониторинга, диагностики и управления запросами к базе данных.
  - Arenadata DB Backup Manager (ADBM) — используется для управления бинарными бэкапами ADB.

AD.CM

Шаг 1.

Установка ADCM

AD.CM

Шаг 2.

Подготовка хостов

AD.CM

Шаг 3.

Установка кластера ET

AD.CM

ARENADATA DB

Шаг 4.

Установка кластера ADB

Установка АСКМ

Назад к содержанию

Содержание

Шаг 1. Предварительные требования
Требования к ресурсам и ПО
Требования к ресурсам виртуальных машин
Требования к серверам и рабочим станциям
Требования к системам управления базами данных
Требования к ПО
Установка Java
Шаг 2. Запуск АСКМ
Шаг 3. Установка лицензии
Получение лицензии на сервере АСКМ
Получение лицензии на рабочем месте АСКМ
Установка лицензии на рабочем месте АСКМ
Шаг 4. Настройка АСКМ
Настройка доступа к Oracle 12c без прав sys
Шаг 5. Установка АСКМ
Шаг 6. Запуск АСКМ
Шаг 7. Проверка установки
Проверка статуса Oracle экземпляра
Проверка доступа к Oracle по сетевому протоколу
Проверка доступа к веб-интерфейсу АСКМ
Установка клиентского ПО АСКМ
Установка JRE, АСКМ
Шаг 8. Обновление АСКМ
Обновление
Обновление ПО АСКМ
Обновление ПО Oracle
Обновление ПО Java
Обновление ПО ОС
Обновление ПО Apache
Обновление ПО Tomcat
Обновление ПО Apache Tomcat

Подготовка хостов

Назад к содержанию

Перед установкой нового кластера необходимо добавить информацию о его будущих хостах в ADCM. Это задача **хостпровайдеров** — специальных плагинов, отвечающих за взаимодействие между хостами и ADCM.

ADCM поддерживает несколько типов хостпровайдеров. Для получения подробной информации о том, как их настроить и использовать для создания хостов, обратитесь к документации:

- SSH
- Yandex Cloud
- VCloud (VMware)
- Cloud.ru Advanced
- CROC Cloud
- ADB Cloud Bundle

Для offline-установки рекомендуется использование SSH-хостпровайдера.

ВНИМАНИЕ

Начиная с версии 6.23.3.44 ADB поддерживает как FQDN, так и короткие имена хостов (без доменов). В предыдущих версиях ADB использование FQDN не рекомендуется (поскольку это приводит к ошибкам в ходе выполнения действия ***Expand*** и требует применения обходного решения).

Создание кластера

Дарья Барышева

Содержание

Назад к содержанию

Шаг 1. Получение Enterprise-версии бандла

Шаг 2. Загрузка бандла в ADCM

Шаг 3. Создание кластера на базе загруженного бандла

В ADCM **кластер** означает набор сервисов, запущенных на группе хостов.

Дистрибутивы кластеров для ADCM поставляются в виде бандлов. В общем случае бандл представляет обычный архив, описывающий логику развертывания кластера.

Шаги для добавления нового кластера приведены ниже.

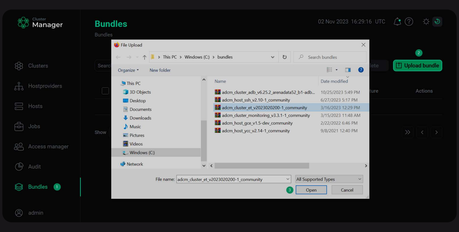
Шаг 1. Получение Enterprise-версии бандла

Пожалуйста, запросите бандл у команды поддержки Arenadata.

Шаг 2. Загрузка бандла в ADCM

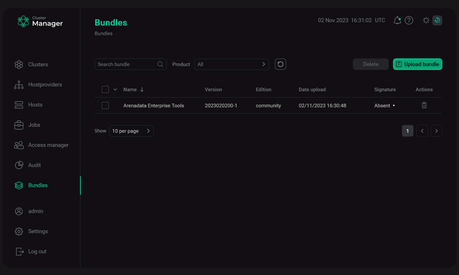
Чтобы загрузить бандл в ADCM:

1. Выберите пункт ***Bundles*** в левом навигационном меню и нажмите ***Upload bundle***.
2. Выберите бандл в открывшемся диалоговом окне.



Переход к загрузке бандла

3. После успешной загрузки бандл отображается на странице ***Bundles***.



Результат успешной загрузки бандла

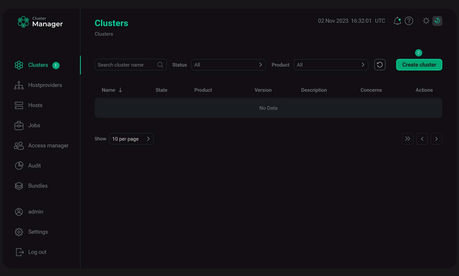
Шаг 3. Создание кластера на базе загруженного бандла

ВАЖНО

На этапе создания кластера через web-интерфейс ADCM генерируется только сам кластер. Это означает добавление записи о кластере в ADCM — сервисы на этом шаге еще не устанавливаются.

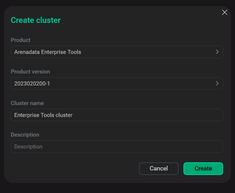
Для создания кластера необходимо выполнить шаги:

1. Выберите пункт ***Clusters*** в левом навигационном меню и нажмите ***Create cluster***.



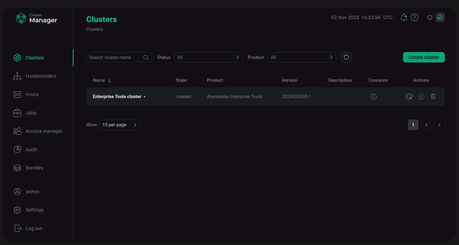
Переход к созданию кластера

2. В открывшемся окне требуется описать новый кластер:
   - Выберите загруженный бандл в поле ***Product***.
   - Выберите версию бандла в поле ***Product version***. Несколько версий становятся доступны после загрузки нескольких бандлов для одного типа продуктов.
   - Введите имя кластера в поле ***Cluster name***.
   - При необходимости добавьте описание кластера в поле ***Description***.
   - Установите флаг ***I accept Terms of Agreement*** для подписания лицензионного соглашения. Флаг отображается только для бандлов версии Enterprise. Чтобы ознакомиться с текстом соглашения, перейдите по ссылке ***Terms of Agreement***. Подписать соглашение также возможно на странице ***Bundles***, нажав на соответствующий продуктовый бандл.
   - Нажмите ***Create***.



Создание кластера

3. После успешного добавления кластер отображается на странице ***Clusters***.



Результат успешного добавления кластера

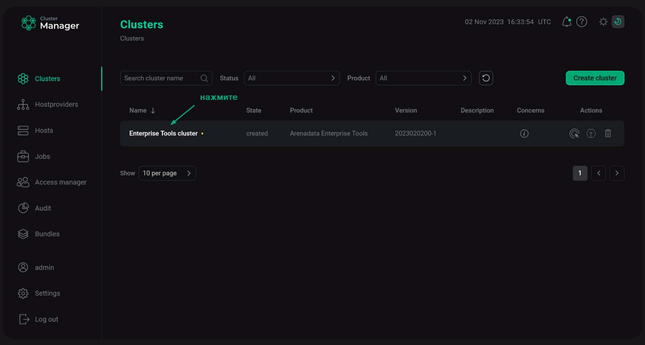
Добавление сервисов

Дарья Барышева

Назад к содержанию

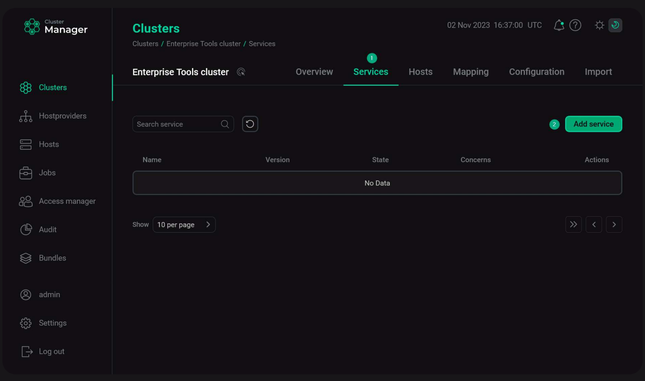
В ADCM **сервис** означает программное обеспечение, выполняющее некоторую функцию. Для добавления сервисов в кластер:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



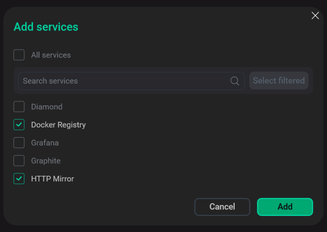
Выбор кластера

2. Откройте вкладку ***Services*** на странице кластера и нажмите ***Add service***.



Переход к добавлению сервисов

3. В открывшемся окне выберите сервисы для добавления в кластер и нажмите ***Add***.



Выбор сервисов

Краткое описание доступных сервисов приведено ниже.

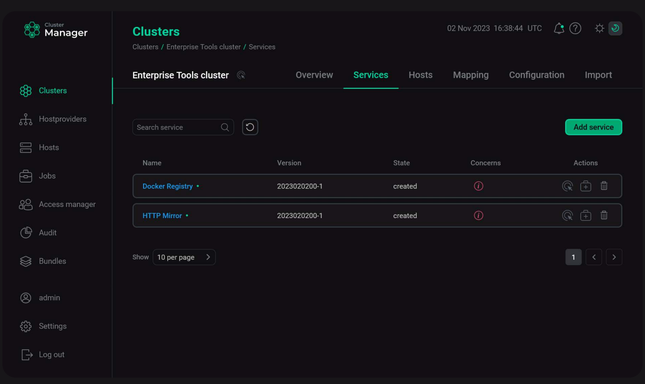
Сервисы, которые необходимо добавить в кластер Enterprise Tools

Сервис	Назначение
HTTP Mirror	Сервис, используемый для размещения пакетов репозиториев, необходимых для offline-установки продуктов
Docker Registry	Сервис, используемый для хранения образов Docker, необходимых для offline-установки продуктов

ПРИМЕЧАНИЕ

Сервис CoreDNS не используется (deprecated). Сервисы Diamond, Graphite и Grafana необходимы для настройки мониторинга — они будут описаны отдельно.

4. В результате успешно добавленные сервисы отображаются на вкладке ***Services***.



Результат успешного добавления сервисов в кластер

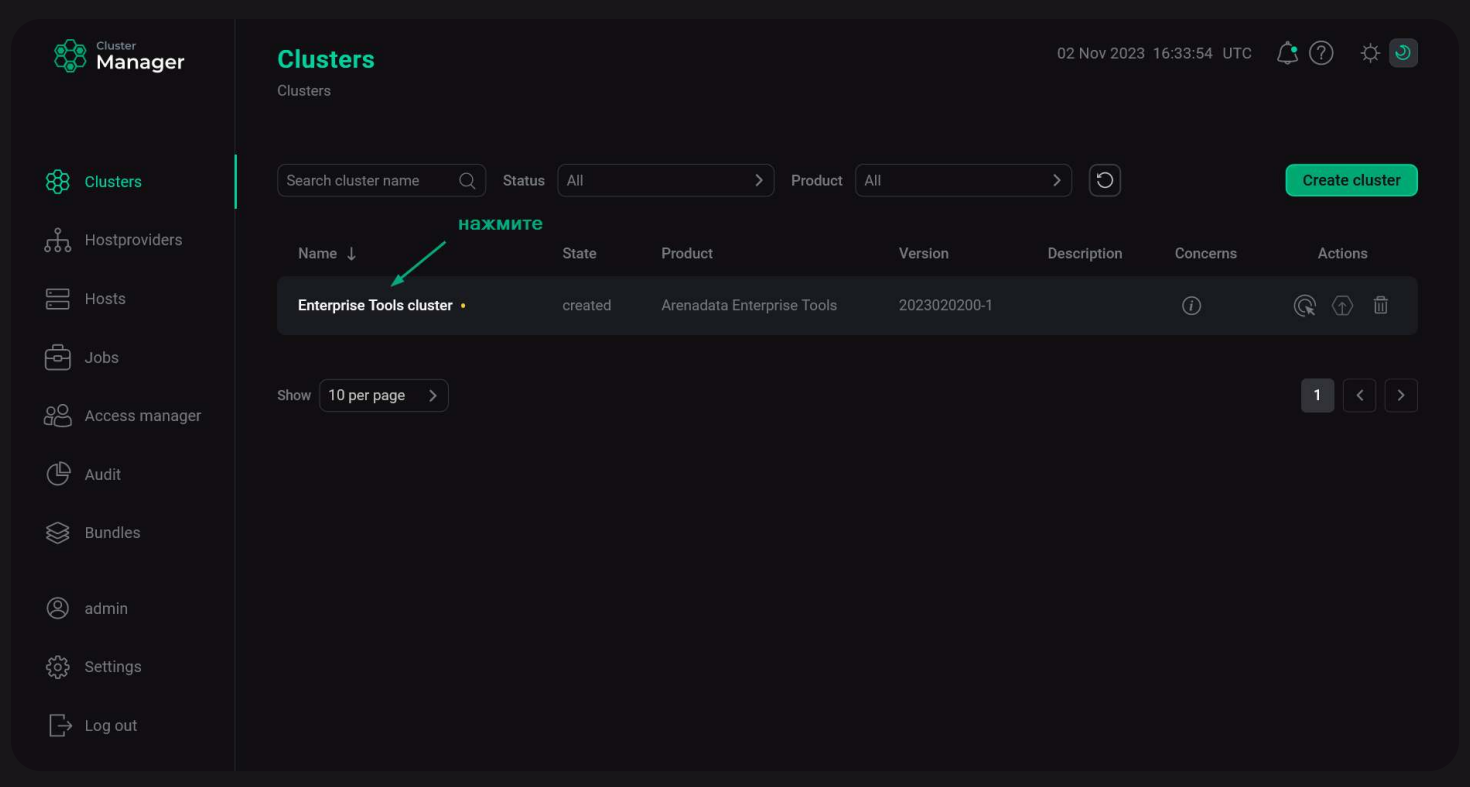
Добавление хостов в кластер

Дарья Барышева

Назад к содержанию

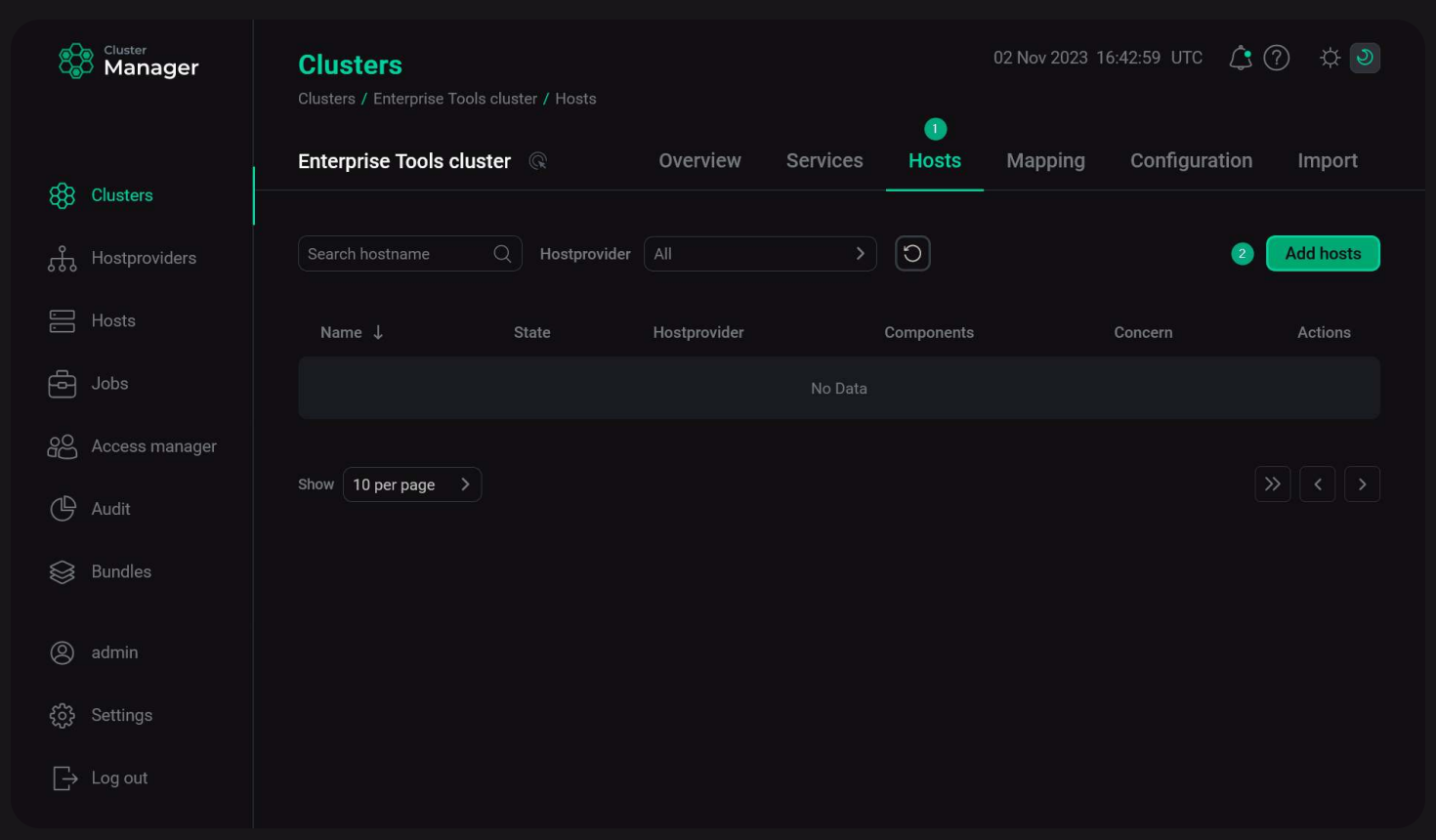
Для добавления хостов в кластер:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



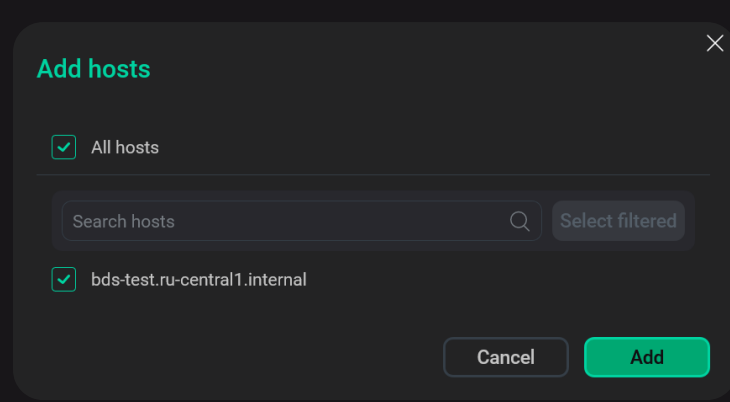
Выбор кластера

2. Откройте вкладку ***Hosts*** на странице кластера и нажмите ***Add hosts***.



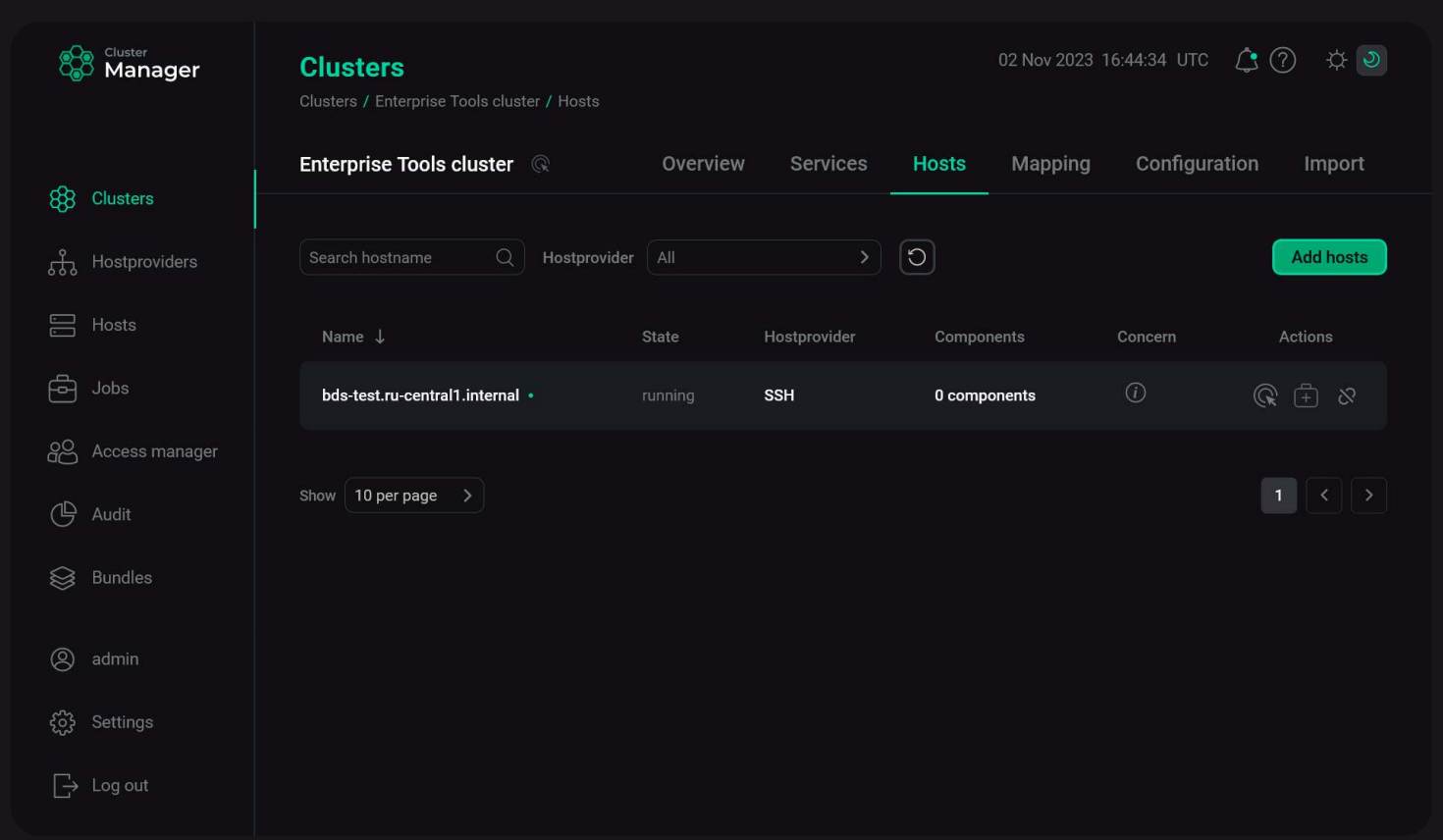
Переход к добавлению хостов

3. В открывшемся окне выберите хосты для добавления в кластер и нажмите ***Add***.



Выбор хостов

4. В результате успешно добавленные хосты отображаются на вкладке ***Hosts***.



Результат успешного добавления хостов в кластер

Добавление компонентов

Дарья Барышева

Назад к содержанию

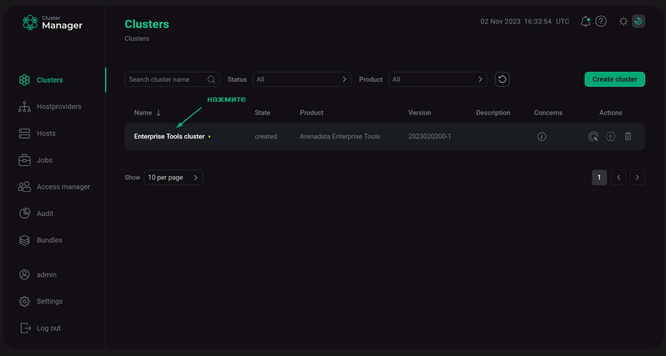
В ADCM **компонент** означает часть сервиса, которая должна быть развернута на одном или нескольких хостах кластера.

ВАЖНО

Изначально на хостах кластера нет ни одного компонента — распределение компонентов между хостами должно быть проведено вручную.

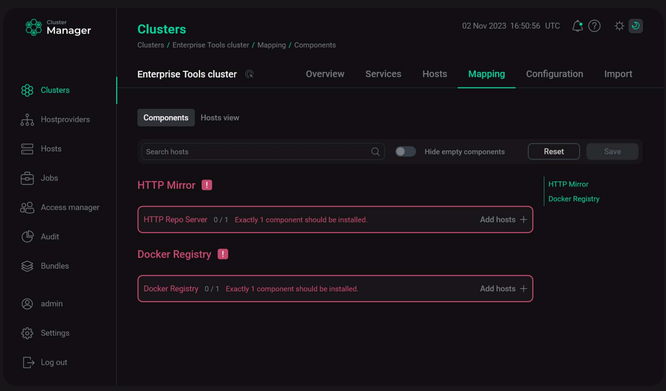
Для добавления компонентов на хосты кластера:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



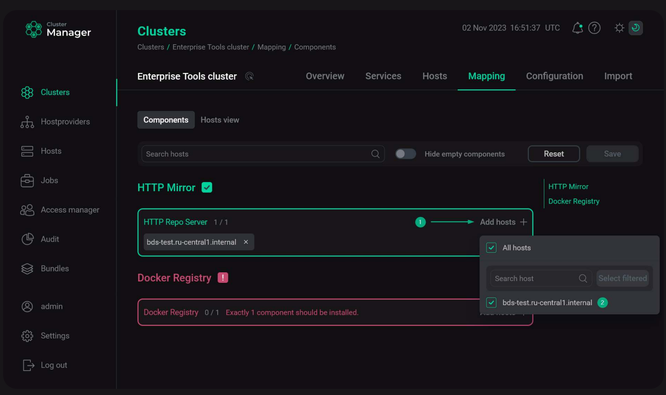
Выбор кластера

2. Откройте вкладку ***Mapping*** на странице кластера.



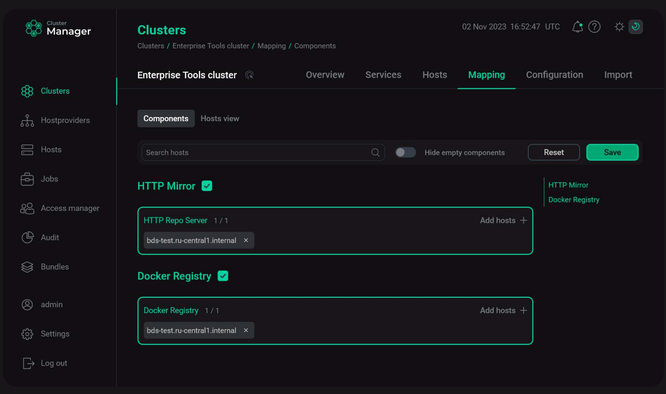
Переход к распределению сервисных компонентов

3. Распределите все компоненты между хостами кластера. Для назначения хостов компоненту нажмите ***Add hosts*** и выберите хосты в открывшемся окне.



Выбор хостов для компонента

4. После завершения распределения компонентов нажмите ***Save***.



Сохранение распределения компонентов

Установка кластера

Дарья Барышева

Содержание

Назад к содержанию

Шаг 1. Запуск установки

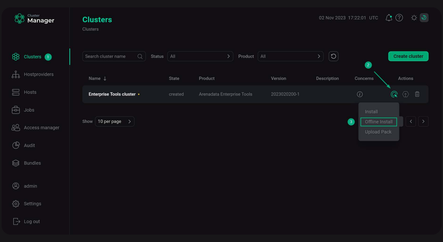
Шаг 2. Загрузка пакетов для offline-установки

Шаг 3. Проверка результатов

Шаг 1. Запуск установки

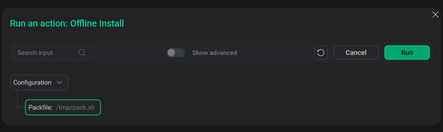
Для установки кластера Enterprise Tools выполните следующие шаги:

1. Запустите действие ***Offline Install*** для кластера, выбранного на странице ***Clusters***, нажав на иконку в столбце ***Actions***.



Переход к установке кластера

2. В открывшемся диалоговом окне введите полный путь к shell-скрипту, полученному от команды поддержки Arenadata, и нажмите ***Run***.

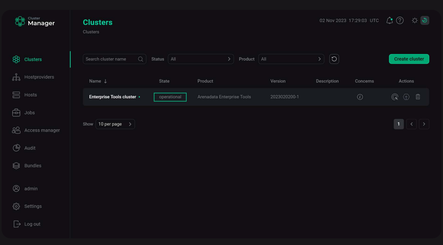


Подтверждение установки

3. Дождитесь завершения процесса установки.

ПРИМЕЧАНИЕ

В случае ошибок логи можно найти на странице ***Jobs***.

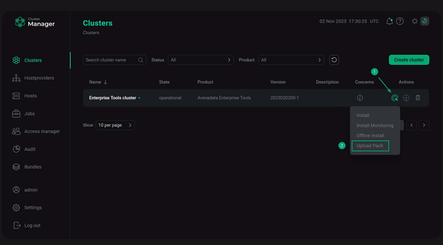


Процесс установки завершен

Шаг 2. Загрузка пакетов для offline-установки

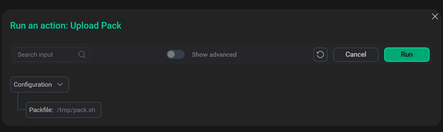
В дополнение к установке сервисов ET требуется загрузка всех установочных файлов и образов в инициализированный http-репозиторий и Docker Registry на хосте, где развернут кластер ET. Это важно для последующего offline-развертывания продуктов Arenadata. Для загрузки выполните шаги:

1. Запустите действие ***Upload Pack*** для кластера, выбранного на странице ***Clusters***, нажав на иконку в столбце ***Actions***.



Переход к загрузке пакетов

2. В открывшемся диалоговом окне введите полный путь к shell-скрипту, полученному от команды поддержки Arenadata, и нажмите ***Run***.



Подтверждение загрузки

3. Дождитесь завершения процесса загрузки.

ПРИМЕЧАНИЕ

В случае ошибок логи можно найти на странице ***Jobs***.

Шаг 3. Проверка результатов

Для проверки установки на хосте, где развернут кластер, выполните шаги:

1. Убедитесь, что порты `443` и `81` находятся в статусе `LISTEN`. Первый порт используется сервисом Docker Registry, второй порт — сервисом HTTP Mirror.

```
$ sudo netstat -ntlp|grep L|grep -E "81|443"
```

Результат команды:

```
tcp6       0      0 :::443                  :::*                    LISTEN      1694/docker-proxy-c
tcp6       0      0 :::81                   :::*                    LISTEN      1708/docker-proxy-c
```

2. Проверьте наличие директории */opt/arenadata/etools/httprepo/arenadata-repo*. Она должна содержать файлы, необходимые для offline-установки продуктов, выбранных вами на этапе получения shell-скрипта от команды поддержки Arenadata.

```
$ ls -la /opt/arenadata/etools/httprepo/arenadata-repo
```

Следующий пример показывает данные, которые могут храниться в директории в случае установки ADB.

```
total 0
drwxr-xr-x 4 root root 28 Feb 28 17:48 .
drwxr-xr-x 3 root root 28 Feb 28 17:48 ..
drwxr-xr-x 3 root root 32 Feb 28 17:48 ADB
drwxr-xr-x 3 root root 17 Feb 28 17:48 ADM
```

Создание кластера

Дарья Барышева

Содержание

Назад к содержанию

Шаг 1. Получение Enterprise-версии бандла

Шаг 2. Загрузка бандла в ADCM

Шаг 3. Создание кластера на базе загруженного бандла

В ADCM **кластер** означает набор сервисов, запущенных на группе хостов.

Дистрибутивы кластеров для ADCM поставляются в виде бандлов. В общем случае бандл представляет обычный архив, описывающий логику развертывания кластера.

Шаги для добавления нового кластера приведены ниже.

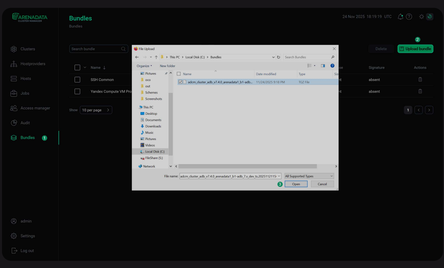
Шаг 1. Получение Enterprise-версии бандла

Пожалуйста, запросите бандл у команды поддержки Arenadata.

Шаг 2. Загрузка бандла в ADCM

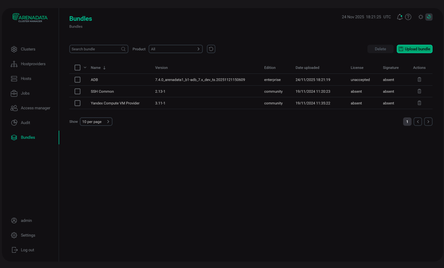
Чтобы загрузить бандл в ADCM:

1. Выберите пункт ***Bundles*** в левом навигационном меню и нажмите ***Upload bundle***.
2. Выберите бандл в открывшемся диалоговом окне.



Переход к загрузке бандла

3. После успешной загрузки бандл отображается на странице ***Bundles***.



Результат успешной загрузки бандла

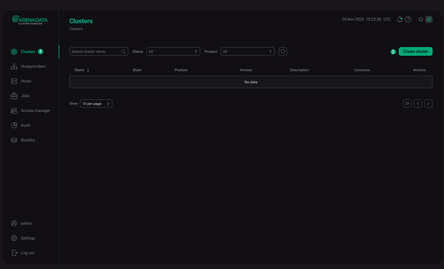
Шаг 3. Создание кластера на базе загруженного бандла

ВАЖНО

На этапе создания кластера через web-интерфейс ADCM генерируется только сам кластер. Это означает добавление записи о кластере в ADCM — сервисы на этом шаге еще не устанавливаются.

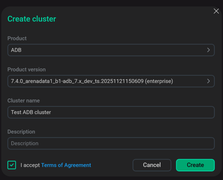
Для создания кластера необходимо выполнить шаги:

1. Выберите пункт ***Clusters*** в левом навигационном меню и нажмите ***Create cluster***.



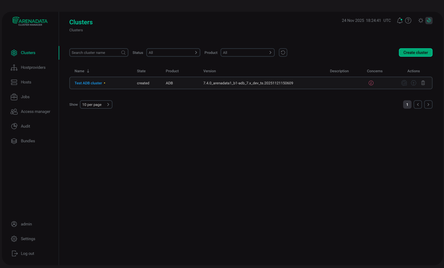
Переход к созданию кластера

2. В открывшемся окне требуется описать новый кластер:
   - Выберите загруженный бандл в поле ***Product***.
   - Выберите версию бандла в поле ***Product version***. Несколько версий становятся доступны после загрузки нескольких бандлов для одного типа продуктов.
   - Введите имя кластера в поле ***Cluster name***.
   - При необходимости добавьте описание кластера в поле ***Description***.
   - Установите флаг ***I accept Terms of Agreement*** для подписания лицензионного соглашения. Флаг отображается только для бандлов версии Enterprise. Чтобы ознакомиться с текстом соглашения, перейдите по ссылке ***Terms of Agreement***. Подписать соглашение также возможно на странице ***Bundles***, нажав на соответствующий продуктовый бандл.
   - Нажмите ***Create***.



Создание кластера

3. После успешного добавления кластер отображается на странице ***Clusters***.



Результат успешного добавления кластера

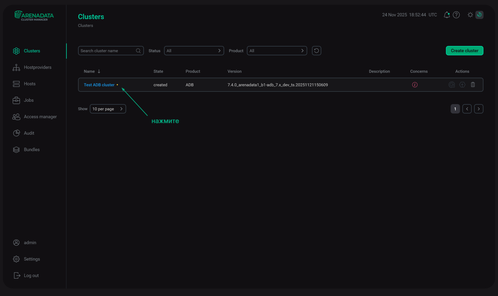
Добавление сервисов

Дарья Барышева

Назад к содержанию

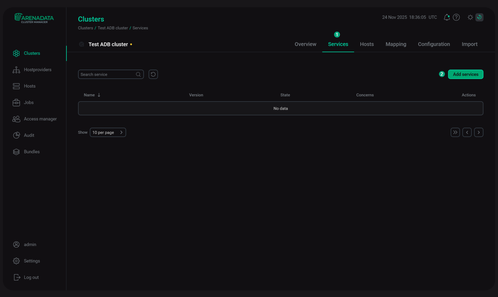
В ADCM **сервис** означает программное обеспечение, выполняющее некоторую функцию. Примеры сервисов в кластерах ADB: ADB, ADB Control, Chrony и так далее. Для добавления сервисов в кластер:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



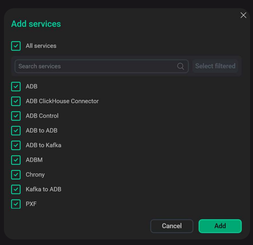
Выбор кластера

2. Откройте вкладку ***Services*** на странице кластера и нажмите ***Add service***.



Переход к добавлению сервисов

3. В открывшемся окне выберите сервисы для добавления в кластер и нажмите ***Add***.



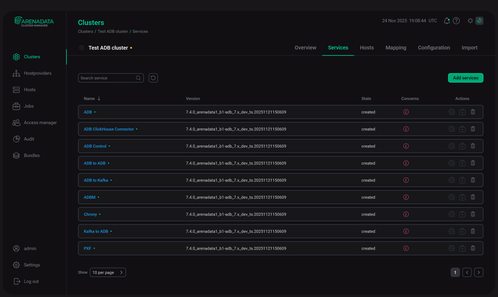
Выбор сервисов

Краткое описание доступных сервисов приведено ниже. `CE` означает Common-версию ADB (Common Edition), `EE` — Enterprise-версию ADB (Enterprise Edition).

Сервисы, которые могут быть добавлены в кластер ADB

Сервис	Обязательность	Версия ADB	Назначение
ADB	Да	CE/EE	Основной сервис, соответствующий MPP-системе на базе Greengage DB
Chrony	Нет	CE/EE	Сервис времени. Обеспечивает синхронизацию часов на хостах кластера, что необходимо для его корректной работы. Установка Chrony не требуется, если на хостах уже используется аналогичный сервис
PXF	Нет	CE/EE	Сервис, обеспечивающий взаимодействие ADB с внешними источниками данных
ADB ClickHouse Connector	Нет	EE	ADB ClickHouse Connector — сервис, обеспечивающий двустороннее взаимодействие ADB с ClickHouse. Требует установки PXF
ADB to Kafka	Нет	EE	ADB to Kafka Connector — сервис, обеспечивающий одностороннее взаимодействие ADB с Kafka. Требует установки PXF
Kafka to ADB	Нет	EE	Kafka to ADB Connector — сервис, обеспечивающий одностороннее взаимодействие Kafka с ADB
ADB to ADB	Нет	EE	ADB to ADB Connector — сервис, обеспечивающий двустороннее взаимодействие между кластерами ADB
ADBM	Нет	EE	Arenadata DB Backup Manager — сервис для управления бинарными бэкапами ADB. Может быть установлен только offline
ADB Control	Нет	EE	Arenadata DB Control (ADB Control) — сервис для мониторинга, диагностики и управления запросами, запущенными в кластере ADB. Может быть установлен только offline

4. В результате успешно добавленные сервисы отображаются на вкладке ***Services***.



Результат успешного добавления сервисов в кластер

ПРИМЕЧАНИЕ

Опциональные сервисы могут быть добавлены в кластер позднее. Процесс добавления сервисов в уже развернутый кластер не отличается от описанного выше.

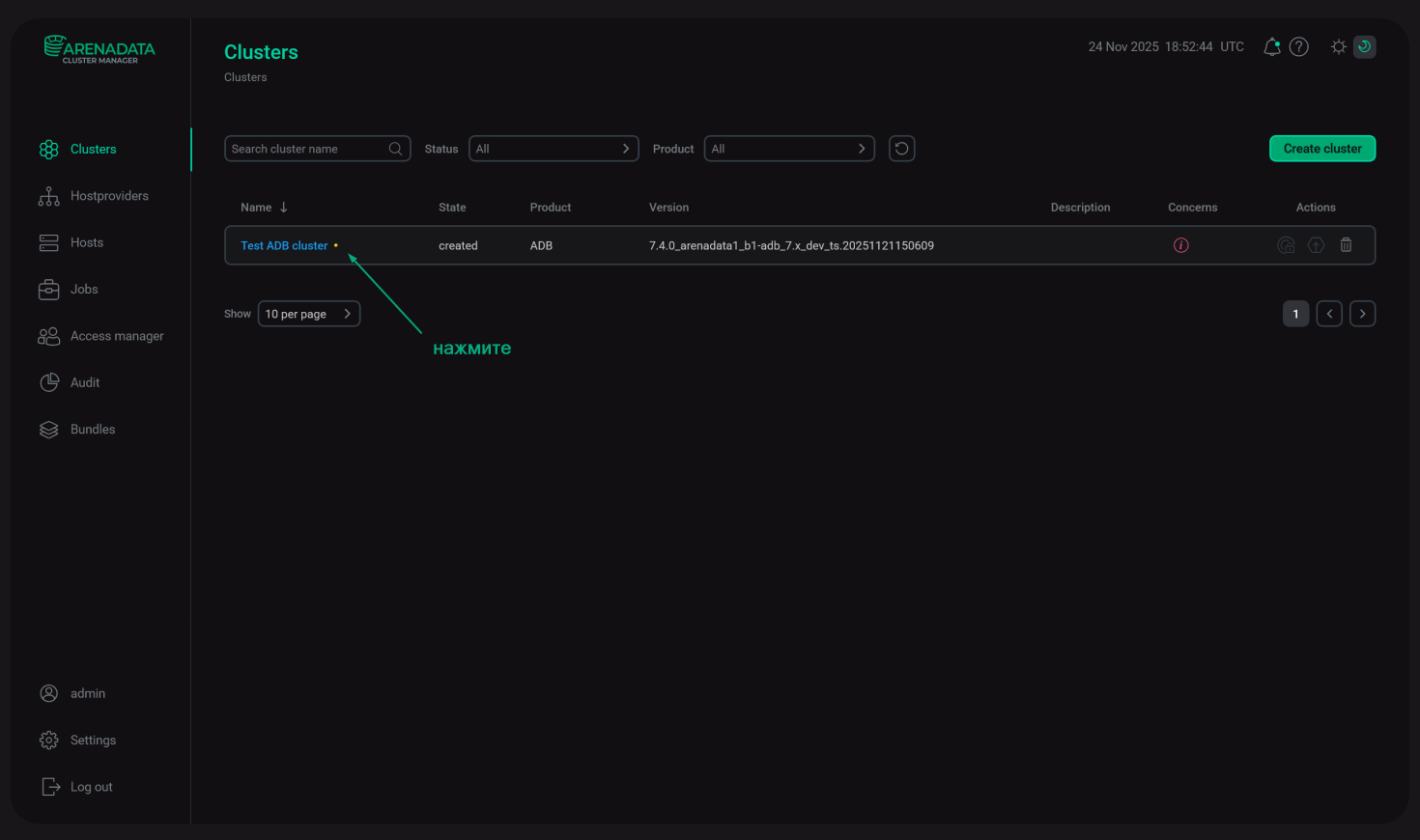
Добавление хостов в кластер

Дарья Барышева

Назад к содержанию

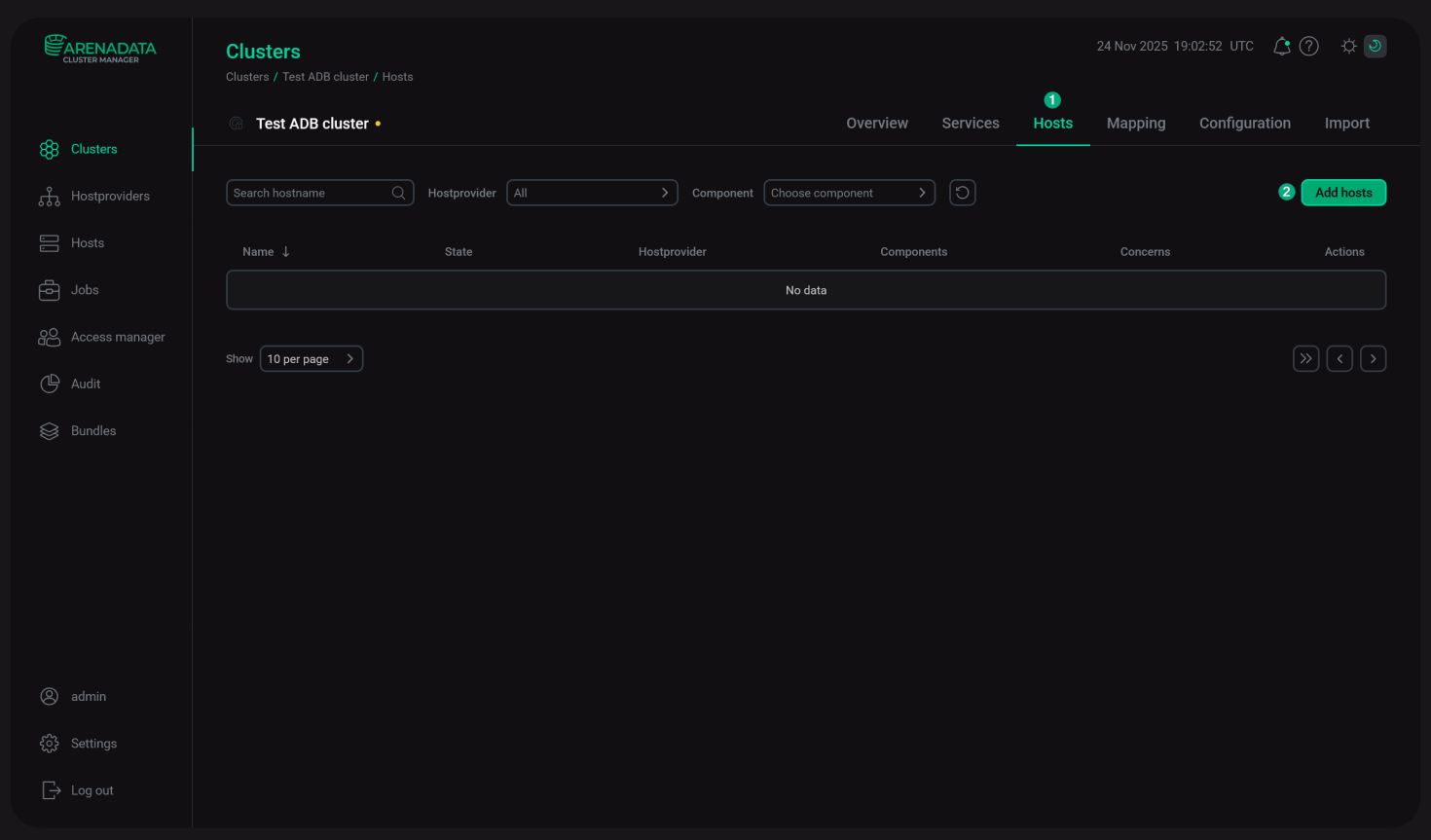
Для добавления хостов в кластер:

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



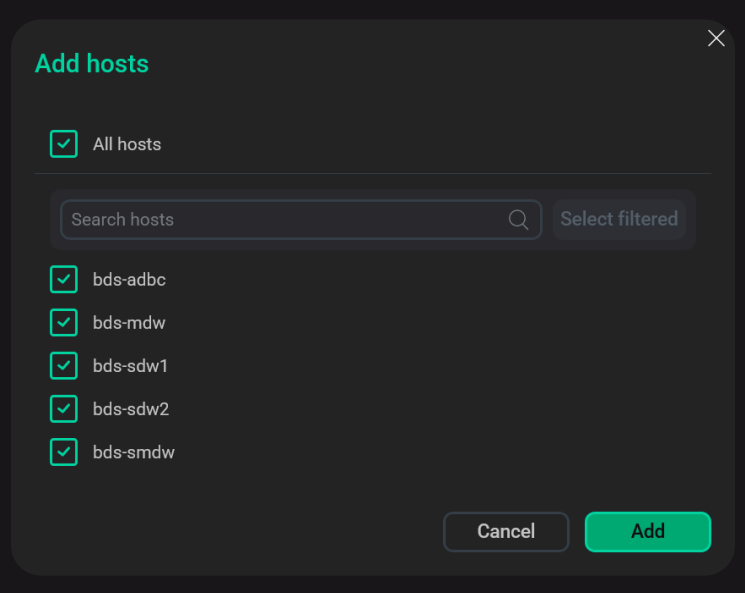
Выбор кластера

2. Откройте вкладку ***Hosts*** на странице кластера и нажмите ***Add hosts***.



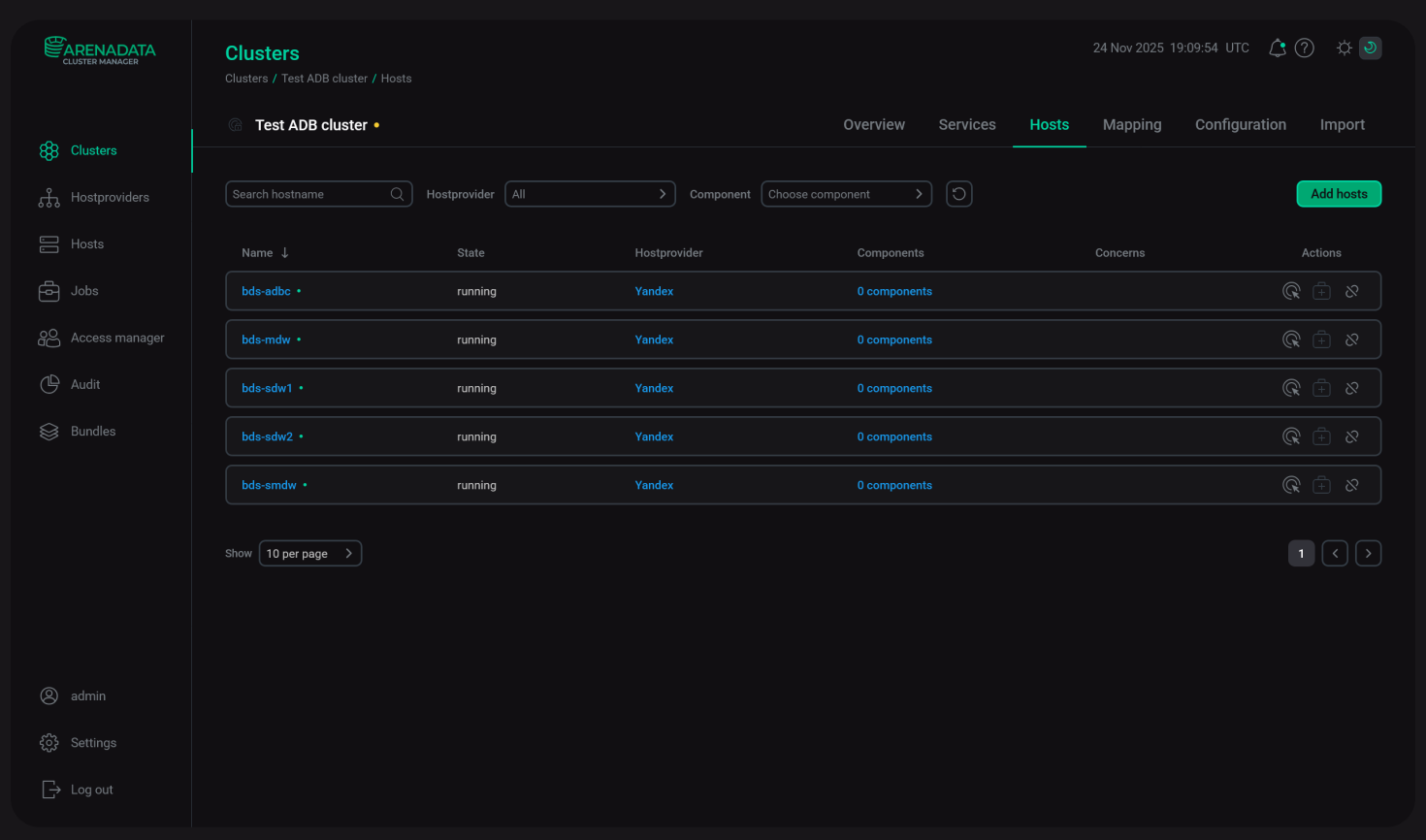
Переход к добавлению хостов

3. В открывшемся окне выберите хосты для добавления в кластер и нажмите ***Add***.



Выбор хостов

4. В результате успешно добавленные хосты отображаются на вкладке ***Hosts***.



Результат успешного добавления хостов в кластер

Добавление компонентов

Д. Дарья Бирюшева

Назад к содержанию

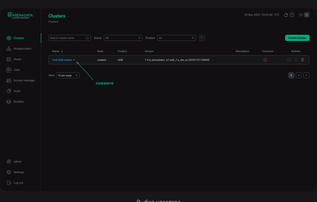
В ADCM **компонент** означает часть сервиса, которая должна быть развернута на одном или нескольких хостах кластера. Например, сервис ADB состоит из таких компонентов, как ADB Master, ADB Segment и ADB Standby.

ВАЖНО

Изначально на хостах кластера нет ни одного компонента – распределение компонентов между хостами должно быть проведено вручную.

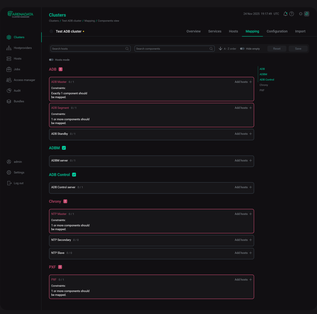
Для добавления компонентов на хосты кластера:

1. Выберите кластер на странице *Clusters*. Для этого нажмите на имя кластера в столбце *Name*.



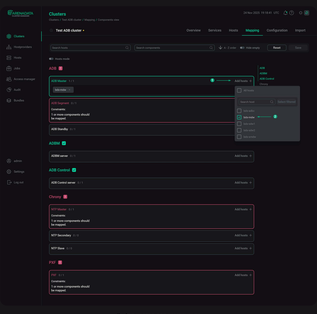
Выбор кластера

2. Откройте вкладку *Mapping* на странице кластера.



Переход к распределению сервисных компонентов

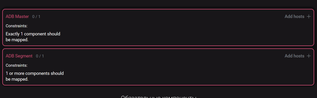
3. Распределите все компоненты между хостами кластера. Для назначения хостов компоненту нажмите *Add hosts* и выберите хосты в открывшемся окне.



Выбор хостов для компонента

Обязательные компоненты

Каждый сервис может иметь обязательные и опциональные компоненты. Обязательные подсвечиваются красным цветом. Также в интерфейсе ADCM для них выводится информация о том, в каком количестве необходимо установить каждый компонент.



Обязательные компоненты

ВНИМАНИЕ

Без назначения нужного числа хостов обязательным компонентам сохранение общей схемы распределения невозможно.

Типовая схема распределения компонентов в кластере ADB приведена в таблице ниже. Она основана на Enterprise-версии ADB.

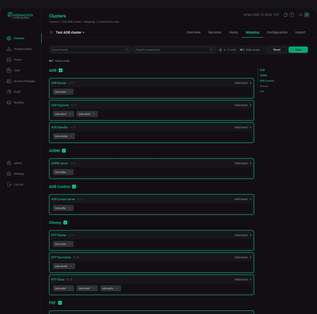
Распределение компонентов ADB

Хост	Сервис	Компонент	Примечание
Хост координатора	ADB	ADB Master	Необходимо установить на координатор-хосте
	Chrony	NTP Master	Необходимо установить на хосте координатор
Хост резервного координатора (standby)	ADB	ADB Standby	Необходимо установить на хосте резервного координатора (standby)
	Chrony	NTP Secondary	Необходимо установить на хосте резервного координатора (standby)
Сегмент-хост (1..N)	ADB	ADB Segment	Необходимо установить на каждом сегмент-хосте
	Chrony	NTP Slave	Необходимо установить на каждом сегмент-хосте
	PXF	PXF	Необходимо установить на сегмент-хостах и — опционально — на координатор-хосте
ADB Control/ADBM host	ADB Control	ADB Control server	Рекомендуется использовать отдельный хост для этого компонента
	ADBM	ADBM server	Можно установить вместе с ADB Control server на одном хосте
	Chrony	NTP Slave	Необходимо установить на хосте ADB Control/ADBM

ВНИМАНИЕ

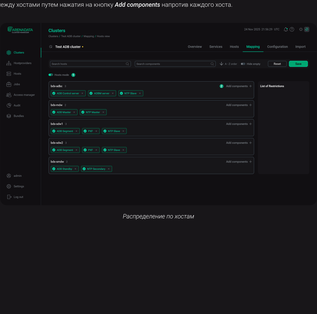
Не следует устанавливать компоненты ADB Control server и ADBM server на одном хосте с ADB Master или ADB Segment. Эта возможность оставлена исключительно для ознакомительных целей. Разместите указанные компоненты на отдельном хосте.

4. После завершения распределения компонентов нажмите *Save*.



Сохранение распределения компонентов

Начиная с версии 2.2.0 ADCM поддерживает альтернативный способ сопоставления — по хостам. Для этого переведите в активное положение переключатель ***Hosts mode*** и распределите компоненты между хостами путем нажатия на кнопку ***Add components*** напротив каждого хоста.



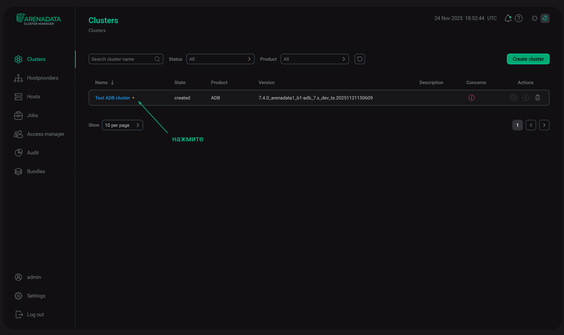
Распределение по хостам

Настройка сервисов

Дарья Барышева

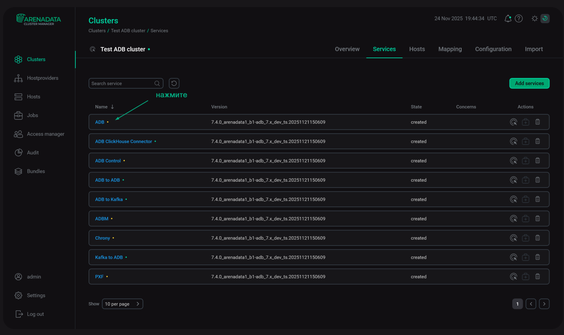
Назад к содержанию

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



Выбор кластера

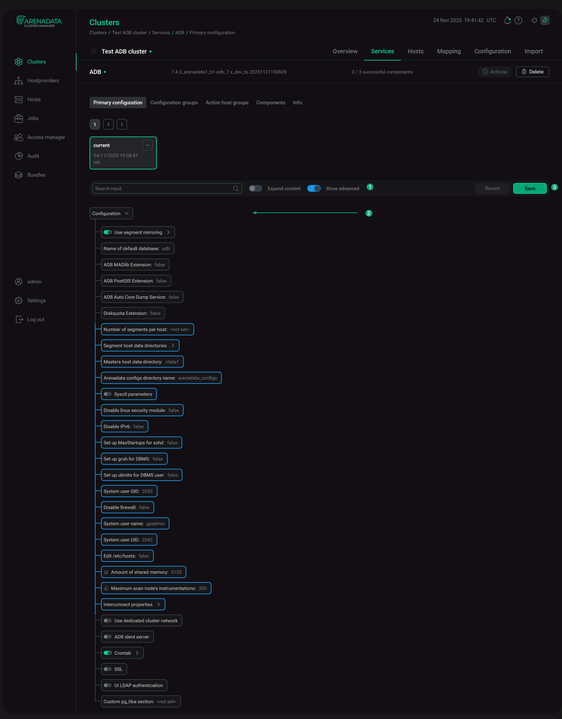
2. Откройте вкладку ***Services*** на странице кластера. Нажмите на имя сервиса в столбце ***Name***.



Переход к настройке сервиса

3. В открывшемся окне:

   - Откройте вкладку ***Primary Configuration***.
   - Переведите в активное состояние переключатель ***Show advanced***.
   - Заполните конфигурационные параметры выбранного сервиса. Поля, подсвеченные красным, обязательны для заполнения.
   - Нажмите ***Save***.



Настройка сервиса

Обратите внимание на следующие параметры:

- ***ADB → Edit /etc/hosts***. Установите значение `true`, чтобы сервис ADB мог добавить имена хостов кластера в файл */etc/hosts*. Если значение — `false`, требуется вручную внести изменения в этот файл или настроить DNS-сервер. Через настройки сервиса ADB вы можете изменить и другие параметры хостов, чтобы привести их к рекомендуемым значениям.
- ***ADB → Disable linux security module*** и ***ADB → Set up GRUB for DBMS***. Изменение этих параметров требует перезапуска хостов, поэтому рекомендуется установить для них значение `true` до установки кластера, а затем установить значение `true` для параметра ***Reboot cluster servers after installation*** в кластерном действии Install, чтобы хосты перезапустились автоматически. Если менять их значения после установки, потребуется остановить кластер и перезапустить хосты вручную.
- ***Chrony → NTP servers***. Хотя этот параметр необязателен, мы рекомендуем установить в нем адрес ближайшего NTP-сервера (например, используемого в вашей компании).

ПРИМЕЧАНИЕ

- Более подробную информацию по всем конфигурационным параметрам можно получить в статье Конфигурационные параметры.
- В случае использования внешних БД для ADB Control и/или ADBM требуются дополнительные действия по настройке этих сервисов перед установкой:
  - ADB Control
  - ADBM

Содержание

Перед содержанием

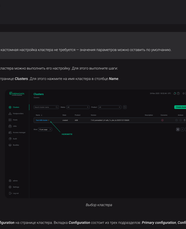
Primary configuration
Configuration groups
Ansible settings

ВАЖНО

В большинстве случаев настроек кластера по умолчанию — значения параметров можно оставить по умолчанию.

После добавления нового кластерного менеджера выполните его настройку. Для этого выполните шаги:

1. Выберите кластер на странице **Clusters**. Для этого нажмите на имя кластера в списке **Name**.



Выбор кластера

2. Откройте вкладку **Configuration** на странице кластера. Вкладка **Configuration** состоит из трех подразделов: **Primary configuration**, **Configuration groups**, **Ansible settings**.
3. Заполните необходимые конфигурационные параметры на одной вкладке и нажмите **Save**.

Primary configuration

Вкладка **Configuration** → **Primary configuration** содержит основные конфигурационные параметры кластера.

ВНИМАНИЕ!

Используйте кастомные инстанции *Use existing docker*, *Use custom repmgr*, *Use custom JRA_HOME for cluster* с осторожностью.

Доступные для редактирования параметры:

- *Cluster UUID* — уникальный идентификатор кластера ADB, формируемый автоматически при его установке.
- *Timeout in which package manager* — параметр этой опции предназначен для настройки тайм-аута ожидания разблокировки пакетного менеджера (RF: 8 Ubuntu после перезагрузки в фоновом обновлении, по умолчанию 2 Ожидание (RF

Параметр	Описание	По умолчанию
retries	Максимальное число повторных попыток	3
delay	Тайм-аут между повторными попытками (в секундах)	10

- Пулы репозиториев, которые будут использованы в процессе установки в зависимости от операционной системы хостов (RE.Linux, Astra Linux, РЕД ОС, Ubuntu), на основе которых регистрируются только стандартные репозитории

Altlinux 10 repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

Astra linux repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

redos 7 repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

Ubuntu 22.04 repositories

Параметр	Описание
Set up Annexdata repo	Необходимо использовать, регистрируй, указанный в поле **Annexdata repo url**
Annexdata repo url	Репозиторий, необходимый для установки ADB
Set up custom Annexdata enterprise repo	Необходимо использовать, регистрируй, указанный в поле **Custom Annexdata enterprise repo**
Custom Annexdata enterprise repo	Кастомный репозиторий Annexdata Enterprise. Если не заполнен — используется репозиторий по умолчанию
Set up ADB monitoring repo	Необходимо использовать, регистрируй, указанный в поле **prometheus_repo** (для будущего использования)
prometheus_repo	Репозиторий, необходимый для установки мониторинга (для будущего использования)

- *Use existing docker* — флаг, указывающий на необходимость использования существующей инстанции Docker в процессе установки пакетов ADB (репрезиз, ADB Control, ADBM). Отображается, если установлен флаг **Use advanced** в верхней части формы. Если выражение перечислено в списке для исключения поля **JRA_HOME**, то все требуемые ресурсы будут использованы. Для выражения по умолчанию (если выражение не перечислено) в бинарном файлом (exe, exe.config) скопируются новый файл. При отсутствии файла по умолчанию или отсутствии совместимости с ошибкой. Дополнительная информация о требованиях к для установки Docker см. в разделе «Требования к Docker» настоящего руководства.
  При пустом значении поля **JRA_HOME** для поиска Java используется по умолчанию один из стандартных путей в зависимости от операционной системы.

Ubuntu
/usr/lib/jvm/java-17-openjdk-amd64/
РЕД ОС
/usr/lib/jvm/jre-17-openjdk/
Astra linux
/usr/lib/jvm/jre-17/

ВАЖНО!

- Если вы планируете использовать собственную Java (например, от Azul), для установки требуется Java версии 17.
- В Astra Linux запоминание поля **JRA_HOME** обнаружено начиная с версии 1.0.1.1.1

Импорт настроек ET

Дарья Барышева

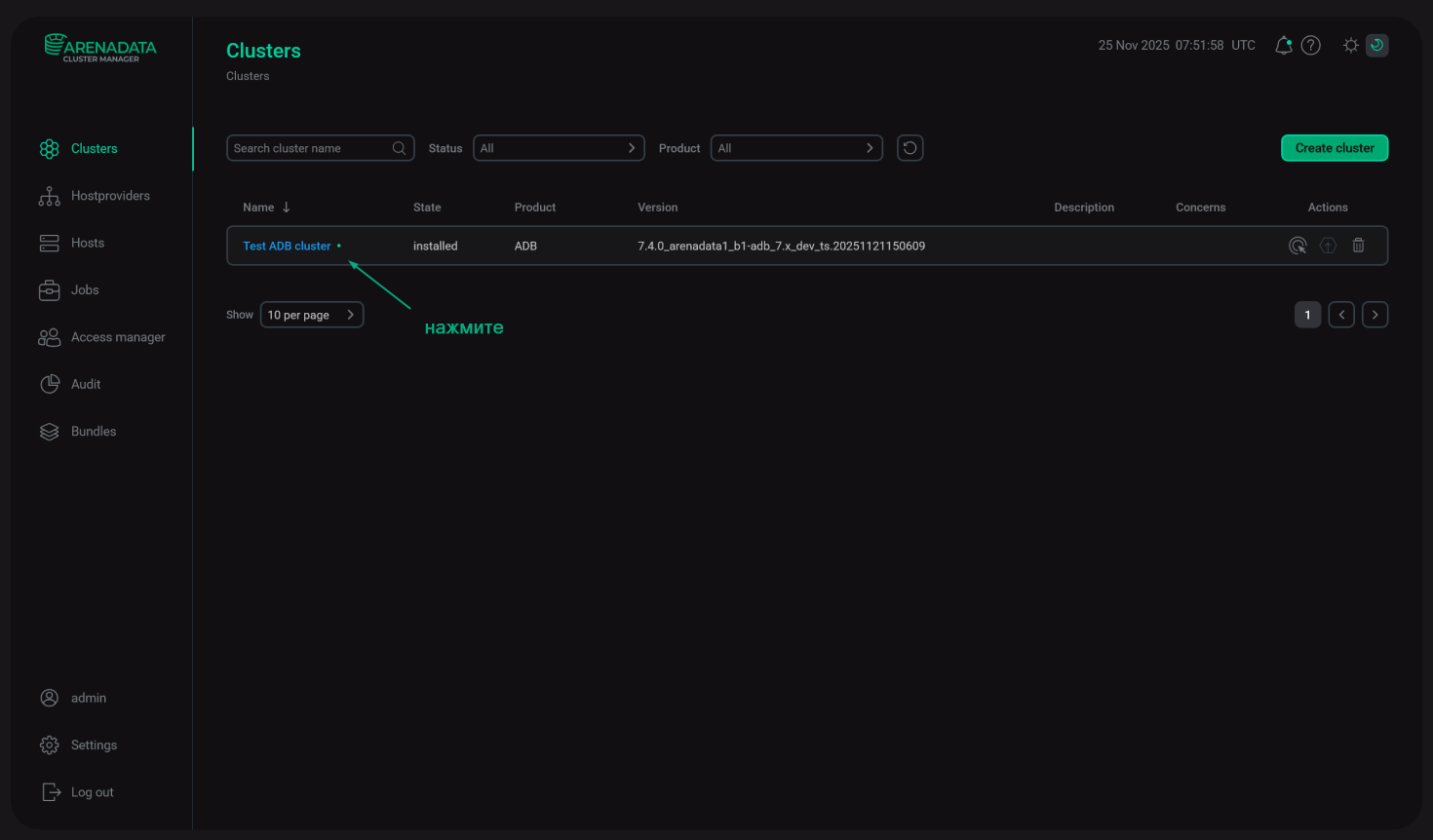
Назад к содержанию

ВАЖНО

Этот шаг необходим только в случае offline-установки.

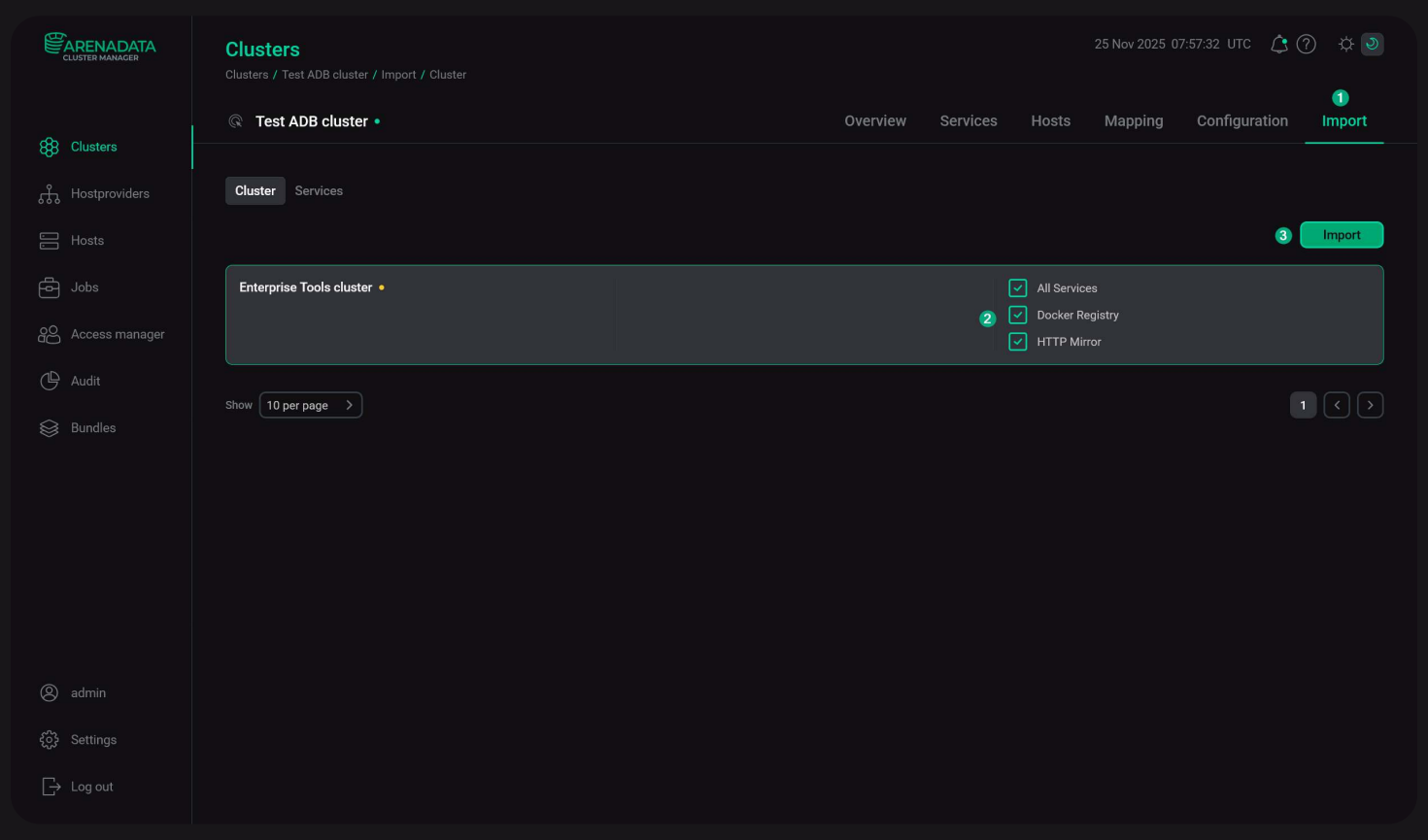
Для установки кластера в режиме offline требуется интегрировать его с ранее созданным кластером Enterprise Tools (ET):

1. Выберите кластер на странице ***Clusters***. Для этого нажмите на имя кластера в столбце ***Name***.



Выбор кластера

2. Откройте вкладку ***Import*** на странице кластера.
3. Выберите сервисы ***Docker Registry*** и ***HTTP Mirror*** предварительно созданного кластера ET.
4. Нажмите ***Import***.



Интеграция с кластером ET

ПРИМЕЧАНИЕ

После успешного импорта данные, хранимые локально в кластере Enterprise Tools, могут быть использованы на этапе установки продуктового кластера. Таким образом, их загрузка через Интернет не требуется.

Установка кластера

Содержание

Назад к содержанию

Шаг 1. Предварительная проверка (precheck)

Шаг 2. Запуск установки

- Установка всего кластера
- Установка отдельных сервисов

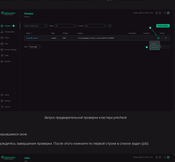
Шаг 3. Проверка статуса кластера после установки

Шаг 4. Проверка создания базы данных по умолчанию

Шаг 1. Предварительная проверка (precheck)

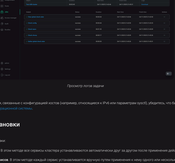
Перед установкой нового кластера рекомендуется выполнить его предварительную проверку – **Precheck**. Проверка покажет убедится в том, что все необходимые результаты достигнуты, а которые не могут к данному моменту быть выполнены и какие пункты требуют наблюдения. Для запуска проверки:

1. Проверьте действие **Precheck** в кластере, выбранном на странице **Clusters**, нажав на иконку в столбце **Actions**.



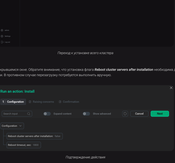
Запуск предварительной проверки кластера precheck

2. Подтвердите действие в открывшемся окне.
3. Откройте страницу **Jobs** и дождитесь завершения проверки. После этого измените по первой строке в списке задач (job).



Переход к списку задач (job)

4. Просмотрите логи в открывшемся окне. Для загрузки логов нажмите **Download**.



Просмотр логов задачи

Если в логах содержится ошибка, связанная с конфигурацией хостов (например, относящаяся к IPv6 или параметрам sysctl), убедитесь, что была выполнена предварительная подготовка хостов.

Шаг 2. Запуск установки

Существует два способа установки:

- **Установка всего кластера**. В этом методе кластер создаётся и устанавливается автоматически до другого после применения действия **Install** к кластеру.
- **Установка отдельных сервисов**. В этом методе каждый сервис устанавливается вручную путём применения к нему одного или нескольких действий. Этот способ также можно использовать для добавления новых сервисов в уже развёрнутый кластер.

В обоих методах каждый сервис запускается автоматически после установки.

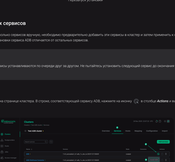
ПРИМЕЧАНИЕ

В случае ошибок логи можно найти на странице **Jobs**.

Установка всего кластера

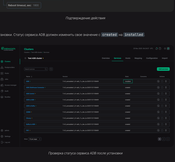
Чтобы установить или развернуть кластер на одном операциям, выполните цепи:

1. Проверьте действие **Install** в кластеру, выбранному на странице **Clusters**, нажав на иконку в столбце **Actions**.



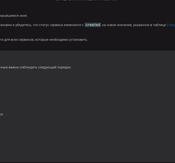
Переход к установке всего кластера

2. Подтвердите действие в открывшемся окне. Обратите внимание, что установка флага **Reboot cluster servers after installation** необходима для перезагрузки системы после завершения операции. Если отсутствует перезагрузка необходима вручную.



Подтверждение действия

3. Дождитесь завершения установки.
4. Если один или несколько серверов не удается установить, можно определить причину ошибки по логам, исправить их и перезапустить установку. Для этого необходимо выбрать действие **Install** в столбце **Actions** на странице **Clusters**.



Перезапуск установки

Установка отдельных сервисов

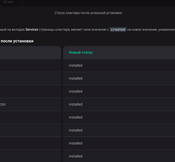
Чтобы установить один или несколько сервисов вручную, необходимо последовательно добавить эти сервисы в кластер и затем применить к ним действие, описанные в подразделах ниже. Пропуск установки каждого АОВ отличается от отдельных сервисов.

ВНИМАНИЕ!

Обратите внимание, что первичы устанавливается по порядку друг за другом. Не пытайтесь установить следующий сервис до окончания установки предыдущего.

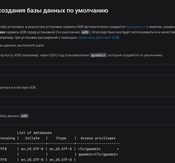
Сервис АОВ

1. Откройте вкладку **Services** на странице кластера. В случае, соответствующий сервису АОВ, нажмите на иконку в столбце **Actions** и выберите действие **Install AOB** из выпадающего списка.



Переход к установке АОВ

2. Подтвердите действие в открывшемся окне.



Подтверждение действия

3. Дождитесь завершения установки. Статус сервиса АОВ должен измениться значение с `installed` на `installed`.

Проверка статуса сервиса АОВ после установки

Другие сервисы

1. Откройте вкладку **Services** на странице кластера. Для сервиса, который должен быть установлен, нажмите на иконку в столбце **Actions** и выберите действие **Install** из выпадающего списка.

Переход к установке отдельных сервисов

2. Подтвердите действие в открывшемся окне.
3. Дождитесь завершения установки и убедитесь, что статус сервиса изменился с `installed` на новое значение, указанное в таблице Статусы сервисов АОВ после установки.
4. Повторите предыдущие шаги для всех сервисов, которые необходимо установить.

ВАЖНО!

При установке сервисов вручную важно соблюдать следующий порядок:

1. AOB
2. Chrony
3. AOB to AOB
4. PKI
5. AOB Clickhouse Connector
6. AOB to Kafka
7. Kafka to AOB
8. AOB Control
9. AOBM

Шаг 3. Проверка статуса кластера после установки

В результате успешной установки произойдут обновления статуса кластера и его сервисов:

- Статус кластера, отображаемый в столбце **State** на странице **Clusters**, изменит свое значение с `installed` на `installed`.

Статус кластера после успешной установки

- Статус сервисов, отображаемый на вкладке **Services** страницы кластера, изменит свое значение с `installed` на новое значение, указанное в таблице ниже:

Статусы сервисов АОВ после установки

Сервис	Новый статус
AOB	installed
Chrony	installed
PKI	installed
AOB Clickhouse Connector	installed
AOB to AOB	installed
AOB to Kafka	installed
Kafka to AOB	installed
AOB Control	installed
AOBM	installed

Статус сервисов кластера после успешной установки

Шаг 4. Проверка создания базы данных по умолчанию

Независимо от выбранного способа установки в результате успешного развертывания сервиса АОВ автоматически создается base database в кластере, указанном в configuration parameter **Name of default database** сервиса АОВ перед установкой (по умолчанию **aob**). Этот кластер затем будет использоваться в качестве базы данных по умолчанию для клиентских подключений (например, при установке расширений) с помощью psql(1) или psql(1) АОВ.

Чтобы убедиться в создании базы данных, выполните шаги:

1. Подключитесь к владелецу-кластер АОВ (например, через SSH) под пользователем `gpadmin`, который создается по умолчанию.

```
$ sudo su - gpadmin
```

2. Получите список всех баз данных в кластере АОВ.

```
psql -l
```

Вывод команды содержит базу данных `aob`:

```
                              List of databases
   Name    | Owner | Encoding |  Collate   |   Ctype    | Access privileges
-----------+-------+----------+------------+------------+-------------------
 aob       | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =Tc/gpadmin       +
           |       |          |            |            | gpadmin=CTc/gpadmin
 postgres  | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =Tc/gpadmin       +
 template0 | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/gpadmin        +
           |       |          |            |            | gpadmin=CTc/gpadmin
 template1 | gpadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =Tc/gpadmin       +
           |       |          |            |            | gpadmin=CTc/gpadmin
(4 rows)
```

Аутентификация LDAP в ADB

Дарья Барышева

Содержание

Назад к содержанию

Настройка LDAP-аутентификации

Синхронизация ADB и LDAP

Проверка LDAP-аутентификации

В ADB, как и в Greenplum, поддерживается аутентификация пользователей с использованием протокола Lightweight Directory Access Protocol (LDAP). При использовании LDAP-аутентификации проверка учетных записей пользователей и их паролей выполняется на выбранном LDAP-сервере. Порядок настройки LDAP-аутентификации в ADB с помощью ADCM описан ниже.

Настройка LDAP-аутентификации

Для настройки LDAP-аутентификации в кластере ADB выполните следующие шаги:

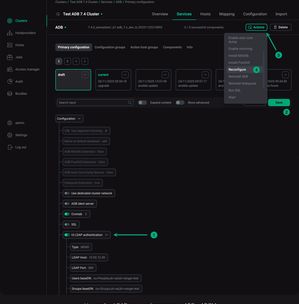
1. Убедитесь, что LDAP-сервер настроен и доступен. ADB поддерживает две реализации протокола LDAP: Microsoft Active Directory (MS AD) и 389 Directory server в составе FreeIPA.
2. Откройте конфигурационную страницу сервиса ADB в ADCM.
3. Переведите переключатель **Ldap authentication** в активное состояние.
4. Заполните следующие поля.

Поле	Описание	По умолчанию	Пример в MS AD
Type	Тип LDAP-сервера. Возможные значения: ▪ `MSAD` — Microsoft Active Directory; ▪ `FreeIPA` — 389 Directory server в составе FreeIPA.	MSAD	MSAD
LDAP Host	Доменное имя или IPv4-адрес LDAP-сервера	—	10.92.12.49
LDAP Port	Номер порта для подключения к серверу LDAP	—	389
Users baseDN	Ограничение на область поиска объектов в каталоге LDAP; применяемое в запросах на поиск пользователей	—	ou=People,dc=ad,dc=ranger-test
Groups baseDN	Ограничение на область поиска объектов в каталоге LDAP; применяемое в запросах на поиск групп пользователей	—	ou=Groups,dc=ad,dc=ranger-test
Users filter	LDAP-фильтр, используемый для поиска пользователей. Заполняется в соответствии с RFC 2254	(&(objectClass=person)(objectClass=inetOrgPerson)(cn=*))	(&(objectClass=person)(objectClass=organizationalPerson)(userName=*)(sn=*)(sAMAccountName=*))
Groups filter	LDAP-фильтр, используемый для поиска групп пользователей. Заполняется в соответствии с RFC 2254	(&(objectClass=group)(objectClass=groupofnames)(objectClass=groupOfUniqueNames))(cn=*))	(cn=*)
SA Login	Имя пользователя, используемое в сервисных запросах к LDAP-серверу	—	cn=admin,dc=ad,dc=ranger-test
Password	Пароль пользователя, используемый в сервисных запросах к LDAP-серверу	—	Пароль пользователя `admin`
Lowercase login	Необходимо ли преобразование имен пользователей в нижний регистр	false	false
Lowercase group	Необходимо ли преобразование имен групп в нижний регистр	false	false
Sync interval	`cron` выражение, описывающее периодичность синхронизации пользователей и групп в ADB на основе данных LDAP-сервера. Для синхронизации используется утилита pg-ldap-sync, поставляемая в бандле ADB	0 * * * *	0 * * * *

ПРИМЕЧАНИЕ

Различные реализации LDAP используют разные имена для типов и идентификаторов объектов. Формат параметров для конкретной реализации LDAP рекомендуется уточнить у администратора LDAP-сервера.

5. Нажмите **Save**. Примените действие **Reconfigure** к сервису ADB.



Настройка LDAP-аутентификации для ADB в ADCM

Синхронизация ADB и LDAP

Если LDAP-аутентификация настроена успешно, в ADB автоматически выполняются следующие действия:

- **Обновление файла pg_hba.conf**. В файле *pg_hba.conf*, расположенный в data-директории хоста координатора (в приведенном примере */data1/master/gpseg-1/*), добавляется новая запись об LDAP-аутентификации следующего вида:

```
# BEGIN LDAP
host all \"ldap_users 0.0.0.0/0 ldap ldapserver=<LDAP Host> ldapport=<LDAP Port> ldapbasedn="<Users baseDN>" ldapbinddn="<SA Login>" ldapbindpasswd="<Password>" ldapsearchattribute="<attribute>"
# END LDAP
```

  где:

  - `ldap_users` — имя роли в ADB, в которую будут добавлены все полученные со стороны LDAP пользователи.
  - `<LDAP Host>`, `<LDAP Port>`, `<Users baseDN>`, `<SA Login>`, `<Password>` — значения одноименных параметров сервиса ADB.
  - `<attribute>` — атрибут LDAP, значение которого будет использовано в качестве имени каждой роли в ADB при синхронизации пользователей. Принимает одно из следующих значений:
    - `sAMAccountName` — для MS AD.
    - `uid` — для FreeIPA.

- **Синхронизация пользователей и групп ADB с LDAP**. В базу данных ADB добавляется информация о группах и пользователях, заведенных на LDAP-сервере (и удовлетворяющих критериям отбора, указанным на этапе конфигурации). Синхронизация выполняется путем вызова утилиты pg-ldap-sync. Чтобы убедиться в успешной синхронизации, можно выполнить запрос к системной таблице `pg_roles`:

```
SELECT rolname FROM pg_roles;
```

  Как показывает вывод команды, в ADB автоматически созданы следующие роли в результате синхронизации с LDAP:

  - `ldap_users` — роль, в которую добавляются все LDAP-пользователи (путем автоматического вызова команды `CREATE ROLE … LOGIN IN ROLE ldap_users …`)
  - `ldap_groups` — роль, в которую добавляются все LDAP-группы (путем автоматического вызова команды `CREATE ROLE … NOLOGIN IN ROLE ldap_groups …`)
  - `ffedorov`, `ppetrov`, `ssemenov`, `zeptest` — конкретные пользователи, полученные со стороны LDAP и добавленные в роль `ldap_users`.
  - `ADCM`, `ADH`, `Adccadmins`, `Analysts`, `Managers` — конкретные группы пользователей, полученные со стороны LDAP и добавленные в роль `ldap_groups`. Членство пользователей в этих группах устанавливается путем автоматического вызова команды `GRANT`:

```
  rolname
------------
 gpadmin
 gpmon
 ldap_users
 ldap_groups
 ADCM
 ADH
 Adccadmins
 Analysts
 Managers
 ffedorov
 ppetrov
 ssemenov
 zeptest
(13 rows)
```

Особенности синхронизации

- В дальнейшем синхронизация пользователей будет проводиться автоматически в соответствии с расписанием, указанным в параметре Sync interval. Необходимости в ручном добавлении ролей путем вызова команд `CREATE ROLE` и `GRANT` нет.
- Для синхронизации данных с LDAP-сервером вне настроенного расписания следует применить действие **Reconfigure** к сервису ADB.
- После настройки LDAP-аутентификации подключение к БД становится возможным только для пользователей, зарегистрированных на LDAP-сервере. В случае удаления пользователя на стороне LDAP его дальнейшая аутентификация в ADB становится невозможна.
- Случай, когда пользователь зарегистрирован в ADB, но при этом отсутствует на стороне LDAP-сервера, валиден. При этом синхронизация данных с LDAP никак не влияет на этого пользователя. Однако, если пользователь с одним и тем же именем заведен и в ADB, и на LDAP-сервере — при следующей синхронизации данные пользователя в ADB перезаписываются информацией, полученной с LDAP-сервера.
- Утилита pg-ldap-sync автоматизирует только следующие DCL-команды: `CREATE ROLE <ROLE>`, `DROP ROLE <ROLE>`, `GRANT <ROLE> TO <ROLE>` и `REVOKE <ROLE> FROM <ROLE>`. Назначение прав доступа при помощи команд вида `GRANT SELECT`, `GRANT EXECUTE`, `GRANT CONNECT` и `GRANT PRIVILEGES ON <OBJECT> TO <ROLE>` при необходимости следует выполнить самостоятельно в ADB.

Проверка LDAP-аутентификации

Для того чтобы проверить возможность подключения к ADB после настройки LDAP-аутентификации, можно воспользоваться клиентом `psql`, указав в качестве аргумента `-U` имя одного из пользователей, заведенных на стороне LDAP (в приведенном примере — `ffedorov`).

```
$ psql adb -h bds-adw -U "ffedorov"
```

Если настройка LDAP-аутентификации выполнена успешно, на экране появится приглашение для ввода пароля пользователя. После ввода корректного пароля (назначенного на стороне LDAP-сервера) пользователю открывается доступ к работе с БД.

```
Password for user ffedorov:
psql (12.22)
Type "help" for help.

adb=>
```

Ошибки LDAP-аутентификации

- При попытке подключения к ADB под пользователем, не включенным в роль `ldap_users`, формируется следующая ошибка:

```
psql: FATAL:  no pg_hba.conf entry for host "10.92.42.236", user "bds", database "adb", SSL off
```

- Если пользователь, под которым выполняется подключение, существует в ADB и включен в роль `ldap_users`, но при этом он отсутствует на стороне LDAP или его пароль не совпадает с введенным, возвращается следующая ошибка:

```
psql: FATAL:  LDAP authentication failed for user "bds"
```

В обоих случаях необходимо дождаться синхронизации с LDAP-сервером по расписанию либо запустить ее вручную, применив действие **Reconfigure** к сервису ADB.

Использование PAM для LDAP-аутентификации в ADB

Содержание

- Установка и настройка сервера OpenLDAP
  - Настройка пакета openldap-servers сервера OpenLDAP
  - Установка пакетов OpenLDAP
  - Настройка сервера OpenLDAP
  - Добавление базовых групп пользователей
  - Создание пользователя
  - Проверка аутентификации
- Установка и настройка ADB
- Настройка хоста координатора ADB
- Отключение возможности анонимной привязки

Назад к содержанию

Использование PAM для LDAP-аутентификации в ADB

Настройка сервера LDAP для использования с ADB

Подготовительные шаги для использования Pluggable Authentication Modules, PAM — это набор разделяемых библиотек, обеспечивающий возможность использования различных методов аутентификации. Вместо того, чтобы приложение напрямую обращалось к системе аутентификации, оно вызывает стандартный интерфейс PAM, который затем обрабатывает процесс аутентификации, используя настроенные модули. Это позволяет гибко менять способы аутентификации без изменения кода приложения — например, переходить с локальных паролей на LDAP, Kerberos или двухфакторную аутентификацию.

Настройка сервера LDAP и его интеграция с ADB позволяет использовать единую учетную запись для входа на все серверы кластера, упрощает управление доступом и обеспечивает централизованный аудит всех подключений.

Установка и настройка сервера OpenLDAP

Настройка сервера LDAP и его интеграция с ADB позволяет использовать единую учетную запись для входа на все серверы кластера, упрощает управление доступом и обеспечивает централизованный аудит всех подключений.

Настройка пакета openldap-servers сервера OpenLDAP

1. Выполнить следующую команду для установки пакета openldap-servers на ОС CentOS/RedHat/Oracle Linux 7/8/9:

```
[root@localhost]# yum install openldap-servers
```

2. Скопировать для корневого каталога базы данных:

```
[root@localhost]# cp /usr/share/doc/openldap-servers-2.4.44/DB_CONFIG /var/lib/ldap/
```

3. Изменить для корневого каталога базы данных:

```
[root@localhost]# chown -R ldap:ldap /var/lib/ldap/
```

4. Проверить изменения:

```
[root@localhost]# ls -ld /var/lib/ldap
```

Результат:

```
drwxrwx---. 2 ldap ldap 4096 Feb 18 21:11 /var/lib/ldap/
```

5. Запустить сервер slapd:

```
[root@localhost]# systemctl start slapd
```

6. Проверить статус сервера slapd:

```
[root@localhost]# systemctl status slapd
```

Результат:

```
slapd.service - OpenLDAP Stand-alone LDAP Daemon
   Loaded: loaded (/usr/lib/systemd/system/slapd.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-02-19 21:11:36 UTC; 7s ago
     Docs: man:slapd
           man:slapd-config
           file:///usr/share/doc/openldap-servers-2.4.44/guide/admin-guide/index.html
 Main PID: 1234 (slapd)
   CGroup: /system.slice/slapd.service
           └─1234 /usr/libexec/slapd -h ldapi:/// ldap:///
Feb 19 21:11:36 localhost.localdomain systemd[1]: Starting OpenLDAP Stand-alone LDAP Daemon...
Feb 19 21:11:36 localhost.localdomain slapd[1234]: slapd starting
Feb 19 21:11:36 localhost.localdomain systemd[1]: Started OpenLDAP Stand-alone LDAP Daemon.
```

Установка пакетов OpenLDAP

1. Обновить репозиторий пакетов, устанавливаемых yum:

```
[root@localhost]# yum update
```

2. Установить пакеты, необходимые для OpenLDAP:

```
[root@localhost]# yum install -y openldap-clients
```

3. Установить и настроить файл /etc/openldap/ldap.conf:

```
[root@localhost]# vi /etc/openldap/ldap.conf
```

Вставить следующие строки:

```
BASE dc=adb,dc=local
URI ldapi:///
TLS_CACERTDIR /etc/openldap/cacerts
```

4. Проверить статус пакетов:

```
[root@localhost]# rpm -qa | grep -i openldap
```

Результат:

```
openldap-clients-2.4.44-21.el7_9.x86_64
openldap-servers-2.4.44-21.el7_9.x86_64
openldap-2.4.44-21.el7_9.x86_64
```

Настройка сервера OpenLDAP

В процессе настройки вы можете будет сталкиваться с ошибками вида:

```
ldap_modify: Server is unwilling to perform (53)
  additional info: olcDatabase: no such value
```

Это означает, что сервер LDAP не распознает изменения, которые вы пытаетесь внести. Причина обычно в том, что не загружены необходимые схемы (cosine, inetorgperson, nis).

Решение: добавьте в конфигурацию нужные схемы перед тем, как вносить изменения в базу данных.

1. Выполнить следующие команды для генерации хешированного пароля для пользователя root:

```
[root@localhost]# slappasswd -h {SSHA} -s root
{SSHA}Ebfc+Ed+aNHZ+Gn9qDIbUf6qk+Z9tAPNx
```

2. Сгенерировать файл конфигурации для подключения к серверу OpenLDAP:

```
[root@localhost]# vi /root/rootldap.ldif
```

Вставить следующие строки:

```
dn: olcDatabase={0}config,cn=config
changetype: modify
add: olcRootPW
olcRootPW: {SSHA}Ebfc+Ed+aNHZ+Gn9qDIbUf6qk+Z9tAPNx
```

3. Выполнить следующую команду для применения изменений:

```
[root@localhost]# ldapadd -Y EXTERNAL -H ldapi:/// -f /root/rootldap.ldif
```

Результат:

```
SASL/EXTERNAL authentication started
SASL username: gidnumber=0+uidnumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={0}config,cn=config"
```

4. Выполнить следующую команду для проверки изменений:

```
[root@localhost]# ldapsearch -Y EXTERNAL -H ldapi:///
```

Результат:

```
SASL/EXTERNAL authentication started
SASL username: gidnumber=0+uidnumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={0}config"
modifying entry "olcDatabase={1}monitor"
modifying entry "olcDatabase={2}hdb"
modifying entry "olcDatabase={3}hdb"
modifying entry "olcDatabase={4}monitor"
```

Добавление базовых групп пользователей

1. Создать файл для формата открытой информации:

```
[root@localhost]# vi /root/groups.ldif
```

Вставить следующие строки:

```
dn: ou=People,dc=adb,dc=local
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=adb,dc=local
objectClass: organizationalUnit
ou: Group
```

2. Сохранить изменения и выйти:

```
:wq
```

3. Выполнить следующую команду для добавления нового файла в сервер LDAP:

```
[root@localhost]# ldapadd -x -D "cn=root,dc=adb,dc=local" -W -f /root/groups.ldif
```

Вводите пароль root, созданный при настройке сервера.

Результат успешного выполнения команды:

```
adding new entry "ou=People,dc=adb,dc=local"
adding new entry "ou=Group,dc=adb,dc=local"
```

4. Выполнить следующую команду для проверки того, что группы успешно созданы:

```
[root@localhost]# ldapsearch -x -b "dc=adb,dc=local" -LLL
```

Результат (сокращен информацией о следующем шагеПо):

```
dn: dc=adb,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: adb
dc: adb

dn: ou=People,dc=adb,dc=local
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=adb,dc=local
objectClass: organizationalUnit
ou: Group
```

Создание пользователя

1. Сгенерировать пароль пользователя:

```
[root@localhost]# slappasswd -h {SSHA} -s pwd
{SSHA}ZvzzXqzqzqzqzqzqzqzqzqzqzqzqzqzq
```

Вставьте полученную строку (начинающуюся с {SSHA}) для дальнейшего использования.

2. Создать файл для LDIF формата открытой информации:

```
[root@localhost]# vi /root/user.ldif
```

Вставьте следующие строки:

```
dn: cn=pwd,ou=People,dc=adb,dc=local
objectClass: posixAccount
objectClass: shadowAccount
cn: pwd
uid: pwd
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/pwd
loginShell: /bin/bash
gecos: pwd
shadowLastChange: 0
shadowMax: 99999
shadowWarning: 7
```

2. Сохранить изменения и выйти:

```
:wq
```

3. Выполнить следующую команду для добавления нового пользователя в сервер LDAP:

```
[root@localhost]# ldapadd -x -D "cn=root,dc=adb,dc=local" -W -f /root/user.ldif
```

Вводите пароль root при подтверждении операции.

Результат успешного выполнения операции:

```
adding new entry "cn=pwd,ou=People,dc=adb,dc=local"
```

4. Проверить, что пользователь успешно создан:

```
[root@localhost]# ldapsearch -x -b "dc=adb,dc=local" "(uid=pwd)" -LLL
```

Результат:

```
dn: cn=pwd,ou=People,dc=adb,dc=local
objectClass: posixAccount
objectClass: shadowAccount
cn: pwd
uid: pwd
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/pwd
loginShell: /bin/bash
gecos: pwd
shadowLastChange: 0
shadowMax: 99999
shadowWarning: 7
```

Отключение возможности анонимной привязки

ВАЖНО!

Если вы планируете развертывание нескольких серверов (рекомендуется 3 и более), то настройка LDAP выполняется аналогично, кроме ADB. Для настройки LDAP на нескольких ADB — см. раздел «Настройка репликации openldap» на странице Настройка высокой доступности ADB.

Если требуется полностью исключить возможность успешной привязки (рекомендуется для продуктивных сред), дополнительно отредактировать схему:

1. Создать файл для LDIF формата открытой информации:

```
[root@localhost]# vi /root/disable_bind_anon.ldif
```

Добавьте в файл следующие строки:

```
dn: olcDatabase={-1}frontend,cn=config
changetype: modify
add: olcDisallows
olcDisallows: bind_anon

dn: olcDatabase={0}config,cn=config
changetype: modify
add: olcDisallows
olcDisallows: bind_anon

dn: olcDatabase={1}monitor,cn=config
changetype: modify
add: olcDisallows
olcDisallows: bind_anon
```

2. Сохранить изменения и выйти:

```
:wq
```

3. Выполнить следующую команду для применения изменений:

```
[root@localhost]# ldapadd -Y EXTERNAL -H ldapi:/// -f /root/disable_bind_anon.ldif
```

Результат:

```
SASL/EXTERNAL authentication started
SASL username: gidnumber=0+uidnumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
modifying entry "olcDatabase={-1}frontend,cn=config"
modifying entry "olcDatabase={0}config,cn=config"
modifying entry "olcDatabase={1}monitor,cn=config"
```

Настройка хоста координатора ADB

Настройка сервера LDAP для использования с ADB

1. Обновить репозиторий пакетов, устанавливаемых yum:

```
[root@localhost]# yum update
```

2. Установить пакет pam-ldapd:

```
[root@localhost]# yum install -y pam-ldapd
```

3. Скопировать файлы конфигурации. Только первый раз для каждого сервера ADB:

```
[root@localhost]# cp /usr/share/doc/nss-pam-ldapd-0.7.19/pam.d/ldap /etc/pam.d/
[root@localhost]# cp /usr/share/doc/nss-pam-ldapd-0.7.19/nslcd.conf /etc/nslcd.conf
```

4. Изменить для файла /etc/nslcd.conf:

```
[root@localhost]# vi /etc/nslcd.conf
```

Вставьте следующие строки:

```
# Connection to LDAP server.
uri ldapi:///
# Search for LDAP users.
base dc=adb,dc=local
# Use StartTLS.
ssl start_tls
# Require and verify server certificate.
tls_reqcert demand
# Certificate authorities.
tls_cacertdir /etc/openldap/cacerts
```

5. Сохранить изменения и выйти:

```
:wq
```

6. Выполнить следующую команду для проверки изменений:

```
[root@localhost]# cat /usr/share/doc/nss-pam-ldapd-0.7.19/pam.d/ldap
```

Результат:

```
# Authentication.
auth required pam_ldap.so
# Accounts.
account required pam_ldap.so
# Password.
password required pam_ldap.so
# Sessions.
session required pam_ldap.so
```

7. Выполнить следующую команду для проверки изменений:

```
[root@localhost]# cat /etc/nslcd.conf | grep -v '^#' | grep -v '^$'
```

Результат:

```
uid nslcd
gid ldap
uri ldapi:///
base dc=adb,dc=local
ssl start_tls
tls_reqcert demand
tls_cacertdir /etc/openldap/cacerts
```

Проверка аутентификации

Для проверки PAM-аутентификации с LDAP-пользователем необходимо обеспечить доступ к файлам конфигурации сервера:

1. Установить утилиту chmod и выдать права доступа для файлов конфигурации сервера:

```
[root@localhost]# yum install -y chmod
```

2. Обновить для root группу ldap (группу openldap) командой usermod -a -G ldap root. Для проверки выполнить id root. Ожидаемый вывод:

```
uid=0(root) gid=0(root) groups=0(root),1000(ldap) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

3. Предоставить доступ к файлам конфигурации сервера:

```
[root@localhost]# chmod -R 750 /var/log/slapd.log /var/log/nslcd.log /var/log/secure /var/log/messages /var/log/audit/audit.log
```

4. Проверить изменения:

```
[root@localhost]# ls -lrtd /var/log/slapd.log /var/log/nslcd.log /var/log/secure /var/log/messages /var/log/audit/audit.log
```

Ожидаемый вывод:

```
-rwxr-xr-x. 1 root root 0 Feb 19 21:11 /var/log/slapd.log
-rwxr-xr-x. 1 root root 0 Feb 19 21:11 /var/log/nslcd.log
-rwxr-xr-x. 1 root root 0 Feb 19 21:11 /var/log/secure
-rwxr-xr-x. 1 root root 0 Feb 19 21:11 /var/log/messages
-rwxr-xr-x. 1 root root 0 Feb 19 21:11 /var/log/audit/audit.log
```

5. Выполнить следующую команду для проверки аутентификации:

```
[root@localhost]# pam_auth -v -u pwd -p pwd
```

Ожидаемый вывод:

```
pam_authenticate() succeeded
pam_acct_mgmt() succeeded
```

6. Перезапустить сервер slapd:

```
[root@localhost]# systemctl restart slapd
```

7. Перезапустить сервер nslcd:

```
[root@localhost]# systemctl restart nslcd
```

8. Проверить статус сервера slapd:

```
[root@localhost]# systemctl status slapd
```

Ожидаемый вывод:

```
slapd.service - OpenLDAP Stand-alone LDAP Daemon
   Loaded: loaded (/usr/lib/systemd/system/slapd.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-02-19 21:11:36 UTC; 7s ago
     Docs: man:slapd
           man:slapd-config
           file:///usr/share/doc/openldap-servers-2.4.44/guide/admin-guide/index.html
 Main PID: 1234 (slapd)
   CGroup: /system.slice/slapd.service
           └─1234 /usr/libexec/slapd -h ldapi:/// ldap:///
Feb 19 21:11:36 localhost.localdomain systemd[1]: Starting OpenLDAP Stand-alone LDAP Daemon...
Feb 19 21:11:36 localhost.localdomain slapd[1234]: slapd starting
Feb 19 21:11:36 localhost.localdomain systemd[1]: Started OpenLDAP Stand-alone LDAP Daemon.
```

9. Проверить статус сервера nslcd:

```
[root@localhost]# systemctl status nslcd
```

Ожидаемый вывод:

```
nslcd.service - Name Service LDAP Daemon
   Loaded: loaded (/usr/lib/systemd/system/nslcd.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-02-19 21:11:36 UTC; 7s ago
     Docs: man:nslcd(8)
           man:nslcd.conf(5)
 Main PID: 1235 (nslcd)
   CGroup: /system.slice/nslcd.service
           └─1235 /usr/sbin/nslcd
Feb 19 21:11:36 localhost.localdomain systemd[1]: Starting Name Service LDAP Daemon...
Feb 19 21:11:36 localhost.localdomain nslcd[1235]: version 0.7.19 (commit 2ad0ccf) (cfg 0x0000000000000000)
Feb 19 21:11:36 localhost.localdomain nslcd[1235]: nslcd: no valid authentication method or password specified
Feb 19 21:11:36 localhost.localdomain systemd[1]: Started Name Service LDAP Daemon.
```

10. Выполнить следующую команду для проверки аутентификации:

```
[root@localhost]# pam_auth -v -u pwd -p pwd
```

Ожидаемый вывод:

```
pam_authenticate() succeeded
pam_acct_mgmt() succeeded
```

11. Выполнить следующую команду для проверки изменений:

```
[root@localhost]# id pwd
```

Ожидаемый вывод:

```
uid=1000(pwd) gid=1000(pwd) groups=1000(pwd),1000(ldap) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

12. Отредактировать файл /etc/sysconfig/slapd:

```
[root@localhost]# vi /etc/sysconfig/slapd
```

Вставьте следующие строки:

```
# Configuration options for the OpenLDAP daemon.
# Interface to bind to, default is all interfaces.
SLAPD_LDAPI=yes
SLAPD_LDAP=yes
SLAPD_LDAPS=no
# Options passed to the daemon.
SLAPD_OPTIONS="-h ldapi:/// ldap:///"
```

13. Сохранить изменения и выйти:

```
:wq
```

14. Перезапустить сервер slapd:

```
[root@localhost]# systemctl restart slapd
```

15. Проверить статус сервера slapd:

```
[root@localhost]# systemctl status slapd
```

Ожидаемый вывод:

```
slapd.service - OpenLDAP Stand-alone LDAP Daemon
   Loaded: loaded (/usr/lib/systemd/system/slapd.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-02-19 21:11:36 UTC; 7s ago
     Docs: man:slapd
           man:slapd-config
           file:///usr/share/doc/openldap-servers-2.4.44/guide/admin-guide/index.html
 Main PID: 1234 (slapd)
   CGroup: /system.slice/slapd.service
           └─1234 /usr/libexec/slapd -h ldapi:/// ldap:///
Feb 19 21:11:36 localhost.localdomain systemd[1]: Starting OpenLDAP Stand-alone LDAP Daemon...
Feb 19 21:11:36 localhost.localdomain slapd[1234]: slapd starting
Feb 19 21:11:36 localhost.localdomain systemd[1]: Started OpenLDAP Stand-alone LDAP Daemon.
```

16. Проверить статус сервера nslcd:

```
[root@localhost]# systemctl status nslcd
```

Ожидаемый вывод:

```
nslcd.service - Name Service LDAP Daemon
   Loaded: loaded (/usr/lib/systemd/system/nslcd.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-02-19 21:11:36 UTC; 7s ago
     Docs: man:nslcd(8)
           man:nslcd.conf(5)
 Main PID: 1235 (nslcd)
   CGroup: /system.slice/nslcd.service
           └─1235 /usr/sbin/nslcd
Feb 19 21:11:36 localhost.localdomain systemd[1]: Starting Name Service LDAP Daemon...
Feb 19 21:11:36 localhost.localdomain nslcd[1235]: version 0.7.19 (commit 2ad0ccf) (cfg 0x0000000000000000)
Feb 19 21:11:36 localhost.localdomain nslcd[1235]: nslcd: no valid authentication method or password specified
Feb 19 21:11:36 localhost.localdomain systemd[1]: Started Name Service LDAP Daemon.
```

17. Выполнить следующую команду для проверки аутентификации:

```
[root@localhost]# pam_auth -v -u pwd -p pwd
```

Ожидаемый вывод:

```
pam_authenticate() succeeded
pam_acct_mgmt() succeeded
```

18. Выполнить следующую команду для проверки изменений:

```
[root@localhost]# id pwd
```

Ожидаемый вывод:

```
uid=1000(pwd) gid=1000(pwd) groups=1000(pwd),1000(ldap) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

19. Отредактировать файл /etc/pam.d/sshd:

```
[root@localhost]# vi /etc/pam.d/sshd
```

Вставьте следующие строки:

```
#%PAM-1.0
auth       required     pam_sepermit.so
auth       substack     password-auth
auth       include      postlogin
account    required     pam_sepermit.so
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be before pam_selinux.so open
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed
# in the user context
session    required     pam_selinux.so open
session    optional     pam_keyinit.so force revoke
session    include      password-auth
session    include      postlogin
# pam_ldap.so
auth       sufficient   pam_ldap.so
account    sufficient   pam_ldap.so
password   sufficient   pam_ldap.so
session    optional     pam_ldap.so
```

20. Сохранить изменения и выйти:

```
:wq
```

21. Отредактировать файл /etc/sysconfig/authconfig:

```
[root@localhost]# vi /etc/sysconfig/authconfig
```

Вставьте следующие строки:

```
USEMKHOMEDIR=yes
USEPAMACCESS=yes
USELDAPAUTH=yes
USELDAP=yes
USELOCAUTHORIZE=yes
USEPAMPASSWD=yes
USESHADOW=yes
USEMD5=yes
USEPAMALWAYS=yes
USEPAMSESSION=yes
USEPAMPASSWORD=yes
USELDAPTLS=yes
USEPAMACCES
```

Аутентификация с использованием MIT Kerberos KDC в ADB

Версия 1.0

Содержание

Перейдите к содержанию

Шаг 1. Установка и настройка MIT Kerberos KDC
Шаг 2. Создание принципалов для ADB и MIT Kerberos KDC
Шаг 3. Настройка хоста кластера ADB в проверке аутентификации
Шаг 4. Сопоставление ролей ADB с принципалами Kerberos KDC
Шаг 5. Настройка Kerberos аутентификации для клиентских приложений
Шаг 6. Настройка JDBC-подключений

Шаг 1. Установка и настройка MIT Kerberos KDC

Выполняется на сервере Linux (в данном примере Oracle Linux 8).

Также можно развернуть службу Kerberos в среде Oracle Cloud Infrastructure (OCI) — см. **MIT Kerberos for Linux Systems** (Руководство по Oracle Cloud Infrastructure). В этом случае пропустите шаги 1.1–1.3 и перейдите к шагу 1.4 (создание принципалов).

1.1 Установка пакетов

```
[root@kdc-server ~]# dnf install krb5-server krb5-workstation
```

1.2 Настройка /etc/krb5.conf

Создайте файл **/etc/krb5.conf** (замените **MYREALM.COM** на ваше имя домена Kerberos, например **EXAMPLE.COM**):

```
[root@kdc-server ~]# vi /etc/krb5.conf
[root@kdc-server ~]# cat /etc/krb5.conf
# Configuration snippets may be placed in this directory instead.
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
spake_preauth_kinds = enc_timestamp
[root@kdc-server ~]
[root@kdc-server ~]# vi /var/kerberos/krb5kdc/kdc.conf
[root@kdc-server ~]# cat /var/kerberos/krb5kdc/kdc.conf
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
 MYREALM.COM = {
 database_name = /var/kerberos/krb5kdc/principal
 key_stash_file = /var/kerberos/krb5kdc/stash
 acl_file = /var/kerberos/krb5kdc/kadm5.acl
 max_renewable_life = 7d 0h 0m 0s
 supported_enctypes = aes256-cts:normal aes128-cts:normal
 default_principal_flags = +preauth
}
[root@kdc-server ~]
```

1.3 Настройка /var/kerberos/krb5kdc/kadm5.acl

Создайте файл **/var/kerberos/krb5kdc/kadm5.acl** (замените **MYREALM.COM** на ваше имя домена Kerberos):

```
[root@kdc-server ~]# vi /var/kerberos/krb5kdc/kadm5.acl
[root@kdc-server ~]# cat /var/kerberos/krb5kdc/kadm5.acl
*/admin@MYREALM.COM	*
[root@kdc-server ~]
```

1.4 Создание базы данных Kerberos

```
[root@kdc-server ~]# kdb5_util create -s -r MYREALM.COM
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for realm 'MYREALM.COM',
master key name 'K/M@MYREALM.COM'
[root@kdc-server ~]
```

1.5 Запуск служб KDC и kadmin

```
[root@kdc-server ~]# systemctl start krb5kdc kadmin
[root@kdc-server ~]# systemctl enable krb5kdc kadmin
[root@kdc-server ~]# systemctl status krb5kdc
● krb5kdc.service - Kerberos 5 KDC
   Loaded: loaded (/usr/lib/systemd/system/krb5kdc.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-12-17 09:41:12 GMT; 1 day 3 hours ago
 Main PID: 1234 (krb5kdc)
   CGroup: /system.slice/krb5kdc.service
           └─1234 /usr/sbin/krb5kdc -P /var/run/krb5kdc/krb5kdc.pid -S
[root@kdc-server ~]# systemctl status kadmin
● kadmin.service - Kerberos Administration Daemon
   Loaded: loaded (/usr/lib/systemd/system/kadmin.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-12-17 09:41:12 GMT; 1 day 3 hours ago
 Main PID: 1235 (kadmind)
   CGroup: /system.slice/kadmin.service
           └─1235 /usr/sbin/kadmind -P /var/run/kadmin/kadmin.pid -S
[root@kdc-server ~]
```

1.6 Проверка работоспособности KDC

```
[root@kdc-server ~]# kinit -p admin/admin@MYREALM.COM
Authenticated with proxy.
Password for admin/admin@MYREALM.COM:
[root@kdc-server ~]# klist -ef
Ticket cache: KCM:0:58800
Default principal: admin/admin@MYREALM.COM

Valid starting       Expires              Service principal
12/18/25 09:41:15    12/18/25 09:41:12    krbtgt/MYREALM.COM@MYREALM.COM
  renew until 12/25/25 09:41:12, Flags: FRIA
  Etype (skey, tkt): aes256-cts-hmac-sha384-192, aes256-cts-hmac-sha384-192
[root@kdc-server ~]
```

Шаг 2. Создание принципалов для ADB и MIT Kerberos KDC

Для аутентификации в кластере ADB требуются два принципала: один для сервера ADB (**adb/hostname**) и один для клиента (**client/hostname**). Оба принципала должны быть экспортированы в keytab-файлы и развернуты на соответствующих хостах.

Примечание. Для простоты в данном примере используется один keytab-файл для обоих принципалов. В производственной среде рекомендуется использовать отдельные keytab-файлы для каждого принципала.

2.1 Создание принципала для сервера ADB

```
[root@kdc-server ~]# kadmin.local -q "addprinc -randkey adb/adb.example.com@MYREALM.COM"
Authenticating as principal root/admin@MYREALM.COM with password.
WARNING: no policy specified for adb/adb.example.com@MYREALM.COM; defaulting to no policy
Principal "adb/adb.example.com@MYREALM.COM" created.
[root@kdc-server ~]
```

2.2 Создание принципала для клиента

```
[root@kdc-server ~]# kadmin.local -q "addprinc client/adb.example.com@MYREALM.COM"
Authenticating as principal root/admin@MYREALM.COM with password.
WARNING: no policy specified for client/adb.example.com@MYREALM.COM; defaulting to no policy
Principal "client/adb.example.com@MYREALM.COM" created.
[root@kdc-server ~]
```

2.3 Экспорт keytab-файлов

```
[root@kdc-server ~]# kadmin.local -q "ktadd -k /tmp/adb.keytab adb/adb.example.com@MYREALM.COM"
Authenticating as principal root/admin@MYREALM.COM with password.
WARNING: no policy specified for adb/adb.example.com@MYREALM.COM; defaulting to no policy
Entry for principal adb/adb.example.com@MYREALM.COM with kvno 2, encryption type aes256-cts-hmac-sha384-192 added to keytab WRFILE:/tmp/adb.keytab.
Entry for principal adb/adb.example.com@MYREALM.COM with kvno 2, encryption type aes128-cts-hmac-sha256-128 added to keytab WRFILE:/tmp/adb.keytab.
[root@kdc-server ~]
[root@kdc-server ~]# kadmin.local -q "ktadd -k /tmp/client.keytab client/adb.example.com@MYREALM.COM"
Authenticating as principal root/admin@MYREALM.COM with password.
WARNING: no policy specified for client/adb.example.com@MYREALM.COM; defaulting to no policy
Entry for principal client/adb.example.com@MYREALM.COM with kvno 2, encryption type aes256-cts-hmac-sha384-192 added to keytab WRFILE:/tmp/client.keytab.
Entry for principal client/adb.example.com@MYREALM.COM with kvno 2, encryption type aes128-cts-hmac-sha256-128 added to keytab WRFILE:/tmp/client.keytab.
[root@kdc-server ~]
[root@kdc-server ~]# klist -kt /tmp/adb.keytab
Keytab name: FILE:/tmp/adb.keytab
KVNO Timestamp         Principal
   2 12/18/25 09:41:18 adb/adb.example.com@MYREALM.COM
   2 12/18/25 09:41:18 adb/adb.example.com@MYREALM.COM
[root@kdc-server ~]
[root@kdc-server ~]# klist -kt /tmp/client.keytab
Keytab name: FILE:/tmp/client.keytab
KVNO Timestamp         Principal
   2 12/18/25 09:41:18 client/adb.example.com@MYREALM.COM
   2 12/18/25 09:41:18 client/adb.example.com@MYREALM.COM
[root@kdc-server ~]
```

Шаг 3. Настройка хоста кластера ADB в проверке аутентификации

Выполняется на сервере Linux (в данном примере Oracle Linux 8).

3.1 Установка пакетов Kerberos и настройка /etc/krb5.conf

```
[root@adb-host ~]# dnf install krb5-workstation
[root@adb-host ~]# vi /etc/krb5.conf
[root@adb-host ~]# cat /etc/krb5.conf
# Configuration snippets may be placed in this directory instead.
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
spake_preauth_kinds = enc_timestamp
[root@adb-host ~]
```

3.2 Копирование keytab-файлов на хост ADB

Скопируйте keytab-файлы с сервера KDC на хост ADB:

```
[root@kdc-server ~]# scp /tmp/adb.keytab /tmp/client.keytab opc@adb-host:/tmp/
adb.keytab                                    100%  1234    1.2MB/s   00:00
client.keytab                                 100%  1234    1.2MB/s   00:00
[root@kdc-server ~]
```

3.3 Настройка keytab-файлов и их проверка

```
[root@adb-host ~]# mkdir -p /etc/security/keytabs
[root@adb-host ~]# mv /tmp/adb.keytab /tmp/client.keytab /etc/security/keytabs/
[root@adb-host ~]# chmod 600 /etc/security/keytabs/*.keytab
[root@adb-host ~]# chown root:root /etc/security/keytabs/*.keytab
[root@adb-host ~]# ls -l /etc/security/keytabs/
total 8
-rw------- 1 root root 1234 Dec 18 09:41 adb.keytab
-rw------- 1 root root 1234 Dec 18 09:41 client.keytab
[root@adb-host ~]
[root@adb-host ~]# klist -kt /etc/security/keytabs/adb.keytab
Keytab name: FILE:/etc/security/keytabs/adb.keytab
KVNO Timestamp         Principal
   2 12/18/25 09:41:18 adb/adb.example.com@MYREALM.COM
   2 12/18/25 09:41:18 adb/adb.example.com@MYREALM.COM
[root@adb-host ~]
[root@adb-host ~]# klist -kt /etc/security/keytabs/client.keytab
Keytab name: FILE:/etc/security/keytabs/client.keytab
KVNO Timestamp         Principal
   2 12/18/25 09:41:18 client/adb.example.com@MYREALM.COM
   2 12/18/25 09:41:18 client/adb.example.com@MYREALM.COM
[root@adb-host ~]
```

Шаг 4. Сопоставление ролей ADB с принципалами Kerberos KDC

В отличие от Oracle Database, где используется файл **$ORACLE_HOME/network/admin/kerberos.ora**, в ADB для настройки сопоставления используется пакет **DBMS_CLOUD_ADMIN**.

4.1 Проверка доступности пакета DBMS_CLOUD_ADMIN

```
SQL> SELECT object_name, status FROM dba_objects WHERE object_name = 'DBMS_CLOUD_ADMIN';

OBJECT_NAME          STATUS
-------------------- -------
DBMS_CLOUD_ADMIN     VALID

SQL>
```

4.2 Сопоставление роли ADB с принципалом Kerberos

Пакет **DBMS_CLOUD_ADMIN** предоставляет процедуры для управления сопоставлениями внешних идентификаторов с ролями базы данных.

```
SQL> -- Сопоставить роль ADMIN с принципалом adb/adb.example.com@MYREALM.COM
SQL> BEGIN
  2  DBMS_CLOUD_ADMIN.CREATE_EXTERNAL_ROLE_MAPPING(
  3  external_identity => 'adb/adb.example.com@MYREALM.COM',
  4  database_role => 'ADMIN');
  5  END;
  6  /

PL/SQL procedure successfully completed.

SQL>
SQL> -- Проверить сопоставления
SQL> SELECT * FROM DBA_CLOUD_EXTERNAL_ROLE_MAPPINGS;

EXTERNAL_IDENTITY                    DATABASE_ROLE
------------------------------------ -------------
adb/adb.example.com@MYREALM.COM      ADMIN

SQL>
SQL> -- Удалить сопоставление (если нужно)
SQL> BEGIN
  2  DBMS_CLOUD_ADMIN.DROP_EXTERNAL_ROLE_MAPPING(
  3  external_identity => 'adb/adb.example.com@MYREALM.COM');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL>
SQL> -- Снова добавить сопоставление
SQL> BEGIN
  2  DBMS_CLOUD_ADMIN.CREATE_EXTERNAL_ROLE_MAPPING(
  3  external_identity => 'adb/adb.example.com@MYREALM.COM',
  4  database_role => 'ADMIN');
  5  END;
  6  /

PL/SQL procedure successfully completed.

SQL>
```

Шаг 5. Настройка Kerberos аутентификации для клиентских приложений

Выполняется на сервере Linux (в данном примере Oracle Linux 8).

5.1 Установка пакетов

```
[root@client-host ~]# dnf install krb5-workstation
[root@client-host ~]# vi /etc/krb5.conf
[root@client-host ~]# cat /etc/krb5.conf
# Configuration snippets may be placed in this directory instead.
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
spake_preauth_kinds = enc_timestamp
[root@client-host ~]
```

5.2 Настройка /etc/krb5.conf

Создайте файл **/etc/krb5.conf** (замените **MYREALM.COM** на ваше имя домена Kerberos):

```
[root@client-host ~]# vi /etc/krb5.conf
[root@client-host ~]# cat /etc/krb5.conf
# Configuration snippets may be placed in this directory instead.
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
spake_preauth_kinds = enc_timestamp
[root@client-host ~]
```

5.3 Копирование keytab-файлов на клиентский хост

```
[root@kdc-server ~]# scp /tmp/client.keytab opc@client-host:/tmp/
client.keytab                                 100%  1234    1.2MB/s   00:00
[root@kdc-server ~]
```

5.4 Настройка keytab-файлов и их проверка

```
[root@client-host ~]# mkdir -p /etc/security/keytabs
[root@client-host ~]# mv /tmp/client.keytab /etc/security/keytabs/
[root@client-host ~]# chmod 600 /etc/security/keytabs/client.keytab
[root@client-host ~]# chown root:root /etc/security/keytabs/client.keytab
[root@client-host ~]# klist -kt /etc/security/keytabs/client.keytab
Keytab name: FILE:/etc/security/keytabs/client.keytab
KVNO Timestamp         Principal
   2 12/18/25 09:41:18 client/adb.example.com@MYREALM.COM
   2 12/18/25 09:41:18 client/adb.example.com@MYREALM.COM
[root@client-host ~]
```

Ошибки Kerberos-аутентификации

Ошибка	Причина
ORA-01017: invalid username/password; logon denied	Неверный принципал или keytab-файл не соответствует ожидаемому принципалу
ORA-12641: Authentication service is not enabled	Не задан параметр sqlnet.authentication_service в sqlnet.ora
ORA-28007: specified authentication service is not valid	Неверный формат имени принципала (должен быть service/hostname@REALM)
ORA-12650: No common encryption or data integrity algorithm	Несовместимые алгоритмы шифрования. Проверьте supported_enctypes в /var/kerberos/krb5kdc/kdc.conf
ORA-28007: specified authentication service is not valid	Истекшее время жизни тикета. Выполните kinit -p -k -t /path/to/keytab principal

Шаг 6. Настройка JDBC-подключений

Для подключения к ADB с использованием Kerberos аутентификации требуется JDBC-драйвер Oracle 23ai или новее. Драйверы JDBC 21c и более ранние версии не поддерживают Kerberos аутентификацию для ADB.

Скачайте последний JDBC-драйвер с сайта Oracle: https://www.oracle.com/database/technology/appdev/jdbc.html

6.1 Добавление записи в tnsnames.ora

```
[oracle@client-host ~]$ cat /opt/oracle/network/admin/tnsnames.ora
# tnsnames.ora Network Configuration File: /opt/oracle/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

MYADB_HIGH =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=120)(RETRY_COUNT=20)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3)
    (ADDRESS = (PROTOCOL = TCPS)(PORT = 1521)(HOST = adb.eu-frankfurt-1.oraclecloud.com))
    (CONNECT_DATA=(SERVICE_NAME=xxxxxxxxxxxxxxxx_myadb_high.adb.oraclecloud.com))
    (SECURITY=(MY_WALLET_DIRECTORY=/opt/oracle/wallet)(SSL_SERVER_DN_MATCH=yes))
  )
[oracle@client-host ~]$
```

6.2 Добавление параметров в sqlnet.ora

```
[oracle@client-host ~]$ cat /opt/oracle/network/admin/sqlnet.ora
# sqlnet.ora Network Configuration File: /opt/oracle/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT, HOSTNAME)

SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY="/opt/oracle/wallet")
    )
  )
SQLNET.AUTHENTICATION_SERVICES= (KERBEROS5)
SQLNET.KERBEROS5_CONF=/opt/oracle/network/admin/krb5.conf
SQLNET.KERBEROS5_CONF_MIT=TRUE
SQLNET.KERBEROS5_KEYTAB=/opt/oracle/network/admin/client.keytab
SQLNET.KERBEROS5_CLOCKSKEW=300
[oracle@client-host ~]$
```

6.3 Настройка JDBC-подключения

```
[oracle@client-host ~]$ export TNS_ADMIN=/opt/oracle/network/admin
[oracle@client-host ~]$ export LD_LIBRARY_PATH=/opt/oracle/lib:$LD_LIBRARY_PATH
[oracle@client-host ~]$ export CLASSPATH=/opt/oracle/jdbc/lib/*:.
[oracle@client-host ~]$ export JAVA_HOME=/usr/lib/jvm/jre-17-oracle-x64
[oracle@client-host ~]$ export PATH=$JAVA_HOME/bin:$PATH
[oracle@client-host ~]$ cat /opt/oracle/jdbc/lib/ojdbc.properties
# ojdbc.properties
# Generated by Oracle configuration tools.

oracle.net.tns_admin=/opt/oracle/network/admin
oracle.net.kerberos5_cache_type=MEMORY
oracle.net.kerberos5_mutual_authentication=true
oracle.net.kerberos5_cache=false
oracle.net.kerberos5_keytab=/opt/oracle/network/admin/client.keytab
oracle.net.kerberos5_conf=/opt/oracle/network/admin/krb5.conf
oracle.net.kerberos5_conf_mit=true
oracle.net.authentication_services=(KERBEROS5)
[oracle@client-host ~]$
[oracle@client-host ~]$ cat /opt/oracle/jdbc/lib/MyConnectionTest.class
[oracle@client-host ~]$
[oracle@client-host ~]$ cat /opt/oracle/jdbc/lib/MyConnectionTest.class | base64 > /tmp/MyConnectionTest.class.b64
[oracle@client-host ~]$ ls -l /tmp/MyConnectionTest.class.b64
-rw-r--r-- 1 oracle oracle 4096 Dec 18 09:41 /tmp/MyConnectionTest.class.b64
[oracle@client-host ~]$
[oracle@client-host ~]$ scp /tmp/MyConnectionTest.class.b64 opc@adb-host:/tmp/
MyConnectionTest.class.b64                    100%  4096    4.0MB/s   00:00
[oracle@client-host ~]$
[oracle@client-host ~]$ exit
[root@client-host ~]# exit
[opc@client-host ~]$ exit
[root@kdc-server ~]# exit
[root@adb-host ~]# exit
[root@kdc-server ~]# exit
[root@adb-host ~]# exit
[root@kdc-server ~]# exit
[root@adb-host ~]#
```

Учетные записи, используемые в ADB

Дарья Барышева

Содержание

Назад к содержанию

Пользователи Linux
Роли Arenadata DB
Учетные записи, используемые в базах данных ADB Control
Пользователи ADB Control UI
Пользователь ADBM API
Пользователь ADCM
Учетные записи LDAP
ADB
ADB Control

Статья содержит полный перечень учетных записей, используемых в Arenadata DB.

Пользователи Linux

Приведенные ниже учетные записи создаются на уровне операционной системы во время установки ADB (/etc/passwd).

Учетная запись	Описание	Пароль по умолчанию	Изменение пароля
adcm	Учетная запись, используемая ADCM для развертывания кластеров	Не установлен	
gpadmin	Учетная запись Linux, необходимая в БД ADB для старта и остановки кластеров, а также для запуска консольных утилит. У пользователя есть права суперюзер ко всем базам данных кластера (метка `trust` в файле *pg_hba.conf*). Имя пользователя этой учетной записи можно изменить с помощью **System user name** — конфигурационного параметра сервиса ADB	Не установлен	
pxf	Учетная запись для PXF. Отмечена как `nologin` в *etc/passwd*	Не установлен	

Роли Arenadata DB

Приведенные ниже учетные записи создаются в базе данных Arenadata DB во время установки (посредством вызова команд `CREATE ROLE`).

Учетная запись	Описание	Пароль по умолчанию	Изменение пароля
gpadmin	Администратор кластера ADB. Эта роль имеет доступ ко всем БД ADB с правами суперпользователя	Не установлен	Выполните следующую команду в `psql`: `ALTER ROLE gpadmin WITH ENCRYPTED PASSWORD '<new_password>';` Нет возможности установить/сбросить пароль в ADCM
adcc	Эта роль используется для подключения из ADB Control к ADB. Доступ необходим для получения информации о топологии кластера и для отмены запросов, инициированных пользователем. Имя пользователя может быть изменено на конфигурационной странице ADB Control при первичной установке ADB Control. Изменение в дальнейшем недоступно. Дополнительную информацию можно получить в статье Конфигурационные параметры (***ADB Control → UI component parameters → ADB user name***)	Не установлен по умолчанию. Пароль не может быть изменен при установке ADB. Пароль хранится в `pg_shadow`	1. Выполните следующую команду в `psql`: `ALTER ROLE adcc WITH ENCRYPTED PASSWORD '<new_password>';` 2. Измените пароль на вкладке ***Configuration → Clusters*** в интерфейсе ADB Control. Дополнительную информацию можно получить в разделе Редактирование подключения к кластеру. Нет возможности установить/сбросить пароль в ADCM

Учетные записи, используемые в базах данных ADB Control

Следующие учетные записи создаются в базах данных ADB Control (внешних либо внутри контейнеров) в ходе установки ADB Control.

Учетная запись	Описание	Пароль по умолчанию	Изменение пароля
postgres	Учетная запись в базе данных PostgreSQL, используемой в ADB Control для хранения информации по запросам (см. Query DB). Имя пользователя может быть изменено на конфигурационной странице ADB Control в ходе установки ADB Control. Дополнительную информацию можно получить в статье Конфигурационные параметры (см. ***ADB Control → Database parameters → Username*** либо ***ADB Control → External database parameters → Username*** для внешнего PostgreSQL)	По умолчанию `123`. Пароль может быть установлен на конфигурационной странице ADB Control в ходе установки ADB Control. Дополнительную информацию можно получить в статье Конфигурационные параметры (см. ***ADB Control → Database parameters → Password*** либо ***ADB Control → External database parameters → Password*** для внешнего PostgreSQL).	При использовании внешнего PostgreSQL изменение пароля в ADCM возможно в любое время. При использовании PostgreSQL в контейнере установка пароля возможна только при исходной установке ADB Control
default	Учетная запись в базе данных ClickHouse, используемой в ADB Control для хранения информации по метрикам (см. Metrics DB). Имя пользователя может быть изменено на конфигурационной странице ADB Control в ходе установки ADB Control. Дополнительную информацию можно получить в статье Конфигурационные параметры (см. ***ADM Control → ADQM database parameters → Username*** либо ***ADB Control → External ADQM parameters → Username*** для внешнего ClickHouse)	Не установлен по умолчанию. Пароль может быть установлен на конфигурационной странице ADB Control в ходе установки ADB Control. Дополнительную информацию можно получить в статье Конфигурационные параметры (см. ***ADB Control → ADQM database parameters → Password*** либо ***ADB Control → External ADQM parameters → Password*** для внешнего ClickHouse). В случае установки пароль хранится в ADCM vault и в ClickHouse	При использовании внешнего ClickHouse изменение пароля в ADCM возможно в любое время. При использовании ClickHouse в контейнере установка пароля возможна только при исходной установке ADB Control

Пользователи ADB Control UI

Следующие учетные записи создаются в ходе установки ADB Control.

Учетная запись	Описание	Пароль по умолчанию	Изменение пароля

admin	Пользователь ADB Control с правами Owner, используемый для работы с интерфейсом ADB Control	Значение по умолчанию — `1234`.
		При первом подключении к ADB Control необходимо изменить пароль. См. Подключение к ADB Control.
		Пароль хранится в таблице `adcc.user` базы данных PostgreSQL (внешней либо внутри контейнера):
		`$ psql -d adcc`
		`SELECT * FROM adcc.user;`
			Изменение пароля возможно на вкладке ***Users → Management*** в интерфейсе ADB Control. См. Редактирование данных пользователя

adcm	Сервисная учетная запись для API (скрытый администратор ADB Control UI)	Пароль может быть установлен на конфигурационной странице ADB Control в ходе установки ADB Control. Дополнительную информацию можно получить в статье Конфигурационные параметры (***ADB Control → UI component parameters → Service password***). Если пароль не заполнен в ADCM, формируется секретный пароль (общий для всех кластеров для обеспечения обратной совместимости). Пароль хранится в зашифрованном виде в контейнере `adcc_database_1`: `SELECT password FROM adcc.system_user WHERE username = 'adcm';`	Пароль может быть изменен на конфигурационной странице ADB Control (***ADB Control → UI component parameters → Service password***)
adbm	Сервисная учетная запись (скрытый администратор ADB Control UI) для API сбора информации об изменениях хранилищ и кластеров, отображаемой на странице ***Audit***	Пароль может быть установлен на конфигурационной странице ADBM. Дополнительную информацию можно получить в статье Конфигурационные параметры (***ADBM → Password for adbm system user***) Пароль хранится в зашифрованном виде в контейнере `adcc_database_1`: `SELECT password FROM adcc.system_user WHERE username = 'adbm';`	Пароль может быть изменен на конфигурационной странице ADBM (***ADBM → Password for adbm system user***)

Пользователь ADBM API

Приведенный ниже пользователь создается в ходе установки ADBM.

Учетная запись	Описание	Пароль по умолчанию	Изменение пароля

adcc	Пользователь, используемый для аутентификации в ADBM CLI API	Хранится как plaintext в файле */opt/adbm/auth.properties*. Значение по умолчанию `1234`.
		`user adcc = 1234`
		Разрешения файла:
		`-rw-r--r-- 1 root root`
			Путем редактирования файла */opt/adbm/auth.properties*. Требуется рестарт ADBM

Пользователь ADCM

Приведенный ниже пользователь создается в ходе установки ADCM.

Учетная запись	Описание	Пароль по умолчанию	Изменение пароля
admin	Администратор ADCM	admin	После установки ADCM изменение пароля возможно в web-интерфейсе ADCM. См. Проверка web-интерфейса ADCM

Учетные записи LDAP

При использовании LDAP-аутентификации необходимо заполнить и сохранить данные административных учетных записей LDAP. Пароли хранятся в виде plaintext в специальных файлах, которые требуется защитить на уровне разрешений файловой системы.

ADB

Настройка LDAP-аутентификации в ADB выполняется в соответствии со статьей Аутентификация LDAP в ADB. Имя сервисной учетной записи LDAP и ее пароль указываются в полях ***SA Login*** и ***Password*** на конфигурационной странице ADB.

Пароль учетной записи LDAP хранится в следующем файле на координатор-хосте ADB: *$COORDINATOR_DATA_DIRECTORY/pg_hba.conf*.

Разрешения файла:

```
-rw-r----- 1 gpadmin gpadmin
```

Синхронизация пользователей и групп ADB с LDAP-сервером выполняется автоматически с помощью утилиты pg ldap sync. См. раздел Синхронизация ADB и LDAP.

ВАЖНО

После синхронизации с LDAP-сервером две локальные роли ADB по-прежнему необходимы и не могут быть заменены пользователями LDAP (см. Роли Arenadata DB):

- `gpadmin`
- `gpmon`

Другие пользователи (включая суперпользователей) могут контролироваться на стороне LDAP-сервера.

ADB Control

Настройка LDAP-аутентификации в ADB Control выполняется в соответствии со статьей Аутентификация с использованием LDAP. Имя сервисной учетной записи LDAP и ее пароль указываются в полях ***Login*** и ***Password*** на конфигурационной странице ADB Control.

Пароль учетной записи LDAP хранится в следующем файле на хосте с ADB Control.

- */opt/adcc/docker-compose.yaml*. Разрешения файла:

```
-rw------ 1 root root
```

- */opt/adcc/adcc-ui-server-application.yml*. Разрешения файла:

```
-rw-r--r-- 1 root root
```

В случае использования LDAP-аутентификации требуется выполнить сопоставление между группами пользователей используемого LDAP-сервера и ролями в ADB Control. Управление подобными сопоставлениями осуществляется на вкладке ***Users → LDAP group mapping*** в web-интерфейсе ADB Control. См. раздел Сопоставление ролей с группами LDAP-серверов.

ВАЖНО

После назначения роли `Administrator` либо `Owner` определенным LDAP-пользователям и удалении/блокировки дефолтного локального пользователя `admin` (см. Пользователи ADB Control UI) весь процесс аутентификации может контролироваться на стороне LDAP-сервера.

Конкретные действия

Содержание

- Обзор
- Activate standby
- Activate standby postprocess
- Init Standby Master
- Move Heart Segment
- Expand
- Shrink
- Remove
- Remount snatacherker
- Reconfigure Vault integration
- Reconfigure parameter archiving
- Check
- Start
- Stop
- Delete
- Create

Перейти к содержанию

Обзор

Конкретные действия выполняются в разделе **Кластеры**.

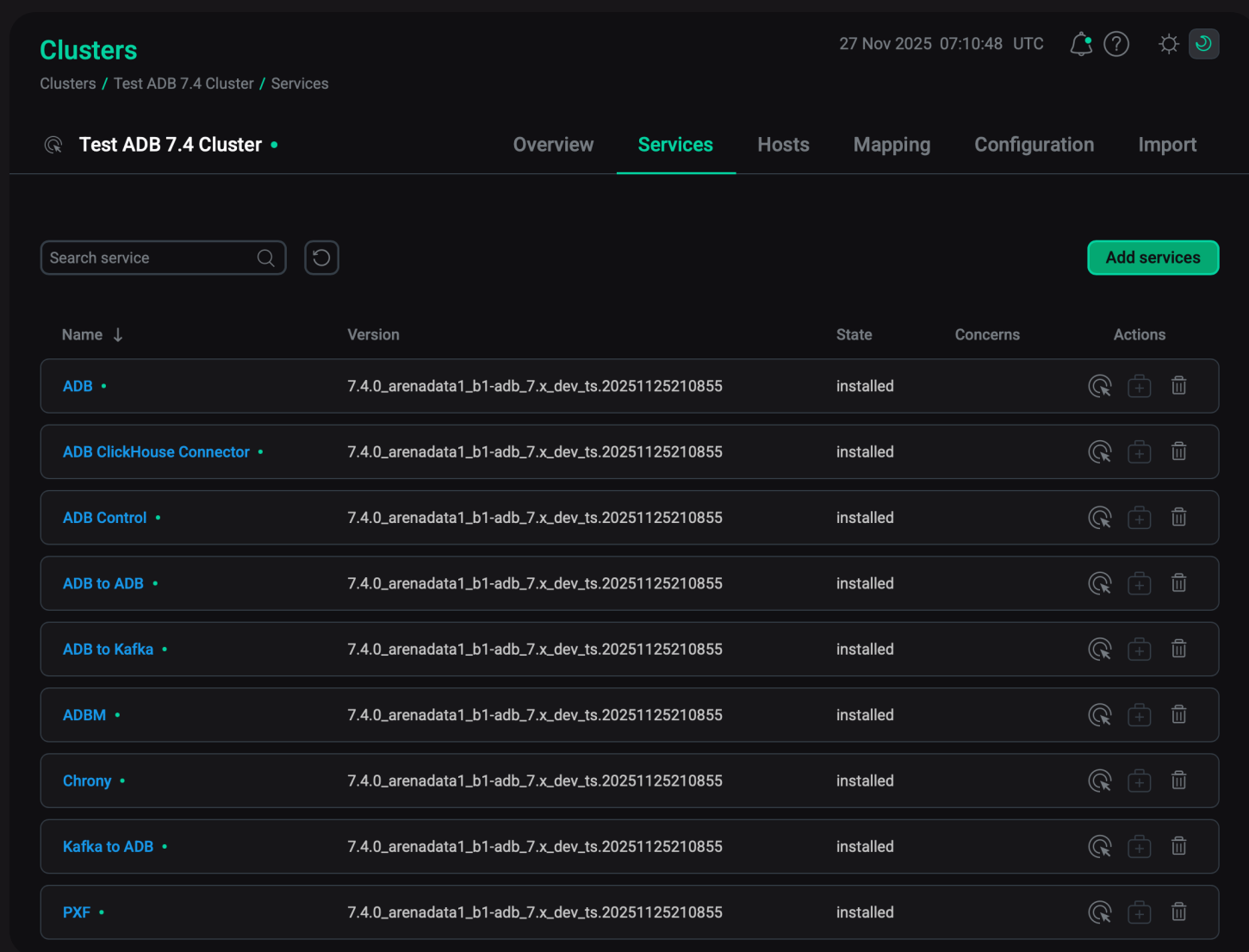
Кластеры
Кластеры
Кластеры

Сервисные действия

Антон Васильев

Назад к содержанию

Интерфейс ADCM позволяет выполнять действия для управления сервисами. Действия доступны на вкладке ***Services***. Чтобы перейти на эту вкладку, нажмите на название кластера на странице ***Clusters*** и откройте вкладку ***Services***.



Сервисы кластера ADB в интерфейсе ADCM

Описание вкладки ***Services*** и общих действий на этой вкладке приведено в документации ADCM.

Действия конкретных сервисов кластера ADB описаны в следующих разделах:

- ADB
- ADB Control
- ADBM
- PXF
- Коннекторы
- Chrony

Действия сервиса ADB

Назад к сервисам

Содержание

Create database
Create role
Enable auto core dump
Disable auto core dump
Enable mirroring
Install dbdiagnostic
Install MADlib
Install PostGIS
Reconfigure
Redistribute
Управление таблицами, схемами, схемами, представлениями
Запуск перенаправления
Обслуживание перенаправления
Мониторинг перенаправления
Run SQL
Restart
Stop
Manage tablespace

Create database

Выполняет `CREATE DATABASE` в кластере по умолчанию. [docs](#)

Если вы хотите использовать другой кластер, сначала создайте его с помощью `Create DB`. Затем его можно выбрать из списка, когда вы будете создавать новую базу данных.

Пароль по умолчанию для роли `postgres` отсутствует, но можно задать пароль, что сделает его доступным для входа.

Вы можете использовать `psql` для подключения к кластеру, например:

```
$ ssh -L 5432:localhost:5432 -N -f -i ~/.ssh/id_rsa dbadmin@<host_name_or_ip_address>

$ psql -h 127.0.0.1 -p 5432 -U postgres
Password:
psql (9.6.2)
SSL connection (protocol: TLSv1.2, compression: off)
Type "help" for help.

postgres=#
```

В каждом кластере доступны следующие таблицы `pg_*`:

```
postgres=# \dt pg*
List of relations
Schema | Name | Type | Owner
--------+----------------+-------+----------
pg_catalog | pg_aggregate | table | postgres
pg_catalog | pg_am | table | postgres
pg_catalog | pg_amop | table | postgres
pg_catalog | pg_amproc | table | postgres
pg_catalog | pg_attrdef | table | postgres
pg_catalog | pg_attribute | table | postgres
pg_catalog | pg_authid | table | postgres
pg_catalog | pg_auth_members | table | postgres
pg_catalog | pg_cast | table | postgres
pg_catalog | pg_class | table | postgres
pg_catalog | pg_constraint | table | postgres
pg_catalog | pg_conversion | table | postgres
pg_catalog | pg_database | table | postgres
pg_catalog | pg_default_acl | table | postgres
pg_catalog | pg_depend | table | postgres
pg_catalog | pg_description | table | postgres
pg_catalog | pg_enum | table | postgres
pg_catalog | pg_extension | table | postgres
pg_catalog | pg_foreign_data_wrapper | table | postgres
pg_catalog | pg_foreign_server | table | postgres
pg_catalog | pg_foreign_table | table | postgres
pg_catalog | pg_index | table | postgres
pg_catalog | pg_inherits | table | postgres
pg_catalog | pg_language | table | postgres
pg_catalog | pg_large_object | table | postgres
pg_catalog | pg_largeobject_metadata | table | postgres
pg_catalog | pg_namespace | table | postgres
pg_catalog | pg_opclass | table | postgres
pg_catalog | pg_operator | table | postgres
pg_catalog | pg_opfamily | table | postgres
pg_catalog | pg_pg_ts_dict | table | postgres
pg_catalog | pg_pg_ts_parser | table | postgres
pg_catalog | pg_pg_ts_template | table | postgres
pg_catalog | pg_policy | table | postgres
pg_catalog | pg_proc | table | postgres
pg_catalog | pg_rewrite | table | postgres
pg_catalog | pg_seclabel | table | postgres
pg_catalog | pg_shdepend | table | postgres
pg_catalog | pg_shdescription | table | postgres
pg_catalog | pg_shseclabel | table | postgres
pg_catalog | pg_statistic | table | postgres
pg_catalog | pg_tablespace | table | postgres
pg_catalog | pg_trigger | table | postgres
pg_catalog | pg_ts_config | table | postgres
pg_catalog | pg_ts_config_map | table | postgres
pg_catalog | pg_ts_dict | table | postgres
pg_catalog | pg_ts_parser | table | postgres
pg_catalog | pg_ts_template | table | postgres
pg_catalog | pg_type | table | postgres
pg_catalog | pg_user_mapping | table | postgres
(62 rows)
```

Create role

Добавляет новую роль в кластер.

Параметр	Описание	Допустимые значения
Role name	Уникальное имя новой роли	
Password	Пароль для новой роли	
Allow this role to log in	Если "y", предоставляет новой роли разрешение на вход в кластер. Если "n", запрещает новой роли вход в кластер.	y or n
Allow role to superuser	Если "y", предоставляет новой роли статус суперпользователя	Yes
Allow role to create roles	Если "y", предоставляет новой роли разрешение на создание ролей	Yes
Allow role to create databases	Если "y", предоставляет новой роли разрешение на создание баз данных	Yes
Allow role to create replication roles	Если "y", предоставляет новой роли разрешение на репликацию	Yes
Connection limit	Максимальное количество одновременных подключений, которые новая роль может использовать. Укажите -1, чтобы разрешить неограниченное количество подключений. Значение по умолчанию: -1	Integer greater than or equal to -1
Valid until	Дата и время, после которых пароль больше не будет действителен. Если оставить пустым, пароль будет действителен всегда.	Timestamp

Enable auto core dump

Выполняет `ALTER SYSTEM SET core_dump_enabled = 'on';` [docs](#)

Включает дампы ядра для всех сбоев в кластере.

Disable auto core dump

Выполняет `ALTER SYSTEM SET core_dump_enabled = 'off';` [docs](#)

Отключает дампы ядра для всех сбоев в кластере.

Enable mirroring

Включает для выбранного кластера `pg_mirror` и `pg_rewind`, а также `wal_log_hints = on`. [docs](#)

Если вы хотите использовать `pg_rewind` для синхронизации ведущего и резервного серверов после отказа, вам необходимо также включить `wal_log_hints`. Для получения дополнительной информации см. раздел «Настройка кластера для использования pg_rewind» в разделе «Высокая доступность, балансировка нагрузки и репликация» руководства по PostgreSQL.

Install dbdiagnostic

Устанавливает расширение `diagnostic` в кластер. [docs](#)

Чтобы выполнить диагностику, см. раздел `dbdiagnostic`.

Install MADlib

Устанавливает библиотеку `madlib`. [docs](#)

Чтобы выполнить статистический анализ, см. раздел `madlib`.

Install PostGIS

Устанавливает расширение `postgis`. [docs](#)

Чтобы выполнить геопространственные запросы, см. раздел `postgis`.

Reconfigure

Обновляет конфигурацию кластера в соответствии с `Cluster DB Config`. [docs](#)

Вы можете изменить конфигурацию кластера, выбрав `Cluster DB Config` и нажмите `Save` для сохранения изменений. Затем выполните `Reconfigure`, чтобы применить изменения.

- Изменения вступят в силу сразу, если изменяется только `Cluster DB Config`.
- Изменения вступят в силу после перезапуска кластера, если изменяется `Cluster DB Config` и `Cluster Instance Config`.

Redistribute

Для действия `Redistribute` выполняется `pg_rewind` для каждого экземпляра кластера, кроме ведущего, для синхронизации данных с ведущим сервером. Если `pg_rewind` завершается с ошибкой, выполняется `pg_basebackup`. Для получения дополнительной информации см. раздел «Высокая доступность, балансировка нагрузки и репликация» руководства по PostgreSQL.

Разнородная таблица, перед перераспределением

Таблицы с распределенными по хешам данными, которые не соответствуют политике `Hash`, не будут перераспределены. Вместо этого `Redistribute` выдаст предупреждение, перечисляющее такие таблицы. Вы можете:

- Изменить политику распределения таблицы, выполнив `ALTER TABLE <table_name> SET DISTRIBUTED BY (<column_name>);`, а затем повторно запустить `Redistribute`.
- Добавить опцию `Allow heterogeneous distribution` и повторно запустить `Redistribute`. Таблица будет перераспределена, но распределение останется неизменным.

Запуск перенаправления

Запуск перенаправления позволяет перенести с кластера для `Redistribute`. Перенаправление перенаправляет подключения к первичной базе данных на резервную, пока `Redistribute` выполняется.

- `Redistribute` выполняется на первичном сервере. Перенаправление перенаправляет подключения к резервной базе данных.
- `Redistribute` завершается на резервном сервере. `Redistribute` завершается, когда все подключения к резервной базе данных завершены.
- Перенаправление прекращается, когда `Redistribute` завершается на первичном сервере.
- Если вы выполняете `Redistribute` на кластере с одним экземпляром, перенаправление не выполняется.

Мониторинг перенаправления

Действия `Redistribute` и `Redirect` отображаются в `Recent Activity`.

- `Redistribute` — `redistribute`
- `Redirect` — `redirect`

Create table

Параметр	Описание
schema	Схема, в которой создается таблица. Если `public`, будет использована схема по умолчанию.
table	Имя таблицы. Должно соответствовать `[a-z_][a-z0-9_]*`.
column	Определение столбца. Например, `id bigint NOT NULL DEFAULT nextval('seq')`. Для нескольких столбцов разделяйте их запятыми.
partkey	Ключ секционирования. Например, `RANGE (id)`. Оставьте пустым, если не используется.
tablespace	Имя табличного пространства. Оставьте пустым, чтобы использовать значение по умолчанию.

Drop table

Параметр	Описание
schema	Схема, из которой удаляется таблица. Если `public`, будет использована схема по умолчанию.
table	Имя таблицы. Должно соответствовать `[a-z_][a-z0-9_]*`.
cascade	Если `true`, удаляет также все объекты, зависящие от таблицы (например, представления). Если `false`, ограничивается удалением только самой таблицы.

Create index

Параметр	Описание
schema	Схема таблицы, для которой создается индекс. Если `public`, будет использована схема по умолчанию.
table	Имя таблицы, для которой создается индекс. Должно соответствовать `[a-z_][a-z0-9_]*`.
column	Определение индекса. Например, `(lower(name))` или `(id, name)`. Для нескольких столбцов разделяйте их запятыми.
tablespace	Имя табличного пространства. Оставьте пустым, чтобы использовать значение по умолчанию.

Drop index

Параметр	Описание
schema	Схема, из которой удаляется индекс. Если `public`, будет использована схема по умолчанию.
index	Имя индекса. Должно соответствовать `[a-z_][a-z0-9_]*`.
concurrently	Если `true`, выполняется `DROP INDEX CONCURRENTLY`, что позволяет избежать блокировки таблицы во время удаления индекса.

Alter table

Параметр	Описание
schema	Схема, в которой находится таблица. Если `public`, будет использована схема по умолчанию.
table	Имя таблицы. Должно соответствовать `[a-z_][a-z0-9_]*`.
action	Выполняемое действие. Например, `ADD COLUMN name text` или `DROP COLUMN old_col`.

Create schema

Параметр	Описание
schema_name	Уникальное имя для новой схемы. Должно соответствовать `[a-z_][a-z0-9_]*`.
authorization	Необязательное имя роли, которая будет владеть новой схемой.
if_not_exists	Если `true`, не выдает ошибку, если схема уже существует.
schema_elements	Необязательный список объектов, которые будут созданы в новой схеме. Например, `CREATE TABLE t1 (id int); CREATE VIEW v1 AS SELECT * FROM t1;`.

Drop schema

Параметр	Описание
schema	Имя схемы, которую нужно удалить. Должно соответствовать `[a-z_][a-z0-9_]*`.
cascade	Если `true`, удаляет также все объекты, содержащиеся в схеме (таблицы, представления, функции и т. д.). Если `false`, ограничивается удалением только пустой схемы.

Create view

Параметр	Описание
schema	Схема, в которой будет создано представление. Если `public`, будет использована схема по умолчанию.
view	Имя представления. Должно соответствовать `[a-z_][a-z0-9_]*`.
select_statement	Определяющий запрос `SELECT`, который будет выполняться при каждом обращении к представлению. Например, `SELECT id, name FROM t1 WHERE active = true`.
materialized	Если `true`, создает материализованное представление, которое сохраняет результаты запроса на диске. Если `false`, создает обычное представление, которое выполняет запрос каждый раз.
with_data	Только для материализованных представлений. Если `true`, заполняет представление данными при создании. Если `false`, создает представление, но не заполняет его данными (требует последующего `REFRESH`).

Drop view

Параметр	Описание
schema	Схема, из которой удаляется представление. Если `public`, будет использована схема по умолчанию.
view	Имя представления, которое нужно удалить. Должно соответствовать `[a-z_][a-z0-9_]*`.
cascade	Если `true`, удаляет также все объекты, зависящие от представления (например, другие представления). Если `false`, ограничивается удалением только самого представления.

Refresh view

Параметр	Описание
schema	Схема, в которой находится материализованное представление. Если `public`, будет использована схема по умолчанию.
view	Имя материализованного представления, которое нужно обновить. Должно соответствовать `[a-z_][a-z0-9_]*`.
concurrently	Если `true`, выполняется `REFRESH CONCURRENTLY`, что позволяет избежать блокировки представления во время обновления. Требует наличия уникального индекса для представления.

Run SQL

Выполняет SQL-скрипт в кластере по умолчанию. [docs](#)

Если вы хотите использовать другой кластер, сначала создайте его с помощью `Create DB`. Затем его можно выбрать из списка, когда вы будете `Run SQL`.

Пароль по умолчанию для роли `postgres` отсутствует, но можно задать пароль, что сделает его доступным для входа.

Вы можете использовать `psql` для подключения к кластеру, например:

```
$ ssh -L 5432:localhost:5432 -N -f -i ~/.ssh/id_rsa dbadmin@<host_name_or_ip_address>

$ psql -h 127.0.0.1 -p 5432 -U postgres
Password:
psql (9.6.2)
SSL connection (protocol: TLSv1.2, compression: off)
Type "help" for help.

postgres=#
```

В каждом кластере доступны следующие таблицы `pg_*`:

```
postgres=# \dt pg*
List of relations
Schema | Name | Type | Owner
--------+----------------+-------+----------
pg_catalog | pg_aggregate | table | postgres
pg_catalog | pg_am | table | postgres
pg_catalog | pg_amop | table | postgres
pg_catalog | pg_amproc | table | postgres
pg_catalog | pg_attrdef | table | postgres
pg_catalog | pg_attribute | table | postgres
pg_catalog | pg_authid | table | postgres
pg_catalog | pg_auth_members | table | postgres
pg_catalog | pg_cast | table | postgres
pg_catalog | pg_class | table | postgres
pg_catalog | pg_constraint | table | postgres
pg_catalog | pg_conversion | table | postgres
pg_catalog | pg_database | table | postgres
pg_catalog | pg_default_acl | table | postgres
pg_catalog | pg_depend | table | postgres
pg_catalog | pg_description | table | postgres
pg_catalog | pg_enum | table | postgres
pg_catalog | pg_extension | table | postgres
pg_catalog | pg_foreign_data_wrapper | table | postgres
pg_catalog | pg_foreign_server | table | postgres
pg_catalog | pg_foreign_table | table | postgres
pg_catalog | pg_index | table | postgres
pg_catalog | pg_inherits | table | postgres
pg_catalog | pg_language | table | postgres
pg_catalog | pg_large_object | table | postgres
pg_catalog | pg_largeobject_metadata | table | postgres
pg_catalog | pg_namespace | table | postgres
pg_catalog | pg_opclass | table | postgres
pg_catalog | pg_operator | table | postgres
pg_catalog | pg_opfamily | table | postgres
pg_catalog | pg_pg_ts_dict | table | postgres
pg_catalog | pg_pg_ts_parser | table | postgres
pg_catalog | pg_pg_ts_template | table | postgres
pg_catalog | pg_policy | table | postgres
pg_catalog | pg_proc | table | postgres
pg_catalog | pg_rewrite | table | postgres
pg_catalog | pg_seclabel | table | postgres
pg_catalog | pg_shdepend | table | postgres
pg_catalog | pg_shdescription | table | postgres
pg_catalog | pg_shseclabel | table | postgres
pg_catalog | pg_statistic | table | postgres
pg_catalog | pg_tablespace | table | postgres
pg_catalog | pg_trigger | table | postgres
pg_catalog | pg_ts_config | table | postgres
pg_catalog | pg_ts_config_map | table | postgres
pg_catalog | pg_ts_dict | table | postgres
pg_catalog | pg_ts_parser | table | postgres
pg_catalog | pg_ts_template | table | postgres
pg_catalog | pg_type | table | postgres
pg_catalog | pg_user_mapping | table | postgres
(62 rows)
```

Restart ADB

Перезапускает кластер ADB.

Вы можете использовать `Cluster DB` для перезапуска кластера по умолчанию или выбрать другой кластер из списка.

- **Graceful restart** — `pg_ctl -m smart restart`. Ожидает завершения всех подключений перед перезапуском.
- **Immediate restart** — `pg_ctl -m immediate restart`. Немедленно прерывает все подключения.

Stop

Останавливает кластер ADB. Если вы выбрали `Graceful restart` в предыдущем шаге, выполняется `pg_ctl -m smart stop`.

Manage tablespace

Управляет табличными пространствами. [docs](#)

Параметр	Описание	Допустимые значения
Name	Уникальное имя для нового табличного пространства. Должно соответствовать `[a-z_][a-z0-9_]*`.	pg_default
Path	Абсолютный путь к каталогу на сервере, где будет храниться табличное пространство. Каталог должен существовать и принадлежать пользователю `postgres`.	/pg_tblspc/
Use only this node, not all cluster nodes	Если `true`, выполняется только на ведущем сервере. Если `false`, выполняется на всех узлах кластера.	false

```
$ psql -h 127.0.0.1 -p 5432 -U postgres
Password:
psql (9.6.2)
SSL connection (protocol: TLSv1.2, compression: off)
Type "help" for help.

postgres=# \db+
List of tablespaces
Name | Owner | Location | Access privileges | Options | Size | Description
------------+----------+----------+-------------------+---------+---------+-------------
pg_default | postgres | | | | 29 MB |
pg_global | postgres | | | | 573 kB |
(2 rows)

postgres=#
```

Пароль по умолчанию для роли `postgres` отсутствует, но можно задать пароль, что сделает его доступным для входа.

Вы можете использовать `psql` для подключения к кластеру, например:

```
$ ssh -L 5432:localhost:5432 -N -f -i ~/.ssh/id_rsa dbadmin@<host_name_or_ip_address>

$ psql -h 127.0.0.1 -p 5432 -U postgres
Password:
psql (9.6.2)
SSL connection (protocol: TLSv1.2, compression: off)
Type "help" for help.

postgres=#
```

Действия сервиса ADB Control

Антон Васильев

Содержание

Назад к содержанию

Install

Reinstall

Reconfigure & Restart

Start

Stop

Uninstall

В этой статье описаны действия, доступные для сервиса ADB Control в веб-интерфейсе ADCM.

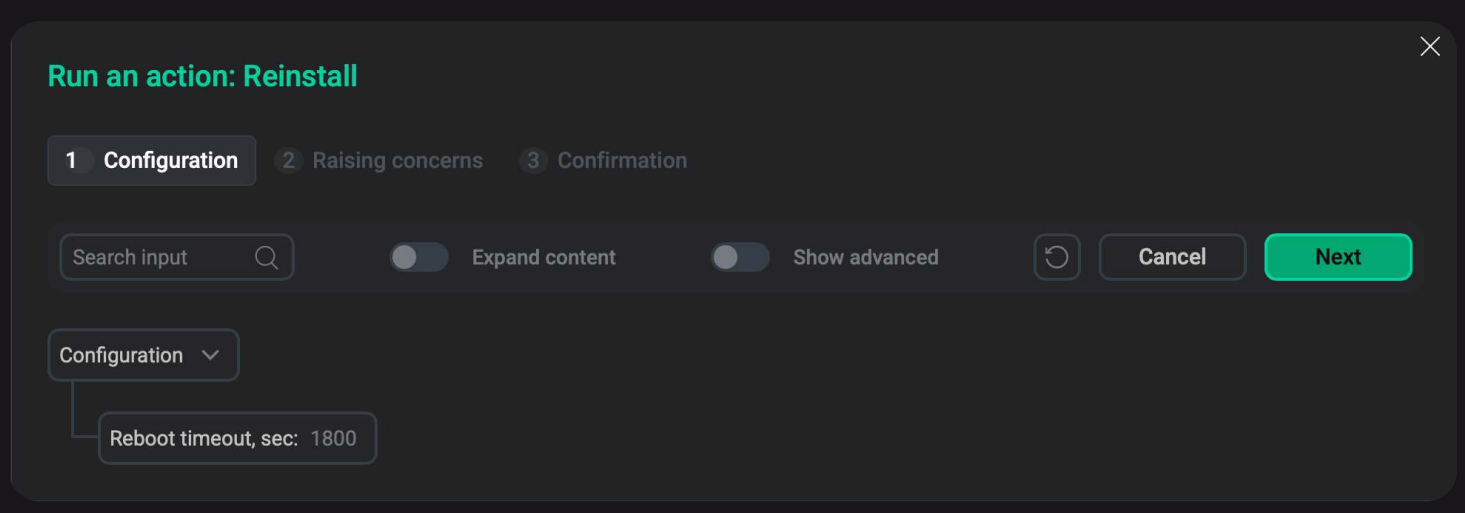
Install

Устанавливает сервис ADB Control. Действие доступно, если сервис добавлен, но пока не установлен.

Reinstall

Переустанавливает сервис ADB Control.

На вкладке ***Configuration*** открывшегося окна в поле ***Reboot timeout, sec*** введите время ожидания перезагрузки хостов (в секундах). Значение по умолчанию — `1800`. Значение не может быть меньше `600`.



Окно Reinstall

Reconfigure & Restart

Обновляет конфигурацию сервиса в соответствии с параметрами конфигурации и перезапустить сервис.

На вкладке ***Configuration*** открывшегося окна с помощью флага ***Unlock ADB Control db*** выберите, что делать с блокировками базы данных, которые могут возникнуть в случае неудачной миграции. Если значение флага — `true`, ADB Control снимет блокировки и сделает свою БД доступной для миграций. Значение по умолчанию — `false`: блокировки сохраняются, чтобы предотвратить потенциально несовместимые миграции.

Start

Запускает сервис ADB Control.

Stop

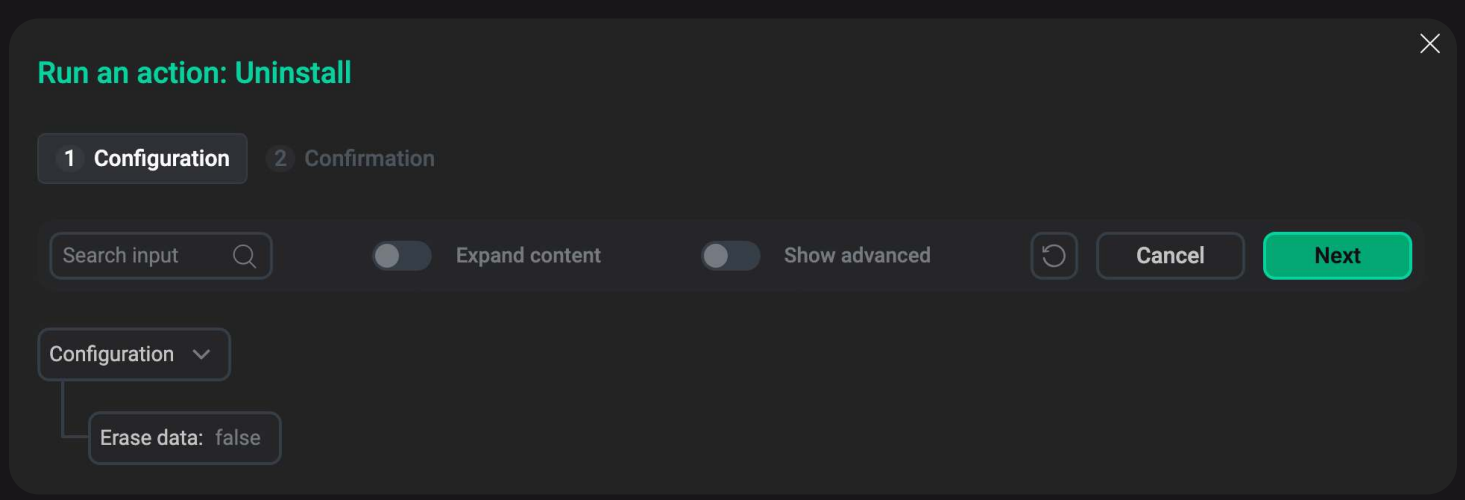
Останавливает сервис ADB Control.

Uninstall

Удаляет сервис ADB Control. На вкладке ***Configuration*** открывшегося окна с помощью флага ***Erase data*** выберите, требуется ли удалять данные, связанные с сервисом.

ПРИМЕЧАНИЕ

Сервис ADBM зависит от ADB Control. Если требуется удалить ADB Control, то сначала нужно удалить ADBM.



Окно Uninstall

Действия сервиса ADBM

Антон Васильев

Содержание

Назад к содержанию

Install

Reinstall

Reconfigure & Restart

Start

Stop

Uninstall

В этой статье описаны действия, доступные для сервиса Arenadata DB Backup Manager (ADBM) в веб-интерфейсе ADCM.

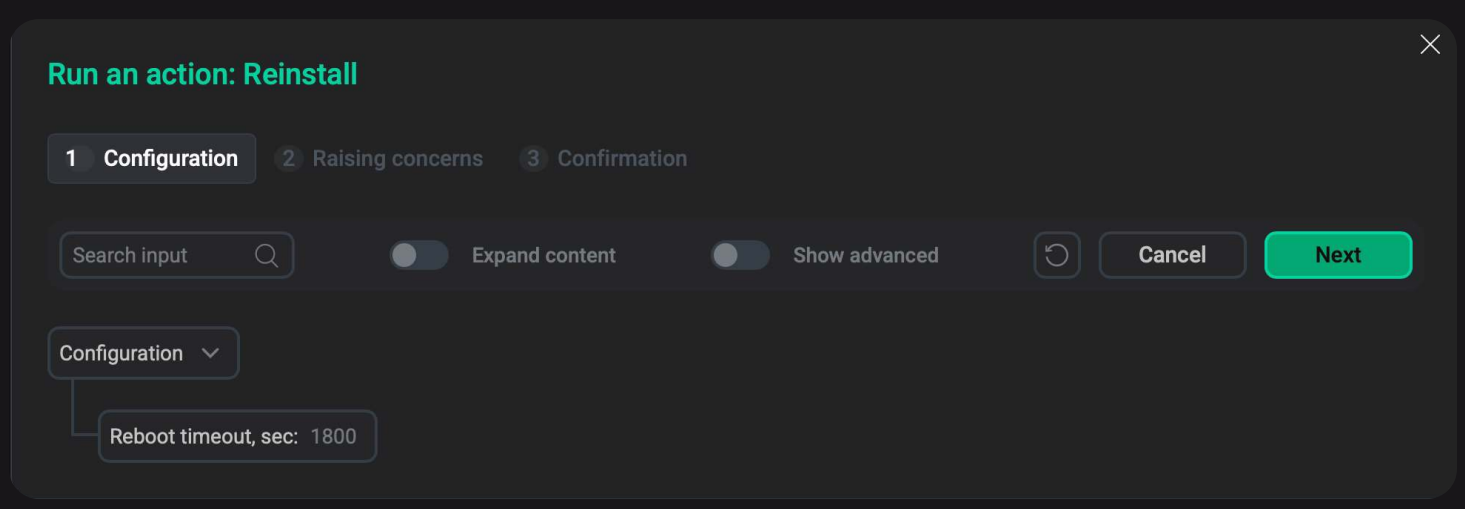
Install

Устанавливает сервис ADBM. Действие доступно, если сервис добавлен, но пока не установлен.

Reinstall

Переустанавливает сервис ADBM.

На вкладке ***Configuration*** открывшегося окна в поле ***Reboot timeout, sec*** введите время ожидания перезагрузки хостов (в секундах). Значение по умолчанию — `1800`. Значение не может быть меньше `600`.



Окно Reinstall

Reconfigure & Restart

Обновляет конфигурацию сервиса в соответствии с параметрами конфигурации и перезапустить сервис.

На вкладке ***Configuration*** открывшегося окна с помощью флага ***Unlock adbm db*** выберите, что делать с блокировками базы данных, которые могут возникнуть в случае неудачной миграции. Если значение флага — `true`, ADBM снимет блокировки и сделает свою БД доступной для миграций. Значение по умолчанию — `false`: блокировки сохраняются, чтобы предотвратить потенциально несовместимые миграции.

Start

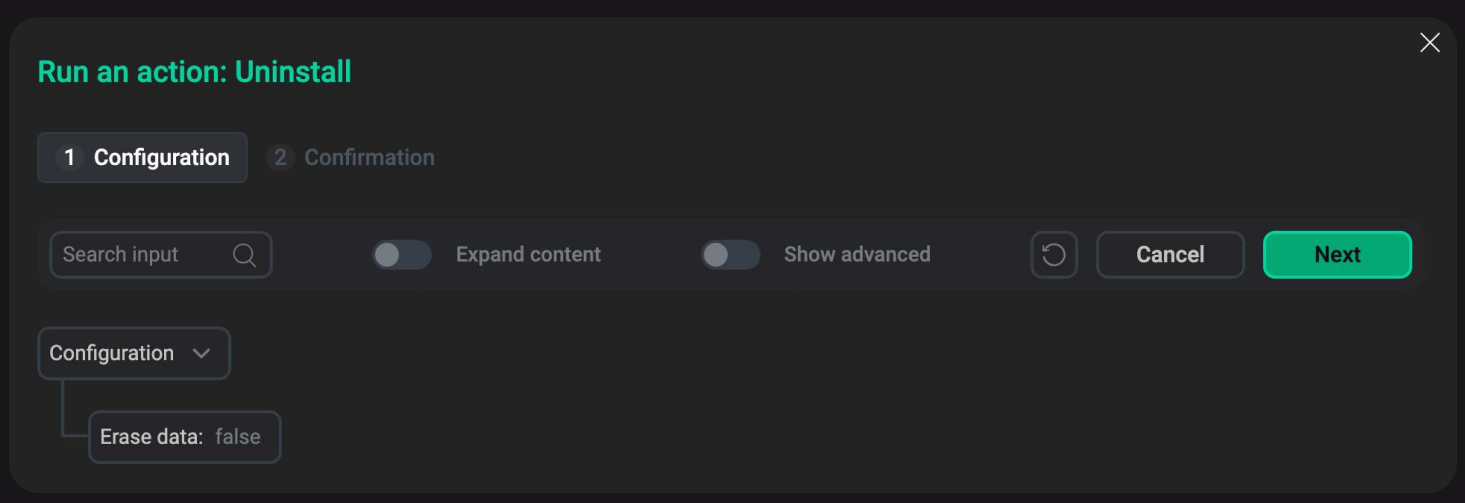
Запускает сервис ADBM.

Stop

Останавливает сервис ADBM.

Uninstall

Удаляет сервис ADBM. На вкладке ***Configuration*** открывшегося окна с помощью флага ***Erase data*** выберите, требуется ли удалять данные, связанные с сервисом.



Окно Uninstall

Действия сервиса PXF

Антон Васильев

Содержание

Назад к содержанию

Install

Reconfigure

Restart

Start

Stop

Uninstall

В этой статье описаны действия, доступные для сервиса PXF в веб-интерфейсе ADCM.

Install

Устанавливает сервис PXF. Действие доступно, если сервис добавлен, но пока не установлен.

Reconfigure

Обновляет конфигурацию сервиса в соответствии с параметрами конфигурации.

Restart

Перезапускает сервис PXF.

Start

Запускает сервис PXF.

Stop

Останавливает сервис PXF.

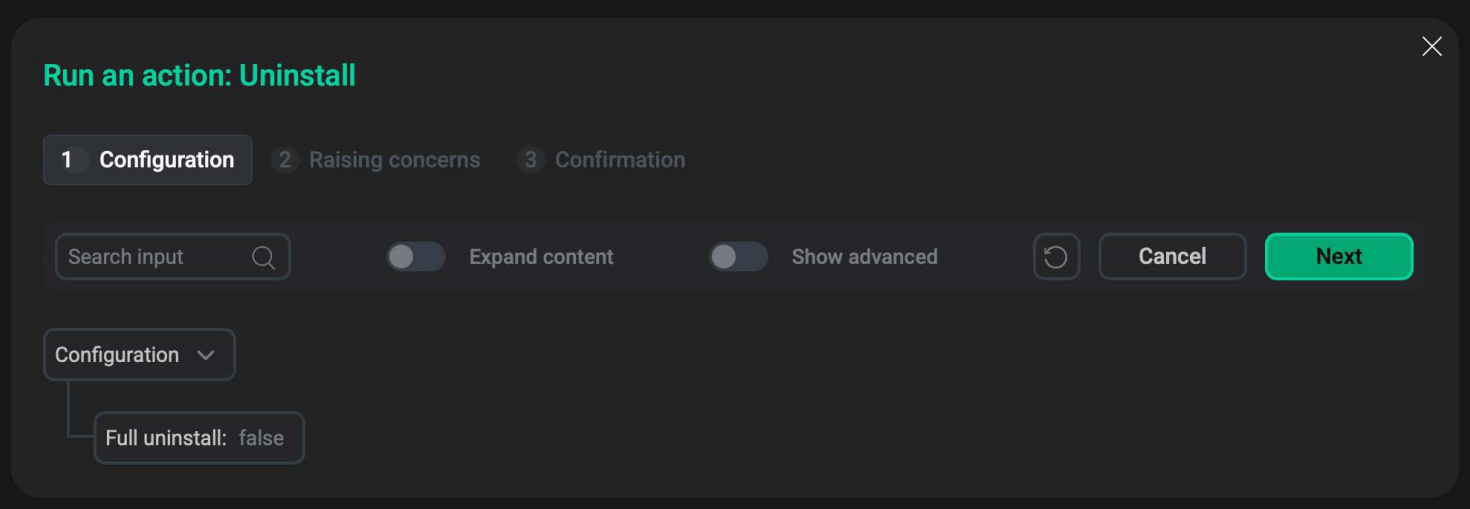
Uninstall

Удаляет сервис PXF.

ПРИМЕЧАНИЕ

Коннекторы ADB to Kafka и ADB ClickHouse зависят от сервиса PXF, поэтому удалить PXF нельзя, если вы используете эти коннекторы.

На вкладке ***Configuration*** открывшегося окна установите значение `true` для флага ***Full uninstall***, если требуется также удалить данные, связанные с PXF: конфигурационные файлы, пользователя `pxf`, расширения `pxf` и `pxf_fdw` в БД и объекты, зависящие от этих расширений. Значение по умолчанию — `false`.



Окно PXF → Uninstall

Действия сервисов-коннекторов

Антон Васильев

Содержание

Назад к содержанию

Install

Reinstall

Uninstall

Действия, описанные ниже, доступны для следующих коннекторов:

- ADB ClickHouse Connector
- ADB to Kafka Connector
- Kafka to ADB Connector
- ADB to ADB Connector

Install

Устанавливает сервис коннектора. Действие доступно, если сервис добавлен, но пока не установлен.

Reinstall

Переустанавливает сервис коннектора.

Uninstall

Удаляет сервис коннектора.

Действия сервиса Chrony

Антон Васильев

Содержание

Назад к содержанию

Install

Add/remove chronys component

Reinstall

Uninstall

Сервис Chrony синхронизирует системное время в кластере. Сегментные хосты синхронизируют время с хостом координатора. В этой статье описаны действия, доступные для сервиса Chrony в веб-интерфейсе ADCM.

Install

Устанавливает сервис Chrony. Действие доступно, если сервис добавлен, но пока не установлен.

После установки можно проверить источники времени, используемые на хосте, выполнив команду:

```
$ chronyc sources
```

Например, если выполнить эту команду на сегментном хосте, то в выводе будет показано, что координатор-хост используется для синхронизации времени, а standby-координатор используется в качестве резервного источника:

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* mdw                      10   9   377   411    -82us[  -98us] +/-  271us
^- smdw                    11   9   377    76   -606us[ -606us] +/- 1326us
```

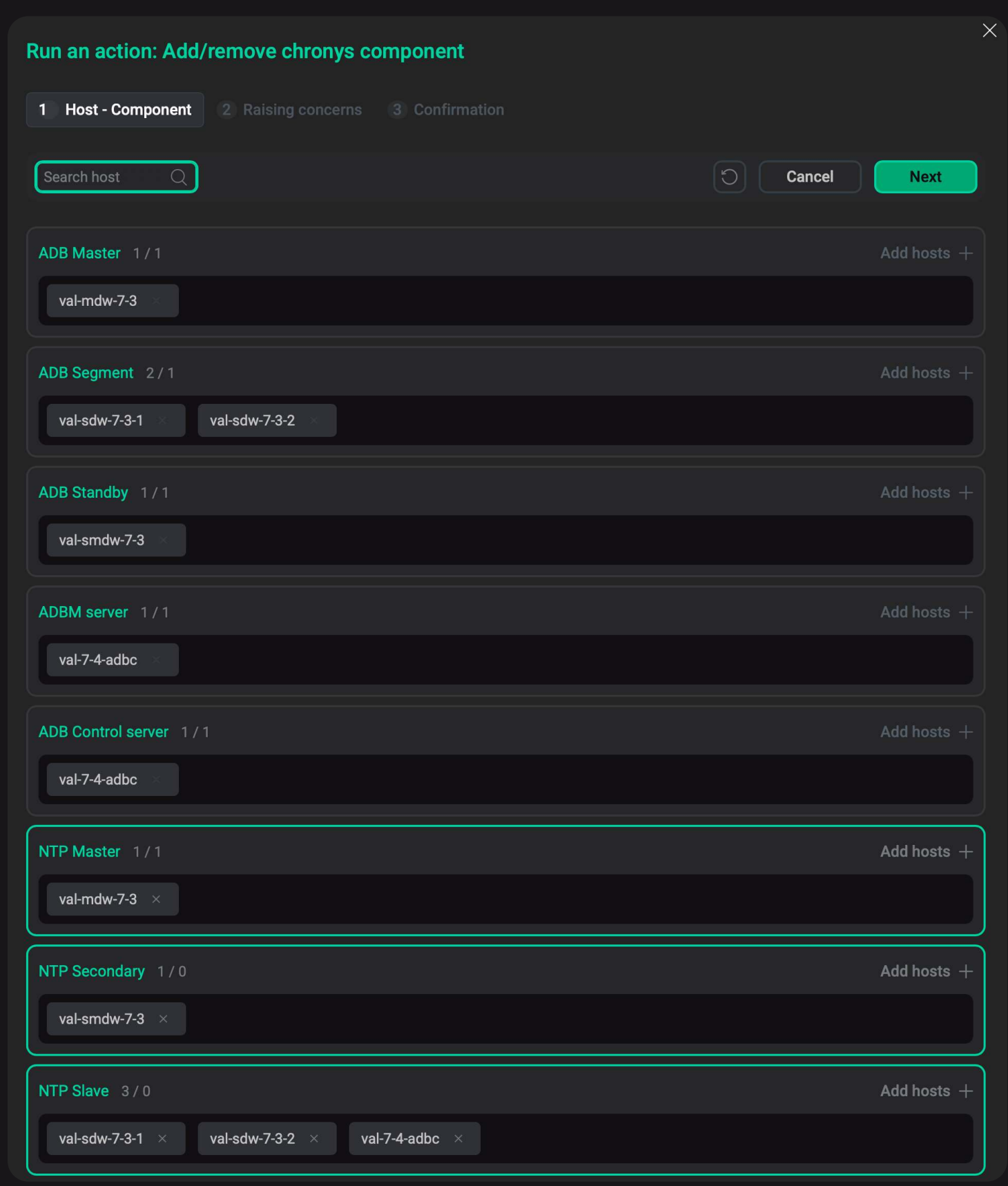
Add/remove chronys component

Позволяет перераспределить компоненты Chrony по хостам: установить Chrony на хост или удалить.

В открывшемся окне нажмите ***Add hosts*** в секции ***NTP Master***, ***NTP Secondary*** или ***NTP Slave*** и выберите хосты, на которых должен быть установлен Chrony.

ПРИМЕЧАНИЕ

Если вы используете Chrony, необходимо установить его на всех хостах кластера.



Окно Add/remove chronys component

Reinstall

Переустанавливает сервис Chrony.

Uninstall

Удаляет сервис Chrony.

Настройка СУБД с помощью GUC

Алексей Лесовский

Содержание

- Обзор
- Установка параметров
  - session and local scope
  - coordinator and local
  - restart and reload
  - boolean
  - enum
  - read only
- Чтение параметров
- Примеры

Перейти к содержанию

Обзор

Система конфигурирования Citus (также называемая **Grand Unified Configuration, GUC**) позволяет адаптировать кластер к различным сценариям нагрузки и конфигурациям aws. Вначале из citus/src/backend/shared/shared_library_init.c:

```
/* define Citus GUC variables and validate them */
RegisterCitusConfigVariables();
```

Сначала выполняется вызов функции RegisterCitusConfig:

```
/* register Citus configuration variables */
RegisterCitusConfig(DistributedTaskQueueHashEntry.gucEntries,
                    DistributedTaskQueueHashEntry.gucStrings,
                    MaxDistributedTaskQueueGucEntries,
                    CitusAllocate, CitusReallocate, CitusFree, ereport, vsnprintf);
```

Citus переопределяет стандартные функции из postgres для использования собственных версий:

```
/* Citus uses palloc() and friends, but GUC uses plain pointers */
#define MaxGucEntries 256
#define MaxGucStrings 256
#define CitusAllocate(size) ((void *) MemoryContextAlloc(TopMemoryContext, (size)))
#define CitusReallocate(ptr, size) ((void *) repalloc((ptr), (size)))
#define CitusFree(ptr) pfree(ptr)
```

Если переменная, что также бы выглядела для citus, начинается с prefix citus_ или citus.:

```
/* Check if the GUC name starts with "citus." or "citus_" */
bool
IsCitusGucVariable(const char *name)
{
    int prefixLen = strlen("citus.");
    int citusPrefixLen = strlen("citus_");

    if (strncmp(name, "citus.", prefixLen) == 0)
        return true;
    else if (strncmp(name, "citus_", citusPrefixLen) == 0)
        return true;
    else
        return false;
}
```

Переменные можно, что также бы выглядели для citus, объявлять как переменные типа int, enum, bool, или string:

```
/* Check if the GUC name is a Citus variable */
bool
IsCitusGucVariable(const char *name)
{
    int prefixLen = strlen("citus.");
    int citusPrefixLen = strlen("citus_");

    if (strncmp(name, "citus.", prefixLen) == 0)
        return true;
    else if (strncmp(name, "citus_", citusPrefixLen) == 0)
        return true;
    else
        return false;
}
```

Переменные можно, что также бы выглядели для citus, создавать как:

- bool — true/false;
- int — целое;
- enum — строка;
- bool — true/false;
- string — строка;
- real — real;

Для каждого типа переменных, создаваемых здесь или citus.header, необходимо поместить значение в переменную memoryvar. И не использовать, пробовать напрямую использовать переменные. Вместо этого использовать функции:

```
/* Define a variable, using the appropriate GUC function */
extern void DefineCustomBoolVariable(const char *name,
                                     const char *shortDesc,
                                     const char *longDesc,
                                     bool *valueAddr,
                                     bool bootValue,
                                     int flags,
                                     void (*hook_assign)(bool newvalue, void *extra),
                                     void (*hook_show)(void));
extern void DefineCustomIntVariable(const char *name,
                                    const char *shortDesc,
                                    const char *longDesc,
                                    int *valueAddr,
                                    int bootValue,
                                    int minValue,
                                    int maxValue,
                                    int flags,
                                    void (*hook_assign)(int newvalue, void *extra),
                                    void (*hook_show)(void));
extern void DefineCustomEnumVariable(const char *name,
                                     const char *shortDesc,
                                     const char *longDesc,
                                     int *valueAddr,
                                     int bootValue,
                                     const struct config_enum_entry *const *const options,
                                     int flags,
                                     void (*hook_assign)(int newvalue, void *extra),
                                     void (*hook_show)(void));
extern void DefineCustomStringVariable(const char *name,
                                       const char *shortDesc,
                                       const char *longDesc,
                                       char *const *valueAddr,
                                       char *bootValue,
                                       int flags,
                                       void (*hook_assign)(char *newvalue, void *extra),
                                       void (*hook_show)(void));
extern void DefineCustomRealVariable(const char *name,
                                     const char *shortDesc,
                                     const char *longDesc,
                                     double *valueAddr,
                                     double bootValue,
                                     int flags,
                                     void (*hook_assign)(double newvalue, void *extra),
                                     void (*hook_show)(void));
```

Установка параметров

Каждый отдельно взятый параметр принимает в зависимости от своей настройки (см. citus.h):

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

session and system

Это стандартный сценарий для postgres, citus.header переменные (и citus.h):

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Это стандартный сценарий для postgres, citus.header переменные (и citus.h):

- **session** — переменная имеет значение для всех сессий;
- **system** — переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Если переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

coordinator and local

Это специфичные переменные только для координатора, значения которых распространяются на шардах, обновлены, пулы подключения к координатору и шардам, и т.д.:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Это специфичные переменные только для координатора, которые распространяются на шардах, обновлены, пулы подключения к координатору и шардам, и т.д.:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

restart and reload

Это стандартные переменные, которые, в отличие от citus.header и citus.h, не распространяются на шарды, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Если переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

boolean

Для переменных с типом bool citus.header или citus.h проверяются корректность и допустимые значения, например, true, false, или может принимать настройки.

Если переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Если переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

enum

Для переменных с типом enum citus.header или citus.h проверяются корректность и допустимые значения, например, true, false, или может принимать настройки.

Если переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Если переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

read only

Переменные, которые нельзя корректно для citus.header переменных. Обычно только пользователям значения или параметров корректно:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Если переменная имеет значение для всех сессий, сервера, кластера, шардов, репликации и всего citus. Если в postgresql.conf:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтение параметров

Для каждого типа переменных citus.header или citus.h предусмотрены функции для чтения значения:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

- Чтобы получить значение переменной, принимающей true/false, нужно использовать citus_getbool(), например:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

- Чтобы получить значение переменной, принимающей int, нужно использовать citus_getint(), например:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

- Чтобы получить значение переменной, принимающей enum, нужно использовать citus_getenum(), например:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

- Чтобы получить значение переменной, принимающей string, нужно использовать citus_getstring(), например:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

- Чтобы получить значение переменной, принимающей real, нужно использовать citus_getreal(), например:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Примеры

Пример переменной (на уровне сессии)

Новая переменная (пример) значения переменной citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Пример переменной (на уровне кластера)

Переменная значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Переменная значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Пример переменной (на уровне координатора)

Переменная значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Переменная значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Пример переменной (read only)

Переменная значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Переменная значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы сделать foo переменной значения citus.foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Пример переменной (boolean)

Новая переменная (пример) переменная переменная bool_enamble_foo:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения true/false:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения true/false:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения true/false:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения true/false:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения true/false:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Пример переменной (enum)

Новая переменная (пример) переменная переменная foo_enumerator. Если переменная имеет следующие значения: foo0, foo1, foo2, foo3:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo0:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo1:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo2:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo3:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Пример переменной (string)

Новая переменная (пример) переменная переменная foo_string. Если переменная имеет следующие значения: foo0, foo1, foo2, foo3:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo0:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo1:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo2:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo3:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Пример переменной (real)

Новая переменная (пример) переменная переменная foo_real. Если переменная имеет следующие значения: foo1, foo2, foo3:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo1:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo2:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Чтобы переменная foo принимала значения foo3:

```
/* Flag bits for Citus GUC variables */
#define GUC_STANDARD (0)
#define GUC_COORDINATOR (1 << 0)
#define GUC_NO_SHOW_ALL (1 << 1)
#define GUC_NO_RESET (1 << 2)
#define GUC_MANAGED (1 << 3)
```

Использование плагина ADB DDBoost

Д. Дмитр Барышников

Содержание

Назад к содержанию

- Обзор
- Установка и настройка
- Использование
  - gpbackup
  - gprestore

Обзор

ADB DDBoost — это плагин-коннектор, обеспечивающий нативное использование системы хранения данных Dell EMC Data Domain с утилитами Greenplum для логического резервного копирования данных (gpbackup) и их восстановления (gprestore).

При совместном использовании плагина `gpbackup` и `gprestore` устанавливается подключение к серверу резервного копирования Data Domain Platform. Для интеграции с Data Domain Platform плагину требуется библиотека DDOBoost.so версии 7.7, которая поставляется в составе пакета Dell DDBoostSDK 7.7. Эта библиотека является проприетарной и должна быть получена через Dell Support или предоставлена Dell Data Domain в вашей стране. За счет дедупликации данных, предоставляемой библиотекой, DDBoost плагин позволяет существенно ускорить формирование бэкапов и сэкономить место для хранения данных.

> **ПРИМЕЧАНИЕ**
>
> - Подробную информацию об архитектуре плагина ADB DDBoost, его преимуществах и результатах нагрузочного тестирования можно получить в ответе технического отдела продаж Arenadata и заказчиков могут на порядок использовать ADB вместо и ПО Dell EMC Data Domain.
> - Плагин DDBoost доступен в Enterprise-версии ADB.

Установка и настройка

Для установки плагина ADB DDBoost выполните следующие шаги:

1. Установите пакет adb-ddp.plugin на **все хосты кластера**, выполнив вход под пользователем с правами `sudo`. Ниже приведен пример установки через пакетный менеджер APT:

```
$ sudo apt install adb-ddp-plugin
```

В результате успешного выполнения команды создается исполняемый файл плагина $GPHOME/bin/adb_ddp_plugin.

2. Перенесите библиотеку libDDBoost.so в директорию lib на **все хосты кластера**.
3. Подключитесь к **координатор-кластер** ADB под пользователем `gpadmin`:

```
$ sudo su - gpadmin
```

4. На **координатор-кластер** создайте конфигурационный YAML-файл в домашней директории пользователя /home/gpadmin (например, с именем adb_ddp_plugin.yaml):

```
$ touch /home/gpadmin/adb_ddp_plugin.yaml
```

Пример файла и его описания приведены ниже:

```
executablepath: /usr/lib/gpdb/bin/adb_ddp_plugin
options:
  hostname: "testhost"
  username: "testuser"
  password: "YXXXXXXXXXX"
  storage_unit: "testunit"
  directory: "/testdir"
  write_buffer_size: "1048576"
  read_buffer_size: "1048576"
  log_level: "DEBUG"
  log_path: "/home/gpadmin"
```

Поля конфигурационного файла плагина ADB DDBoost

Имя	Описание	Значение по умолчанию	Обязательность
executablepath	Абсолютный путь к плагину в файловой системе хостов ADB (вместе с именем исполняемого файла плагина). Плагин должен быть размещен на всех хостах кластера в одном и том же каталоге. По умолчанию плагин устанавливается по следующему пути: $GPHOME/bin/adb_ddp_plugin	–	Да
hostname	IP-адрес или имя хоста, обеспечивающего работу с DDBoost. Может содержать не более 30 символов	–	Да
username	Имя пользователя, которому выданы права для работы с DDBoost. Пользователь конфигурируется на стороне DDBoost, это не пользователь операционной системы или ADB. Может содержать не более 30 символов	–	Да
password	Пароль пользователя, которому выданы права для работы с DDBoost	–	Да
storage_unit	Наименование storage unit, конфигурируемого на стороне DDBoost	–	Да
directory	Директория в файловой системе DDBoost, в которую будут сохранятся все файлы, создаваемые в рамках резервного копирования БД с помощью утилиты `gpbackup`. При создании каждого бэкапа плагин автоматически создает все вложенные каталоги в /<storage_unit>/<directory>, соответствующие дате и времени создания бэкапа: /<storage_unit>/<directory>/YYYYMMDD/YYYYMMDD/HHmmSS/	–	Да
write_buffer_size	Размер буфера записи данных в DDBoost (в байтах). Допустимый диапазон значений: 64 <= write_buffer_size <= 1048576	–	Да
read_buffer_size	Размер буфера чтения данных из DDBoost (в байтах). Допустимый диапазон значений: 64 <= read_buffer_size <= 1048576	–	Да
log_level	Уровень логирования. Возможные значения: - DEBUG - INFO - NOTICE - WARN - ERROR - FATAL	WARN	Нет
log_path	Абсолютный путь к директории для записи логов плагина. В указанной директории будет создан файл adb_ddp_plugin.log	/home/gpadmin/gpAdminLogs	Нет

Использование

gpbackup

Для использования плагина ADB DDBoost при создании бэкапов с помощью утилиты `gpbackup` необходимо указать параметр `--plugin-config <yaml_path>`, где `<yaml_path>` — это абсолютный путь к настроечному файлу plugin.yaml на стороне координатора ADB, плагина в формате YAML.

Ниже приведен пример резервного копирования одной таблицы public.test с помощью плагина `adb`:

1. Подключитесь к базе данных `adb` на координатор-ноде кластера ADB под пользователем `gpadmin` (например, через `psql`). Убедитесь с помощью `psql` команды `\dt`, что искомая таблица (в приведенном примере `public.test`) существует в базе данных:

```
\dt
```

Результат:

```
                   List of relations
 Schema |    Name    | Type |  Owner  |     Storage
--------+------------+------+---------+-------------
 public | spatial_ref_sys | table | gpadmin | heap
 public | test       | table | gpadmin | heap
 public | test2      | table | gpadmin | ao_column
 public | test3      | table | gpadmin | ao_column
(4 rows)
```

2. Отключитесь от базы данных `adb` и запустите утилиту `gpbackup` под пользователем `gpadmin`, заполнив параметр `--plugin-config`. Обратите внимание, что при помощи аргумента `--include-table` запрошен бэкап одной таблицы:

```
$ gpbackup --dbname adb --no-compression --single-data-file --include-table public.test --plugin-config
/home/gpadmin/adb_ddp_plugin.yml
```

> **РЕКОМЕНДАЦИЯ**
>
> При использовании DDBoost плагина вместе с утилитой `gpbackup` рекомендуется выбирать режим копирования "все таблицы сегмента — один файл" (`--single-data-file`). Режим параллельного копирования (`--jobs`) не следует использовать из-за крайне низкой производительности записи.

В выводе команды идентифицируйте временную метку бэкапа, которая будет использоваться в качестве названия директории бэкапа и понадобится при восстановлении данных:

```
[INFO]:-Backup Timestamp = 20251112184114
```

В случае успешного создания бэкапа вывод команды завершается сообщением:

```
[INFO]:-Backup completed successfully
```

3. Убедитесь, что необходимые файлы сформированы и сохранены на выделенном сервере Data Domain Platform. Для этого откройте его директорию /<storage_unit>/<directory>/YYYYMMDD/YYYYMMDD/HHmmSS/, где:

- `<storage_unit>` — `<directory>` — значения одноименных параметров из YAML-файла с конфигурацией плагина;
- `YYYYMMDD` — дата создания бэкапа;
- `YYYYMMDD/HHmmSS` — дата и время создания бэкапа (временная метка, которая будет использоваться в качестве аргумента `--timestamp` при восстановлении данных из текущего бэкапа).

```
$ ls /storage_unit/<directory>/20251112/20251112184114/
```

Содержимое директории в приведенном примере выглядит следующим образом:

```
gpbackup_0_20251112184114.toc                    gpbackup_3_20251112184114.toc
gpbackup_0_20251112184114.toc.yaml               gpbackup_3_20251112184114.toc.yaml
gpbackup_1_20251112184114                        gpbackup_4_20251112184114
gpbackup_1_20251112184114.toc.yaml               gpbackup_4_20251112184114.toc.yaml
gpbackup_20251112184114_config.yaml              gpbackup_5_20251112184114
gpbackup_20251112184114_metadata.sql             gpbackup_5_20251112184114.toc.yaml
gpbackup_20251112184114_plugin_config.yaml       gpbackup_6_20251112184114
gpbackup_20251112184114_report                   gpbackup_6_20251112184114.toc.yaml
gpbackup_20251112184114.toc.yaml                 gpbackup_7_20251112184114
gpbackup_2_20251112184114                        gpbackup_7_20251112184114.toc.yaml
gpbackup_2_20251112184114.toc.yaml
```

На координатор-ноде в директории $COORDINATOR_DATA_DIRECTORY/backups/YYYYMMDD/YYYYMMDD/HHmmSS доступен отчет о выполненном бэкапе с информацией о количестве объектов в резервной копии:

```
$ cat /data/master/gpseg-1/backups/20251112/20251112184114/gpbackup_20251112184114_report
```

Пример отчета об успешном бэкапе одной таблицы

```
Greenplum Database Backup Report

timestamp key:       20251112184114
gpdb version:        7.3.0+dev.8x5.g70dd09efef build 25dev(1704dh9a
gpbackup version:    1.30.4

database name:       adb
command line:        gpbackup --dbname adb --no-compression --single-data-file --include-table public.test --plugin-config
/home/gpadmin/adb_ddp_plugin.yaml
compression:         None
plugin executable:   adb_ddp_plugin
backup section:      All Sections
object filtering:    Include Table Filter
includes statistics: No
data file format:    Single Data File Per Segment
incremental:         False

start time:          Wed Nov 12 2025 16:01:14
end time:            Wed Nov 12 2025 16:01:52
duration:            0:00:38

backup status:       Success

segment count:       8

count of database objects in backup:
constraints          0
conversions          0
default privileges   0
event triggers       0
indexes              0
policies             0
rules                0
statistics_ext       0
sequences            0
tables               1
triggers             0
views                0
```

gprestore

Для использования плагина ADB DDBoost при восстановлении данных из бэкапов с помощью утилиты `gprestore` необходимо указать параметр `--plugin-config <yaml_path>`, где `<yaml_path>` — это абсолютный путь к настроечному файлу plugin.yaml на стороне координатора ADB, плагина в формате YAML.

Ниже приведен пример восстановления таблицы `public.test` для которой ранее был создан бэкап:

1. Подключитесь к базе данных `adb` на координатор-ноде кластера ADB под пользователем `gpadmin` (например, через `psql`). Удалите таблицу `public.test`:

```
DROP TABLE test;
```

2. Убедитесь, что таблица удалена, с помощью `psql` команды `\dt`:

```
\dt
```

Результат:

```
                   List of relations
 Schema |    Name    | Type |  Owner  |     Storage
--------+------------+------+---------+-------------
 public | spatial_ref_sys | table | gpadmin | heap
 public | test       | table | gpadmin | heap
 public | test2      | table | gpadmin | ao_column
 public | test3      | table | gpadmin | ao_column
(4 rows)
```

3. Отключитесь от базы данных `adb` и запустите утилиту `gprestore` под пользователем `gpadmin`, заполнив параметр `--plugin-config`. Обратите внимание, что в аргументе `--timestamp` необходимо указать временную метку создания бэкапа в формате YYYYMMDDHHmmSS:

```
$ gprestore --timestamp 20251112184114 --plugin-config /home/gpadmin/adb_ddp_plugin.yaml --include-table public.test
```

В случае успешного восстановления данных из бэкапа выводу команды завершается сообщением:

```
[INFO]:-Restore completed successfully
```

4. Вновь подключитесь к базе данных `adb` и убедитесь, что таблица `public.test` доступна:

```
\dt
```

Результат:

```
                   List of relations
 Schema |    Name    | Type |  Owner  |     Storage
--------+------------+------+---------+-------------
 public | spatial_ref_sys | table | gpadmin | heap
 public | test       | table | gpadmin | heap
 public | test2      | table | gpadmin | ao_column
 public | test3      | table | gpadmin | ao_column
(4 rows)
```

Инструментарий arenadata_toolkit

Содержание

- Таблицы схемы arenadata_toolkit
- Скрипты для обслуживания кластера
  - Обзор
  - Пример использования настраиваемой функции для VACUUM

Перед установкой или обновлением Greenplum 6 для версий ниже 6.27.1 рекомендуется обновить расширение **arenadata_toolkit**. Это позволит использовать новые версии утилиты и получить доступ к новым функциям. Для обновления расширения выполните команду `gpadmin$ gpadmin$ gpadmin$

Управление spill-файлами

Д. Дилан Карыкина

Содержание

Назад к содержанию

Обзор
Анализ использования spill-файлов
Управление генерацией spill-файлов

Обзор

Spill-файлы (также известные как **work-файлы**) создаются на диске, если оперативной памяти становится недостаточно для хранения временных данных, формируемых при выполнении запросов в БД. Временные данные могут включать в себя хеши `JOIN`, результаты сортировок, пересоединенные таблицы и так далее. Директорию для хранения spill-файлов можно определить с помощью серверного конфигурационного параметра temp_tablespaces.

Поскольку spill-файлы хранятся на жестком диске, а скорость чтения/записи данных для дисков значительно ниже по сравнению с RAM – общее время выполнения запросов возрастает. Это может также привести к увеличению очередей на получение ресурсов со стороны других пользователей. В идеальном случае запросов, формирующих spill-файлы, быть не должно. Однако небольшое количество spill-файлов допускается. Фактически они обеспечивают выполнение запросов, мелкотря на ограниченный размер RAM. Но стоит избегать запросов, формирующих избыточное число spill-файлов.

Существует несколько типовых ситуаций, когда возможна генерация большого количества spill-файлов:

- **В системе присутствует "тяжелое данных" (skew).** Как избежать этого, можно узнать в разделе Рекомендации по равномерному распределению данных.
- **Объем памяти, выделенный под выполнение запросов, слишком мал.** Контролировать его величину возможно с помощью серверных конфигурационных параметров `max_statement_mem` и `statement_mem`. Также существует возможность настройки ресурсных групп.
- **Запрос не оптимизирован:**
  - **Выражение DISTINCT применяется к большому количеству столбцов.** Каждая уникальная комбинация столбцов, указанных в `DISTINCT`, будет помещаться в оперативную память. Соответственно, чем больше таких комбинаций будет обнаружено, тем быстрее произойдет заполнение памяти и начнут создаваться spill-файлы. Рекомендуется ограничивать число искомых полей, а также не помещать в выражение `DISTINCT` столбцы с данными большого размера.
  - **Используется сортировка с помощью ORDER BY.** Промежуточные результаты сортировки изначально сохраняется в RAM. Чем больше объем данных, которые предстоит сортировать, тем быстрее начнется генерация spill-файлов. В Greenplum не рекомендуется использовать сортировку в запросах, особенно для больших таблиц. Вместо этого лучше сортировать результаты непосредственно в применении данных (на стороне клиента).
  - **Большой объем данных обрабатывается без использования фильтров в секции WHERE.** При выполнении `JOIN`-запросов данные одной из таблиц помещается в хеш-таблицу в оперативной памяти. Соответственно, если исходная хеш-таблица не будет отфильтрована – память очень быстро переполнится. Рекомендуется по возможности всегда использовать фильтры, особенно на даты и время. Даже если есть необходимость выборки за весь период – желательно сформировать несколько запросов за короткие временные промежутки, после чего можно объединить результаты с помощью `UNION ALL`. Такой подход эффективнее попытки обработать все данные целиком.
  - **Хеш-таблица, создаваемая при выполнении JOIN-запросов, строится на основе большой из таблиц.** При выполнении `INNER JOIN` хеш-таблица, которая упоминалась выше, строится по таблице с наименьшим объемом данных (чтобы минимизировать использование оперативной памяти). Однако, например, при использовании `LEFT JOIN` за основу для хеша всегда берется таблица, справа, независимо от ее объема, что может привести к проблемам, если эта таблица в паре наибольшая. Поэтому внимательно следите за порядком таблиц в запросах. Также стараайтесь минимизировать количество извлекаемых столбцов.

РЕКОМЕНДАЦИЯ

- Стараайтесь уменьшить количество spill-файлов в первую очередь, через оптимизацию запросов, политики распределения данных и настроек потребления памяти.
- Если указанные выше действий недостаточно, можно настроить серверные конфигурационные параметры для ограничения количества spill-файлов и занимаемого ими дискового пространства.

Анализ использования spill-файлов

Схема `gp_toolkit` содержит ряд представлений (views), показывающих информацию по всем запросам, которые используют spill-файлы в настоящий момент времени. Эти представления могут быть полезны для анализа запросов, их оптимизации и своевременного предотвращения сбоев в системе:

- `gp_workfile_entries` – показывает операторы (query operator), вызывающие создание spill-файла на диске на сегментах в текущий момент времени. Одна результирующая строка относится к одному оператору.

Структура

Столбец	Описание
datname	Название базы данных
pid	Идентификатор серверного процесса (process ID)
sess_id	Идентификатор сессии (session ID)
command_cnt	Идентификатор запроса (command ID)
username	Название роли
query	Запрос к БД, выполняемый в текущий момент времени
segid	Уникальный идентификатор сегмента данных. Ссылается на `gp_segment_configuration.content`. Дополнительную информацию можно получить в разделе **Таблицы системного каталога → *gp_segment_configuration*** статьи Обзор таблиц
slice	Слайс (slice) плана выполнения запроса. Указывает на часть плана, выполняемую в текущий момент времени
optype	Тип оператора (query operator), вызывающего создание spill-файла
size	Общий объем сгенерированных spill-файлов на сегменте (в байтах)
numfiles	Количество созданных spill-файлов
prefix	Префикс, используемый для обозначения группы связанных друг с другом spill-файлов

Пример:

```
SELECT * FROM gp_toolkit.gp_workfile_entries;
```

Результат:

```
datname | pid  | sess_id | command_cnt | username |                    query                    | segid | slice | optype |  size  | numfiles | prefix
--------+------+---------+-------------+----------+---------------------------------------------+-------+-------+--------+--------+----------+--------
adb     | 5332 |      65 |          42 | gpadmin  | SELECT DISTINCT(id1,id2,gen1,gen2) from t1; |    -1 |     0 | Sort   |  4292688 |        1 | Sort_1
(1 row)
```

- `gp_workfile_usage_per_query` – показывает запросы (query), вызывающие создание spill-файлов на диске на сегментах в текущий момент времени. Одна результирующая строка относится к одному запросу.

Структура

Столбец	Описание
datname	Название базы данных
pid	Идентификатор серверного процесса (process ID)
sess_id	Идентификатор сессии (session ID)
command_cnt	Идентификатор запроса (command ID)
username	Название роли
query	Запрос к БД, выполняемый в текущий момент времени
segid	Уникальный идентификатор сегмента данных. Ссылается на `gp_segment_configuration.content`. Дополнительную информацию можно получить в разделе **Таблицы системного каталога → *gp_segment_configuration*** статьи Обзор таблиц
size	Общий объем сгенерированных spill-файлов на сегменте (в байтах)
numfiles	Количество созданных spill-файлов

Пример:

```
SELECT * FROM gp_toolkit.gp_workfile_usage_per_query;
```

Результат:

```
datname | pid  | sess_id | command_cnt | username |                    query                    | segid |   size   | numfiles
--------+------+---------+-------------+----------+---------------------------------------------+-------+----------+----------
adb     | 5332 |      65 |          42 | gpadmin  | SELECT DISTINCT(id1,id2,gen1,gen2) from t1; |    -1 | 20807680 |        1
(1 row)
```

- `gp_workfile_usage_per_segment` – показывает общий объем дискового пространства, занимаемый spill-файлами на каждом сегменте в текущий момент времени. Одна результирующая строка относится к одному сегменту.

Структура

Столбец	Описание
segid	Уникальный идентификатор сегмента данных. Ссылается на `gp_segment_configuration.content`. Дополнительную информацию можно получить в разделе **Таблицы системного каталога → *gp_segment_configuration*** статьи Обзор таблиц
size	Общий объем сгенерированных spill-файлов на сегменте (в байтах)
numfiles	Количество созданных spill-файлов

Пример:

```
SELECT * FROM gp_toolkit.gp_workfile_usage_per_segment;
```

Результат:

```
segid |   size   | numfiles
------+----------+----------
    5 |        0 |        0
   15 |        0 |        0
   10 |        0 |        0
   11 |        0 |        0
    1 |        0 |        0
    6 |        0 |        0
   12 |        0 |        0
    0 |        0 |        0
    2 |        0 |        0
    7 |        0 |        0
   13 |        0 |        0
    4 |        0 |        0
    8 |        0 |        0
    3 |        0 |        0
   -1 | 31752192 |        1
   14 |        0 |        0
    9 |        0 |        0
(17 rows)
```

ПРИМЕЧАНИЕ

Приведенные выше представления могут просматривать все пользователи. Однако те из них, кто не является суперпользователем, могут получить информацию только по тем БД, которые доступны для их ролей.

Управление генерацией spill-файлов

Для ограничения количества spill-файлов и размера выделяемого им дискового пространства можно использовать несколько конфигурационных параметров. Их описание приведено ниже.

Название	Описание	Значение по умолчанию	Тип
gp_workfile_limit_files_per_query	Максимальное количество spill-файлов, допустимое для создания одним запросом на одном сегменте. При превышении лимита возникает следующая ошибка `ERROR: number of workfiles per query limit exceeded.`. Значение `0` соответствует неограниченному количеству spill-файлов	100000	coordinator session reload
gp_workfile_limit_per_query	Максимальный размер дискового пространства, выделенного под spill-файлы, которые генерируются одним запросом на одном сегменте (в КБ). При превышении лимита возникает следующая ошибка `ERROR: workfile per query size limit exceeded.`. Значение `0` означает отсутствие ограничений	25000000	coordinator session reload
gp_workfile_limit_per_segment	Максимальный размер дискового пространства, выделенного под spill-файлы, которые генерируются всеми запущенными запросами на одном сегменте (в КБ). При превышении лимита возникает следующая ошибка `ERROR: workfile per segment size limit exceeded.`. Значение `0` означает отсутствие ограничений	50000000	local system restart
gp_workfile_compression	Определяет, нужно ли использовать сжатие данных для spill-файлов, генерируемых операциями hash aggregation или hash join при выполнении запросов. Значение по умолчанию `off` означает отсутствие сжатия данных. Значение `on` используется для включения компрессии	off	coordinator session reload

ВАЖНО

- Значение параметра `gp_workfile_limit_files_per_query`, используемое по умолчанию (`100000`), достаточно для большинства запросов к БД. Прежде чем увеличивать значение этой величины, постарайтесь оптимизировать запросы, а также настройте распределение данных и потребления памяти.
- Ограничение количества и размера формируемых spill-файлов предотвращает сбои в системе при выполнении ресурсоемких запросов. Но имейте в виду, что это приводит к прерыванию запросов при достижении указанных в параметрах лимитов.
- Настройка сжатия spill-файлов с помощью параметра `gp_workfile_compression` может существенно повысить производительность запросов, генерирующих spill-файлы. Однако обратите внимание, что уменьшение числа операций ввода/вывода всегда происходит за счет повышенного потребления CPU.

Удаление "брошенных" файлов в ADB

Дарья Барышева

Содержание

Назад к содержанию

Обзор

Настройка

Обзор

"Брошенные" (orphaned) файлы — это файлы, не ассоциированные ни с одной из существующих таблиц в базе данных. Они могут остаться в файловой системе после аварийного завершения некоторого бэкенд-процесса, связанного с созданием новой таблицы в рамках транзакции. Причинами сбоя могут быть отказы оборудования, нехватка свободного места на диске и так далее. В результате успешного восстановления системы после сбоя незакоммиченные изменения в базе данных отсутствуют (новой таблицы нет в системном каталоге), однако на сегмент-хостах могут остаться файлы данных этой таблицы. Со временем таких файлов, занимающих место на диске, может становиться все больше, а их поиск и удаление требуют времени и сопряжены с рисками удаления полезных данных.

Начиная с версии 6.29.0.3 в ADB 6 и 7.4.0 в ADB 7 СУБД поддерживает автоматическое отслеживание и удаление "брошенных" файлов на уровне ядра СУБД в случае прерывания транзакций. В предыдущих версиях ADB файлы данных удалялись автоматически лишь для штатно завершившихся транзакций (в случае их отката через `ABORT`), а "брошенные" файлы возможно было удалить лишь вручную (например, используя для их поиска функции и представления расширения gp_check_functions).

Подход к обработке "брошенных" файлов, реализованный в указанных выше версиях ADB, основан на следующих принципах:

- В случае коммита или отката транзакции, создавшей таблицу, ядро СУБД, как и ранее, в штатном режиме самостоятельно определяет, как обработать файлы данных таблицы. В случае вызова `ABORT` файлы удаляются, в случае `COMMIT PREPARED` — сохраняются.
- Для удаления брошенных файлов рассматриваются лишь транзакции в статусе `TRANSACTION_STATUS_IN_PROGRESS`. СУБД полагает, что транзакция в этом статусе может прерваться нештатно и оставить "брошенные" файлы. Транзакции в других статусах не рассматриваются, так как считаются обработанными штатным образом (см. предыдущий пункт).
- Решение о необходимости удаления файлов принимается СУБД автоматически на этапе выполнения восстановления (recovery).

ВАЖНО

- В статье технического блога Удаление брошенных файлов можно получить более подробную информацию о причинах появления orphaned-файлов, подходах к решению этой проблемы в других СУБД, а также познакомиться с деталями реализации от команды ADB/Greengage DB.
- Следует понимать, что установка указанных выше версий ADB не приводит к удалению всех накопившихся ранее "брошенных" файлов, однако предотвращает их накопление в будущем.

Настройка

Автоматическое отслеживание и удаление "брошенных" файлов в нештатных ситуациях регулируется при помощи GUC `gp_track_pending_delete`. По умолчанию параметру присвоено значение `on`, что означает использование новой обработки файлов, реализованной в ADB 6.29.0.3 и 7.4.0. Если по каким-либо причинам ее следует отключить, измените значение параметра на `off`. Следует учитывать, что параметр `gp_track_pending_delete` можно изменить только путем редактирования файла *postgresql.conf* на всех сегментах с помощью утилиты `gpconfig`, а для применения изменений требуется перезапуск СУБД (перезагрузки конфигурационного файла недостаточно).

Ниже показан пример изменения значения `gp_track_pending_delete`:

1. Подключитесь к хосту координатора ADB под пользователем `gpadmin`:

```
$ sudo su - gpadmin
```

2. Выведите текущее значение параметра:

```
$ gpconfig -s gp_track_pending_delete
```

Результат:

```
Values on all segments are consistent
GUC              : gp_track_pending_delete
Coordinator value: on
Segment     value: on
```

3. Измените значение параметра на всех сегментах:

```
$ gpconfig -c gp_track_pending_delete -v off
```

Результат:

```
20250827:10:10:29:2622827 gpconfig:bds-mdw:gpadmin-[INFO]:-completed successfully with parameters '-c gp_track_pending_delete -v
off'
```

4. Перезапустите СУБД:

```
$ gpstop -r -a
```

5. Проверьте значение параметра повторно, чтобы убедиться в применении изменений:

```
$ gpconfig -s gp_track_pending_delete
```

Вывод команды:

```
Values on all segments are consistent
GUC              : gp_track_pending_delete
Coordinator value: off
Segment     value: off
```

Базы данных

Дарья Барышева

Содержание

Назад к содержанию

Обзор
Просмотр существующих баз данных
- Способ 1 — psql
- Способ 2 — SQL-запрос

Создание базы данных

Обзор

База данных (БД) — это набор данных, физически хранимых вместе. Каждый ADB-кластер может содержать несколько баз данных. Однако клиентские приложения могут обращаться не более чем к одной базе данных ADB в рамках одного соединения: кросс-запросы между различными базами данных невозможны по причине жесткого разделения БД.

В каждом ADB-кластере существует несколько шаблонных (template) и используемых по умолчанию (default) БД. Их описание приведено в таблице ниже.

Шаблонные и используемые по умолчанию БД

Название	Описание
adb	Можно использовать в качестве БД по умолчанию для клиентских подключений. Все необходимые объекты можно добавлять в эту БД. Создается автоматически через ADCM — после применения действия Create database сервиса ADB. Тем не менее при необходимости можно создать и дополнительные пользовательские БД. Обратите внимание, что имя БД `adb` можно изменить перед применением действия ***Create database*** — в конфигурационных параметрах сервиса ADB (параметр ***Name of default database*** в секции ***ADB*** → ***Main***)
diskquota	Используется ADB-расширением `diskquota`. Это расширение предназначено для ограничения дискового пространства, выделяемого под схемы и роли. БД автоматически создается при установке ADB, если конфигурационный параметр ***Diskquota Extension*** имеет значение `true`, или при вызове действия Install diskquota сервиса ADB
postgres	Как правило, используется в качестве БД по умолчанию для административных задач. Например, в процессах запуска, в процессе обнаружения взаимных блокировок Global Deadlock Detector и так далее
template0	Этот шаблон используется для внутренних нужд СУБД. Возможно использование `template0` в качестве шаблона при создании "чистых" БД, содержащих только стандартные объекты, определяемые при инициализации СУБД. Не следует удалять или изменять `template0`
template1	Используется в качестве шаблона для новых БД, создаваемых с помощью команды `CREATE DATABASE` — за исключением случаев, когда явно указывается иной шаблон. Создание объектов в `template1` не рекомендуется

ВНИМАНИЕ

Не рекомендуется удалять либо изменять БД, приведенные выше.

Просмотр существующих баз данных

Для получения списка БД в текущем ADB-кластере можно использовать несколько способов. Два из них описаны ниже.

Способ 1 — psql

При работе с терминальным клиентом `psql`, для вывода списка БД можно воспользоваться метакомандой `\l`. Для этого выполните следующие шаги:

1. Подключитесь к координатор-хосту ADB (например, через SSH) под пользователем `gpadmin`, который создается по умолчанию:

```
$ sudo su - gpadmin
```

2. Получите список БД:

```
$ psql -l
```

Результат команды:

```
                              List of databases
   Name    |  Owner  | Encoding |  Collate   |   Ctype    |  Access privileges
-----------+---------+----------+------------+------------+---------------------
 adb       | gpadmin | UTF8     | en_US.utf8 | en_US.utf8 | =Tc/gpadmin         +
           |         |          |            |            | gpadmin=CTc/gpadmin
 diskquota | gpadmin | UTF8     | en_US.utf8 | en_US.utf8 |
 postgres  | gpadmin | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | gpadmin | UTF8     | en_US.utf8 | en_US.utf8 | =c/gpadmin          +
           |         |          |            |            | gpadmin=CTc/gpadmin
 template1 | gpadmin | UTF8     | en_US.utf8 | en_US.utf8 | =c/gpadmin          +
           |         |          |            |            | gpadmin=CTc/gpadmin
(5 rows)
```

Способ 2 — SQL-запрос

После подключения к выбранной БД через `psql`-клиент (или любую иную клиентскую программу) можно выполнить специальный SQL-запрос, который извлекает список БД из таблицы системного каталога `pg_catalog.pg_database`. Для этого выполните следующие шаги:

1. Подключитесь к координатор-хосту ADB (например, через SSH) под пользователем `gpadmin`, который создается по умолчанию:

```
$ sudo su - gpadmin
```

2. Подключитесь к БД через `psql` (или иную клиентскую программу):

```
$ psql adb
```

Результат команды:

```
psql (12.12)
Type "help" for help.
```

3. Выполните следующий SQL-запрос для вывода всех БД в ADB-кластере:

```
SELECT * FROM pg_catalog.pg_database ORDER BY datname;
```

Результат команды:

```
  datname  | datdba | encoding | datcollate |  datctype  | datistemplate | datallowconn | datconnlimit | datlastsysoid |
datfrozenxid | datminmxid | dattablespace |             datacl
-----------+--------+----------+------------+------------+---------------+--------------+--------------+---------------+---------
-----+------------+---------------+-----------------------------------
 adb       |     10 |        6 | en_US.utf8 | en_US.utf8 | f             | t            |           -1 |         12809 |
702 |          1 |          1663 | {=Tc/gpadmin,gpadmin=CTc/gpadmin}
 diskquota |     10 |        6 | en_US.utf8 | en_US.utf8 | f             | t            |           -1 |         12809 |
948 |          1 |          1663 |
 postgres  |     10 |        6 | en_US.utf8 | en_US.utf8 | t             | t            |           -1 |         12809 |
1081 |          1 |          1663 |
 template0 |     10 |        6 | en_US.utf8 | en_US.utf8 | t             | f            |           -1 |         12809 |
702 |          1 |          1663 | {=c/gpadmin,gpadmin=CTc/gpadmin}
 template1 |     10 |        6 | en_US.utf8 | en_US.utf8 | t             | t            |           -1 |         12809 |
1799 |          1 |          1663 | {=c/gpadmin,gpadmin=CTc/gpadmin}
(5 rows)
```

Создание базы данных

Чтобы создать новую базу данных, необходимо использовать команду `CREATE DATABASE` и затем указать имя базы данных.

В следующем примере создается новая база данных `books_store`.

```
CREATE DATABASE books_store;
```

Результат команды:

```
CREATE DATABASE
```

Чтобы подключиться к новой базе данных, можно использовать `psql`-команду `\c`:

```
\c books_store
```

Результат команды:

```
You are now connected to database "books_store" as user "gpadmin".
books_store=#
```

ПРИМЕЧАНИЕ

Чтобы создать базу данных, необходимо иметь соответствующие права для этой операции либо быть суперпользователем. Если у вас нет таких привилегий, пожалуйста, свяжитесь с вашим администратором БД.

Схемы

Д. Дмитр Карпов

Содержание

Назад к содержанию

Обзор
Просмотр существующих схем
Создание схемы
Путь поиска схемы
└ Изменение параметра search_path

Обзор

Схема — это логическое обслуживание сущностей внутри одной базы данных. В отличие от БД, схемы разделены не так жестко: пользователи могут обращаться к нескольким схемам одновременно (при наличии соответствующих прав). Схемы могут быть полезны для решения следующих задач:

- Организация справочников и баз данных, относящихся к одной базе данных.
- Разделение базы данных на несколько логических групп. Это может быть особенно полезно при настройке прав доступа к различным объектам БД.
- Выделение различных схем под стороннее приложение — так, чтобы не возникало конфликтов с именами объектов в различных схемах.

РЕКОМЕНДАЦИЯ

Обратите внимание, что допускается использовать объекты одного и того же типа (например, таблицы) с одинаковыми именами в одной БД — при условии, что эти объекты создаются в различных схемах.

Приведенные ниже таблицы описывают схемы, создаваемые по умолчанию в базе данных ADB.

Схемы, используемые по умолчанию

Название	Описание
arenadata_toolkit	Хранит настройки ADB и информацию о распределении таблиц и выполненных операциях
diskquota	Используется ADB расширением `diskquota`. Это расширение предназначено для ограничения дискового пространства, выделенного под схемы и роли. Схема создается автоматически при установке ADB, если соответствующее значение параметра `Diskquota Extension` имеет значение `True`, или при вызове действия Install Diskquota сервиса ADB.
gp_toolkit	Служебная схема, содержащая внешние таблицы, представления и функции, к которым можно обращаться с помощью SQL-команд. Схема содержит внешние таблицы и других системных метрик (включая размеры таблиц, blocat, индексы, ресурсные очереди, spill файлы и так далее)
information_schema	Состоит из стандартизированного набора представлений, содержащих информацию об объектах базы данных. Эти представления извлекают информацию из таблиц системного каталога и соответствуют со стандартам ANSI SQL 2008
kadb	Используется в Greffe to ADB Connector
madlib	Хранит информацию о функциях библиотеки Apache MADlib. Создается при установке ADB, если параметр соответствующей переменной `ADB MADlib Extension` имеет значение `True`, или при вызове действия Install MADlib сервиса ADB.
pg_aoseg	Хранит служебную информацию о таблицах append-optimized. Схема используется для внутренних нужд СУБД.
pg_bitmapindex	Хранит служебную информацию о bitmap индексах. Схема используется для внутренних нужд СУБД.
pg_catalog	Содержит полную информацию об объектах БД, таблицах системного каталога, встроенных типах данных, функциях и операторах
pg_toast	Схема больших объектов: объекты которых превышают размер страницы (Oversized-Attribute Storage Technique, TOAST). Схема используется для внутренних нужд СУБД.
public	Схема, используемая по умолчанию при создании новых объектов БД — за исключением случаев, когда явно указывается иная схема (или значение значения `search_path` пути поиска схемы). Все роли (пользователи) имеют привилегии `CREATE` и `USAGE` в схеме `public`.

ВНИМАНИЕ

Не рекомендуется удалять либо изменять схемы, приведенные выше.

Просмотр существующих схем

Для получения списка схем в текущей ADB-кластере выполните следующие шаги:

1. Подключитесь к координатору-ноду ADB (например, через SSH) под пользователем `gpadmin`, который создается по умолчанию:

```
$ sudo su - gpadmin
```

2. Подключитесь к БД через `psql` (или иную клиентскую программу):

```
$ psql adb
```

Результат команды:

```
psql (12.12)
Type "help" for help.
```

3. Выполняется одну из следующих команд:

- Сделайте выборку из таблицы `pg_catalog.pg_namespace`, содержащей данные обо всех схемах БД:

```
SELECT * FROM pg_catalog.pg_namespace ORDER BY nspname;
```

Результат команды:

```
  nspname   | nspowner |                   nspacl
------------+----------+--------------------------------------------------
 arenadata_toolkit |       10 | {gpadmin=UC/gpadmin,=U/gpadmin}
 diskquota  |       10 |
 gp_toolkit |       10 | {gpadmin=UC/gpadmin,=U/gpadmin}
 information_schema |       10 | {gpadmin=UC/gpadmin,=U/gpadmin}
 kadb       |       10 |
 madlib     |       10 |
 pg_aoseg   |       10 |
 pg_bitmapindex |       10 |
 pg_catalog |       10 | {gpadmin=UC/gpadmin,=U/gpadmin}
 pg_toast   |       10 |
 public     |       10 | {gpadmin=UC/gpadmin,=UC/gpadmin}
(11 rows)
```

- Сделайте выборку из таблицы `information_schema.schemata`. По сравнению с `pg_catalog.pg_namespace`, эта таблица хранит данные о схемах БД, в соответствии со стандартом ANSI SQL 2008.

```
SELECT * FROM information_schema.schemata ORDER BY schema_name;
```

Результат команды:

```
 catalog_name |   schema_name   | schema_owner | default_character_set_catalog | default_character_set_schema |
default_character_set_name | sql_path
--------------+-----------------+--------------+-------------------------------+------------------------------+
----------------------------+----------
 adb          | arenadata_toolkit | gpadmin      |                               |                              |
|                            |
 adb          | diskquota       | gpadmin      |                               |                              |
|                            |
 adb          | gp_toolkit      | gpadmin      |                               |                              |
|                            |
 adb          | information_schema | gpadmin      |                               |                              |
|                            |
 adb          | kadb            | gpadmin      |                               |                              |
|                            |
 adb          | madlib          | gpadmin      |                               |                              |
|                            |
 adb          | pg_aoseg        | gpadmin      |                               |                              |
|                            |
 adb          | pg_bitmapindex  | gpadmin      |                               |                              |
|                            |
 adb          | pg_catalog      | gpadmin      |                               |                              |
|                            |
 adb          | pg_toast        | gpadmin      |                               |                              |
|                            |
 adb          | public          | gpadmin      |                               |                              |
|                            |
(11 rows)
```

- При работе с терминальным клиентом `psql`, для вывода списка схем БД можно воспользоваться метакомандой `\dnS+`:

```
\dnS+
```

Результат команды:

```
                                   List of schemas
        Name        |  Owner  | Access privileges  |                  Description
----------------------------------------------------------------------------------------------
 arenadata_toolkit | gpadmin | gpadmin=UC/gpadmin |
 diskquota         | gpadmin |                    |
 gp_toolkit        | gpadmin | gpadmin=UC/gpadmin=|
                   |         | =U/gpadmin         |
 information_schema | gpadmin | gpadmin=UC/gpadmin=|
                   |         | =U/gpadmin         |
 kadb              | gpadmin |                    |
 madlib            | gpadmin |                    |
 pg_aoseg          | gpadmin |                    | Reserved schema for Append Only segment list and eof tables
 pg_bitmapindex    | gpadmin |                    | Reserved schema for internal relations of bitmap indexes
 pg_catalog        | gpadmin | gpadmin=UC/gpadmin=|  system catalog schema
                   |         | =U/gpadmin         |
 pg_toast          | gpadmin |                    |  reserved schema for TOAST tables
 public            | gpadmin | gpadmin=UC/gpadmin=|  standard public schema
                   |         | =UC/gpadmin        |
(11 rows)
```

Создание схемы

Чтобы создать новую схему базы данных, необходимо использовать команду `CREATE SCHEMA`, а затем указать имя схемы.

В следующем примере создается новая схема `test_schema` в текущей базе данных `adb`:

```
CREATE SCHEMA test_schema;
```

Результат команды:

```
CREATE SCHEMA
```

Если теперь воспользоваться любой из команд для вывода списка схем БД, результат будет следующим:

```
\dnS+
```

```
                                   List of schemas
        Name        |  Owner  | Access privileges  |                  Description
----------------------------------------------------------------------------------------------
 arenadata_toolkit | gpadmin | gpadmin=UC/gpadmin |
 diskquota         | gpadmin |                    |
 gp_toolkit        | gpadmin | gpadmin=UC/gpadmin=|
                   |         | =U/gpadmin         |
 information_schema | gpadmin | gpadmin=UC/gpadmin=|
                   |         | =U/gpadmin         |
 kadb              | gpadmin |                    |
 madlib            | gpadmin |                    |
 pg_aoseg          | gpadmin |                    | Reserved schema for Append Only segment list and eof tables
 pg_bitmapindex    | gpadmin |                    | Reserved schema for internal relations of bitmap indexes
 pg_catalog        | gpadmin | gpadmin=UC/gpadmin=|  system catalog schema
                   |         | =U/gpadmin         |
 pg_toast          | gpadmin |                    |  reserved schema for TOAST tables
 public            | gpadmin | gpadmin=UC/gpadmin=|  standard public schema
                   |         | =UC/gpadmin        |
 test_schema       | gpadmin |                    |
(12 rows)
```

Путь поиска схемы

Для доступа к объектам БД можно использовать как основные формы записи имен соответствующих ANSI-стандарту запись `database.schema.object`, так более короткий вариант `schema.object`. Вторая форма предпочтительна, поскольку в рамках одного соединения можно обращаться лишь к одной БД. Но существует также третий вариант записи, который не требует явного указания имени схемы и позволяет определять лишь именами объектов (например, именами таблиц). Это возможно благодаря использованию **пути поиска схемы (schema search path)**.

Путь поиска схемы определяется параметром `search_path`. Этот параметр определяет порядок, в котором будет производиться существующие схемы при поиске объекта БД. Фактически параметр `search_path` задает упорядоченный список. Запрашиваемый объект БД ищется в первой схеме списка, в случае неудачного поиска — во второй схеме и так далее. Если объект БД не найден ни в одной из схем списка, возвращается ошибка — даже в случае если объект присутствует в других схемах БД. Первая схема в списке является схемой по умолчанию — **default-схемой** (не также называют **current-схемой**). Это не только первая схема, в которой выполняется поиск, но и схема, в которой создается новый объект БД, если иное схема для них явно не определена.

Для вывода текущего значения пути поиска схемы можно использовать следующую команду:

```
SHOW search_path;
```

По умолчанию команда возвращает следующее значение:

```
  search_path
-----------------
 "$user", public
(1 row)
```

Первый элемент списка `"$user"` указывает на схему с тем же названием, что имя текущего пользователя. При отсутствии схемы это значение игнорируется. Второй элемент списка указывает на схему `public`. Именно по этой причине схема `public` используется для всех новых объектов БД по умолчанию. Для выхода схемы default (или current) можно использовать следующий запрос:

```
SELECT current_schema();
```

По умолчанию команда возвращает следующее значение:

```
 current_schema
----------------
 public
(1 row)
```

ПРИМЕЧАНИЕ

Схема `pg_catalog` всегда является частью пути поиска схемы. Даже если она явно в нем не указана.

Изменение параметра search_path

Изменение параметра `search_path` можно на уровне базы данных (через SQL-команду `ALTER DATABASE`) и на уровне роли (через SQL-команду `ALTER ROLE`). Ниже приведены последовательность шагов для первого метода:

1. Измените значение параметра `search_path` на уровне базы данных.

```
ALTER DATABASE adb SET search_path TO test_schema,public;
```

Результат команды:

```
ALTER DATABASE
```

2. Выйдите из `psql` (или любого иного используемого клиента) — чтобы перезапустить клиентскую сессию.

```
\q
```

3. Повторно подключитесь к базе данных `adb`:

```
$ psql adb
```

Результат команды:

```
psql (12.12)
Type "help" for help.
```

4. Создайте любую таблицу без явного указания схемы.

```
CREATE TABLE test(id INT) DISTRIBUTED BY(id);
```

Результат команды:

```
CREATE TABLE
```

5. Выведите схему, в которой создана таблица `test`:

```
\dt test
```

Убедитесь, что схема `test_schema` и не `public`:

```
                List of relations
  Schema   | Name | Type |  Owner  | Storage
-----------+------+------+---------+---------
 test_schema | test | table | gpadmin | heap
(1 row)
```

6. Верните исходное значение параметра `search_path` и повторно перезапустите клиентскую сессию.

```
ALTER DATABASE adb SET search_path TO "$user",public;
\q
```

ПРИМЕЧАНИЕ

Изменить значение `search_path` возможно также на уровне сессии с помощью запроса `SET search_path TO <new_path_value>;`. Где `<new_path_value>` — это новое значение параметра, например, `test_schema,public`. Так же можно, рекомендуется изменить этот параметр на уровне БД или роли, чтобы он не обрабатался при каждом перезапуске сессии.

Типы таблиц

Дарья Борышева

Содержание

Назад к содержанию

Обзор

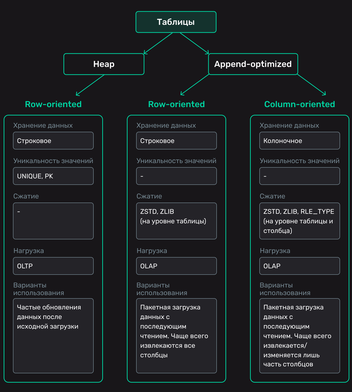
Сравнение таблиц heap и append-optimized

Сравнение таблиц row-oriented и column-oriented

Обзор

ADB, как и Greenplum DB, поддерживает несколько типов таблиц, отличающихся способом хранения данных:

- **Heap**. Этот тип таблиц используется по умолчанию и рекомендуется для OLTP-нагрузок. Heap-таблицы являются оптимальным выбором в случае частого обновления данных после первоначальной загрузки, а также подходят для одиночных операций `INSERT`, `UPDATE` и `DELETE`.
- **Append-optimized (AO)**. Эти таблицы предпочтительны для OLAP-нагрузок. Они также отлично подходят для пакетной загрузки данных (bulk data loading). Их рекомендуется использовать при редких обновлениях данных, то есть когда в системе преобладают read-only запросы. В отличие от heap-таблиц (в которых возможна только строковая ориентация данных — row-oriented), таблицы append-optimized поддерживают две формы ориентации данных:
  - **Строковая (row-oriented)**. Эта модель хранения данных рекомендуется для запросов, в которых одновременно извлекаются все либо большая часть столбцов таблицы.
  - **Колоночная (column-oriented)**. Эта модель подходит для вычислений на базе небольшого набора столбцов таблицы. Ее также рекомендуется использовать при регулярных обновлениях незначительной части столбцов.



Типы таблиц

Тип хранения данных необходимо явно указывать на этапе создания таблицы. Начиная с версии ADB 7 тип определяется **методом доступа (access method)** — соответствующий интерфейс был реализован в PostgreSQL 12.

В последующих подразделах описаны основные отличия существующих типов таблиц. Эта информация может быть полезной для выбора модели хранения данных, наиболее подходящей для нужд проекта и используемых в нем нагрузок.

Сравнение таблиц heap и append-optimized

Критерий	Heap	Append-optimized
Профиль нагрузки	OLTP	OLAP
Размер таблиц	Небольшие таблицы. Например, таблицы измерений (dimension table)	Денормализованные таблицы фактов (fact table), которые, как правило, являются самыми объемными в системе
Частота обновлений данных после изначальной загрузки	Часто	Редко
Преобладающий тип запросов	Множество итеративных транзакций, включая однострочные, а также параллельные пакетные обновления данных	▪ Пакетная загрузка данных. ▪ Частые выборки данных (read-only).
Возможность настройки уникальных значений	Поддерживается — за счет использования модификаторов `UNIQUE` и `PRIMARY KEY` при объявлении таблиц	—
Сжатие данных	—	▪ Row-oriented: `ZSTD`, `ZLIB` (на уровне таблицы). ▪ Column-oriented: `ZSTD`, `ZLIB`, `RLE_TYPE` (на уровне таблицы и столбца).

Синтаксис	Heap-таблицы создаются по умолчанию:
	`CREATE TABLE test_heap (a int, b text) DISTRIBUTED BY (a);`
		Для создания таблицы append-optimized со строковой ориентацией необходимо использовать выражение `USING ao_row`:
		`CREATE TABLE test_ao_row (a int, b text) USING ao_row DISTRIBUTED BY (a);`
		Для создания таблицы append-optimized с колоночной ориентацией необходимо использовать выражение `USING ao_column`:
		`CREATE TABLE test_ao_column (a int, b text) USING ao_column DISTRIBUTED BY (a);`
		Устаревшая форма `WITH (appendoptimized=true|false, orientation=row|column)` допускается

Рекомендации по использованию	OLTP-нагрузки — с множеством итеративных транзакций на базе небольших таблиц	Хранилища data warehouse (OLAP) — с пакетной загрузкой данных и последующими запросами на чтение (read-only) на базе больших денормализованных таблиц

ВНИМАНИЕ

Не рекомендуется применять следующие типы операций к таблицам append-optimized в силу их неэффективности:

- Однострочные операции `INSERT`, `UPDATE` и `DELETE`.
- Пакетные параллельные операции `UPDATE` и `DELETE` (это не касается пакетных запросов `INSERT`).

Сравнение таблиц row-oriented и column-oriented

Критерий	Row-oriented	Column-oriented
Тип таблиц	Heap и append-optimized	Append-optimized
Частота обновлений данных после изначальной загрузки	Часто	Редко
Частота добавления новых данных	Часто	Редко (таблицы с колоночной ориентацией данных не оптимизированы для `INSERT`-запросов, поскольку значения столбцов одной строки могут записываться в различные области на диске)
Количество столбцов в таблицах	Относительно небольшое	Множество столбцов (десятки и более)
Количество запрашиваемых столбцов	Все или большая часть	Небольшое подмножество столбцов

Преобладающий тип запросов	Любые	▪ `SELECT`-запросы, которые:
		• Выполняют агрегацию на базе одного столбца (при этом предикаты `WHERE` и `HAVING` также используют этот столбец):
		`SELECT SUM(price) FROM book; SELECT SUM(price) FROM book WHERE price > 1000;`
		• Возвращают относительно небольшое число столбцов (при этом предикат `WHERE` использует один столбец):
		`SELECT title FROM book WHERE public_year=2022;`
		▪ Запросы `UPDATE`, которые изменяют небольшое подмножество столбцов.

Сжатие данных	Возможно на уровне таблицы — в таблицах append-optimized. Алгоритмы: `ZSTD`, `ZLIB`	Возможно на уровне таблицы и столбца — в таблицах append-optimized. Алгоритмы: `ZSTD`, `ZLIB`, `RLE_TYPE`

Синтаксис	▪ Heap:
	`CREATE TABLE test_heap (a int, b text) DISTRIBUTED BY (a);`
	▪ Append-optimized:
	`CREATE TABLE test_ao_row (a int, b text) USING ao_row DISTRIBUTED BY (a);`
		Только append-optimized:
		`CREATE TABLE test_ao_column (a int, b text) USING ao_column DISTRIBUTED BY (a);`

Рекомендации по использованию	▪ Нагрузки OLTP-типа — с множеством итеративных транзакций, включая частые вставки и обновления данных. ▪ Нагрузки OLAP-типа или смешанные нагрузки — с преобладанием запросов, в которых одновременно извлекаются все или большая часть столбцов таблицы.	Хранилища data warehouse (OLAP) — с агрегацией на основе одного столбца или регулярными выборками/обновлениями небольшого набора столбцов

Сжатие данных

Дарья Барышева

Содержание

Назад к содержанию

Обзор

Рекомендации

Обзор

Для оптимизации I/O и уменьшения размера данных в кластерах ADB можно использовать сжатие — или компрессию — данных (data compression). Сжатие доступно только для append-optimized таблиц. В ADB доступны два типа сжатия данных:

- **Сжатие данных на уровне таблицы (table-level)** — применяется ко всей таблице. Доступно для append-optimized таблиц как со строковой (row-oriented), так и с колоночной (column-oriented) ориентацией данных.
- **Сжатие данных на уровне столбца (column-level)** — применяется к отдельному столбцу. Позволяет использовать различные алгоритмы сжатия для разных столбцов одной таблицы. Этот тип сжатия доступен только для append-optimized таблиц с колоночной ориентацией данных (column-oriented).

Типы сжатия данных

Ориентация данных	Уровень	Алгоритм	Пример
Строковая (row-oriented)	Таблица	▪ `ZLIB` — алгоритм deflate. ▪ `ZSTD` — алгоритм Zstandard.	Только на уровне таблицы: `CREATE TABLE ao_row_compressed (a int, b text) USING ao_row WITH (compresstype=zlib, compresslevel=5) DISTRIBUTED BY (a);`

Колоночная (column-oriented)	Таблица и отдельные столбцы	▪ `ZLIB` — алгоритм deflate.
		▪ `ZSTD` — алгоритм Zstandard.
		▪ `RLE_TYPE` — алгоритм run-length encoding.
			▪ На уровне таблицы:
			`CREATE TABLE ao_col_compressed_table (a int, b text) USING ao_column WITH (compresstype=zlib, compresslevel=5) DISTRIBUTED BY (a);`
			▪ На уровне столбца:
			`CREATE TABLE ao_col_compressed_column (a int, b text ENCODING (compresstype=RLE_TYPE, compresslevel=1)) USING ao_column DISTRIBUTED BY (a);`

Независимо от уровня, на котором применяется сжатие данных, для его настройки можно использовать следующие параметры (storage directive). Они заполняются в выражении `WITH` или `ENCODING`, соответственно:

- `compresstype` — тип сжатия данных. Возможные значения: `ZLIB`, `ZSTD` и `RLE_TYPE`. Значения не чувствительны к регистру. По умолчанию используется значение `none`, при котором сжатие не применяется.
- `compresslevel` — уровень сжатия данных. Возможные значения для различных алгоритмов приведены в таблице ниже. Уровни с наименьшими номерами соответствуют самой быстрой, но при этом наименьшей компрессии данных.

ВАЖНО

При одновременном описании сжатия данных в выражениях `ENCODING` и `WITH` более высокий приоритет у выражения `ENCODING`.

Параметры для настройки сжатия данных

Параметр	Deflate	Zstandard	Run-length encoding
compresstype	ZLIB	ZSTD	RLE_TYPE
compresslevel	Возможные значения: `1` — `9`. ▪ `1` — наиболее быстрый метод с наименьшим сжатием данных. Используется по умолчанию. ▪ `9` — самый медленный метод с наибольшим сжатием данных.	Возможные значения: `1` — `19`. ▪ `1` — наиболее быстрый метод с наименьшим сжатием данных. Используется по умолчанию. ▪ `19` — самый медленный метод с наибольшим сжатием данных.	Возможные значения: ▪ `1` — применить только `RLE_TYPE`. Это наиболее быстрый метод с наименьшим сжатием данных. Используется по умолчанию. ▪ `2` — применить `RLE_TYPE`, а затем `ZLIB` с уровнем сжатия `1`. ▪ `3` — применить `RLE_TYPE`, а затем `ZLIB` с уровнем сжатия `5`. ▪ `4` — применить `RLE_TYPE`, а затем `ZLIB` с уровнем сжатия `9`. Это самый медленный метод с наибольшим сжатием данных. ▪ `5` — применить `RLE_TYPE`, а затем `ZSTD` с уровнем сжатия `1`. ▪ `6` — применить `RLE_TYPE`, а затем `ZSTD` с уровнем сжатия `3`.

Рекомендации

Для наиболее эффективного сжатия данных рекомендуется:

- Выбирать наиболее подходящий под конкретную задачу алгоритм сжатия:
  - `ZSTD` — обеспечивает как скорость, так и хорошую степень сжатия, которую можно сконфигурировать с помощью опции `compresslevel`.
  - `ZLIB` — по большей части, используется для обратной совместимости. Как правило, `ZSTD` превосходит этот тип сжатия при обычных рабочих нагрузках.
  - `RLE_TYPE` — по сравнению с другими алгоритмами сжимает данные лучше в случаях, когда одни и те же значения данных встречаются во множестве последовательных строк. Но этот тип сжатия не подходит для таблиц, которые не содержат больших наборов повторяющихся данных.
- При выборе уровня сжатия данных следует помнить: чем выше этот уровень — тем больше, но дольше по времени сжатие данных. Фактически сжатие данных позволяет улучшить показатели I/O и уменьшить размер данных за счет ресурсов CPU. Поэтому всегда нужно соблюдать баланс между доступными ресурсами CPU и требуемой скоростью сжатия. Постарайтесь найти оптимальные настройки — чтобы избежать длительного сжатия и медленного сканирования данных.
- Конечная производительность сжатых таблиц зависит от множества факторов, включая аппаратное обеспечение, параметры запросов и так далее. Всегда проводите тестирование — чтобы определить, насколько эффективно сжатие данных в конкретном окружении.
- Не рекомендуется использовать сжатые таблицы в файловых системах, применяющих компрессию данных.

РЕКОМЕНДАЦИЯ

- Используйте тип сжатия `RLE_TYPE` для больших наборов повторяющихся данных.
- Для большинства других задач подходит `ZSTD` с уровнем сжатия данных `1` (сопоставим с `ZLIB` с уровнем сжатия `5`).

Распределение данных

Д. Дарья Карпусова

Содержание

Назад к содержанию

- Обзор типов распределения данных
- Рекомендации по равномерному распределению данных
- Примеры
  - Корректный выбор ключа распределения
  - Некорректный выбор ключа распределения

Обзор типов распределения данных

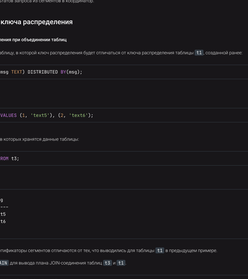
Распределение данных (distribution) – одна из самых важных концепций, лежащих в основе ADB. Она означает хранение данных каждой таблицы на различных сегментах. При этом или более равномерно данные распределяются между сегментами, тем выше производительность всего кластера. Для этого сегменты должны содержать примерно одинаковые порции данных.

Все таблицы в ADB распределяются между сегментами. При создании или изменении таблицы можно выбрать одну из доступных политик распределения данных (distribution policy) – для определения алгоритма распределения табличных данных между сегментами. Возможные политики приведены в таблице ниже.

Политики распределения данных

Тип	Описание	Пример
DISTRIBUTED BY (<column1>[,n])	Хэш-распределение. Конкретный сегмент выбирается на основе хешей, которые рассчитываются по полям, указанным в скобках. Рекомендуется использовать для таблиц, имеющих первичные ключи (`PRIMARY KEY`) либо столбцы с уникальными значениями (`UNIQUE`). Если столбцы могут быть использованы в качестве **ключа распределения** (поля, на основе значений которого выбирается сегмент)	`CREATE TABLE test_distr_key (a INT PRIMARY KEY, b TEXT) DISTRIBUTED BY (a);`
DISTRIBUTED REPLICATED	Распределение данных, при котором копии таблицы сохраняются на каждом сегменте кластера. Рекомендуется для небольших таблиц (например, таблиц справочников). Позволяет избежать любых перемещений данных (motion) при JOIN-запросах	`CREATE TABLE test_distr_replicated (a TEXT, b TEXT) DISTRIBUTED REPLICATED;`
DISTRIBUTED RANDOMLY	Случайное распределение данных с использованием алгоритма round-robin. Последующая система выбирает сегмент, на который будет записана строка. Распределение данных между ними не гарантируется. Рекомендуется для ситуации, когда в таблицах нет столбцов с уникальными значениями, а размер таблицы достаточно большой	`CREATE TABLE test_distr_random (a TEXT, b INT) DISTRIBUTED RANDOMLY;`

Рисунок ниже показывает в упрощенном виде, как выбрать наиболее оптимальную политику распределения данных.



Выбор политики распределения данных

Рекомендации по равномерному распределению данных

Для достижения максимальной производительности все сегменты кластера должны содержать примерно одинаковые порции данных. В противном случае может произойти **"перекос данных" (skew)**. Это фиксируется, во время которой на сегменты, хранящие большее количество данных по сравнению с другими сегментами, приходится более высокая нагрузка по обработке запросов. Чтобы избежать перекоса данных и сделать распределение данных более эффективным, необходимо следовать рекомендациям, перечисленным ниже:

1. Всегда явно указывать политику распределения данных при создании таблицы. При отсутствии выражения `DISTRIBUTED` распределение данных в таблице в последующем будет регулироваться конфигурационным параметром `gp_create_table_random_default_distribution`. А это в отдельных случаях может привести к выбору неудачной политики распределения – если в параметре не указано значение `ON`. Дополнительную информацию можно получить в данном справочнике.
2. Не выбирать в качестве ключей распределения столбцы, данные в которых распределяются неравномерно:
   - Даты и временные метки;
   - Поля, содержащие множество значений `NULL`;
   - Поля с наличием типичных данных, для которых множество дальнейшее распределение данных;

   Вместо этого рекомендуется выбирать ключ распределения, значение которого будет уникально для каждой строки таблицы – например, первичный ключ или любой уникальный столбец без значений `NULL`.

   При необходимости ключ можно сконструировать пользовательским хеш-функции для вычисления ключа распределения. Для этого запишите аргумент `hashalgorithm` в выражении `DISTRIBUTED BY` – именем соответствующего оператора сравнения. Более подробную информацию о синтаксисе можно найти в статье Таблицы.
3. Всегда проверять, что данные равномерно распределяются между сегментами после первоначальной загрузки данных и последующих обновлений. Чтобы узнать сегмент, в котором хранится текущая строка таблицы, можно выполнить следующий запрос:

```
SELECT gp_segment_id, * FROM test_distr_key;
```

Результат команды:

```
gp_segment_id | a | b
---------------+---+-----
            1 | 1 | first

(1 row)
```

Возвращаемое поле `gp_segment_id` соответствует номеру сегмента, хранимому в поле `content` системной таблицы `gp_segment_configuration`. Эта таблица содержит подробную информацию о всех сегментах кластера. По умолчанию поле скрыто. Оно может быть получено для всех таблиц, за исключением тех, для которых выбрана политика `DISTRIBUTED REPLICATED` (так как в этом случае данные дублируются на всех сегментах).

ВНИМАНИЕ

Не рекомендуется использовать системное поле `gp_segment_id` в запросах к большим таблицам на регулярной основе.

4. При необходимости можно изменить выбранный ранее тип распределения данных одним из описанных способов.
   - Изменение политики:

```
ALTER TABLE test_distr_replicated SET DISTRIBUTED RANDOMLY;
```

   - Перераспределение данных без изменения ключа распределения:

```
ALTER TABLE test_distr_key SET WITH (REORGANIZE=true);
```

5. Правильное распределение данных влияет на скорость выполнения запросов, особенно запросов `JOIN`. Для обеспечения максимальной эффективности запросов рекомендуется:
   - Не выбирать в качестве ключей распределения поля, которые будут активно использоваться в предикатах `WHERE`. В оптимальном случае все сегменты кластера должны выполнять равную долю работ по обработке каждого запроса данных. Но если предикат `WHERE` в запросе ссылается на ключ распределения, обработка запроса будет фактически сосредоточена только на одном сегменте (в котором хранится все строки с запрошенным значением ключа распределения).
   - Учитывать, что соединение таблиц с использованием ключей распределения самое быстрое, поскольку одинаковые данные хранятся на одном и том же сегменте – даже если для данных приходящих различными таблицы. Когда таблицы используют общий ключ распределения, соединение и сортировка по этому ключу выполняются локально – без необходимости распределения кортежей данных по другим сегментам. А локальные операции всегда быстрее, чем распределенные.
   - Использовать политику `DISTRIBUTED REPLICATED` для небольших и часто используемых таблиц справочников. Это повышает производительность запросов, позволяя избежать копирования таблицы во все сегменты (так называемый broadcast motion).
   - Использовать одинаковые типы данных в поле, на основе которых выполняется соединение таблиц в запросах `JOIN`.

Примеры

Корректный выбор ключа распределения

1. Создайте две таблицы со структурой, описанной ниже:

```
CREATE TABLE t1 (id INT, msg TEXT) DISTRIBUTED BY(id);
CREATE TABLE t2 (id INT, msg TEXT) DISTRIBUTED BY(id);
```

2. Добавьте в таблицы следующие данные:

```
INSERT INTO t1 (id, msg) VALUES (1, 'test1'), (2, 'test2');
INSERT INTO t2 (id, msg) VALUES (1, 'test1'), (2, 'test2');
```

3. Убедитесь, что данные с одинаковым значением ключа распределения в обоих таблицах сохранены на одном сегменте:

```
SELECT gp_segment_id, * FROM t1;
```

Результат команды:

```
gp_segment_id | id | msg
---------------+----+-----
            4 |  1 | test1
            9 |  2 | test2

(2 rows)
```

```
SELECT gp_segment_id, * FROM t2;
```

Результат команды схожа. Идентификаторы сегментов идентичны приведенным ранее:

```
gp_segment_id | id | msg
---------------+----+-----
            4 |  1 | test1
            9 |  2 | test2

(2 rows)
```

4. Используйте команду `EXPLAIN` для вывода плана JOIN-соединения двух таблиц:

```
EXPLAIN SELECT * FROM t1 JOIN t2 ON t1.id = t2.id;
```

Результат команды:

```
                              QUERY PLAN
-----------------------------------------------------------------------------------
Gather Motion 16:1  (slice1; segments: 16)  (cost=0.00..862.00 rows=2 width=20)
  ->  Hash Join  (cost=0.00..862.00 rows=1 width=20)
        Hash Cond: (t1.id = t2.id)
        ->  Seq Scan on t1  (cost=0.00..431.00 rows=1 width=10)
        ->  Hash  (cost=431.00..431.00 rows=1 width=10)
              ->  Seq Scan on t2  (cost=0.00..431.00 rows=1 width=10)
Optimizer: GPORCA
(7 rows)
```

Вывод команды показывает, что для этого запроса не требуется никаких перемещений данных (motion), кроме `Gather Motion` – это стандартное действие, означающее отправку результатов запроса из сегментов в координатор.

Некорректный выбор ключа распределения

Несовпадающие ключей распределения при объединении таблиц

1. Создайте дополнительную таблицу, в которой ключ распределения будет отличаться от ключа распределения таблицы `t1`, созданной ранее:

```
CREATE TABLE t3 (id INT, msg TEXT) DISTRIBUTED BY(msg);
```

2. Добавьте данные в таблицу:

```
INSERT INTO t3 (id, msg) VALUES (1, 'test3'), (2, 'test3');
```

3. Получите список сегментов, в которых хранятся данные таблицы:

```
SELECT gp_segment_id, * FROM t3;
```

Результат команды:

```
gp_segment_id | id | msg
---------------+----+-----
            3 |  1 | test3
           10 |  2 | test3

(2 rows)
```

Обратите внимание, что идентификаторы сегментов отличаются от тех, что выводились для таблицы `t1` в предыдущем примере.

4. Используйте команду `EXPLAIN` для вывода плана JOIN-соединения таблиц `t3` и `t1`:

```
EXPLAIN SELECT * FROM t1 JOIN t3 ON t1.id = t3.id;
```

Результат команды:

```
                              QUERY PLAN
-----------------------------------------------------------------------------------
Gather Motion 16:1  (slice2; segments: 16)  (cost=0.00..862.00 rows=2 width=20)
  ->  Hash Join  (cost=0.00..862.00 rows=1 width=20)
        Hash Cond: (t1.id = t3.id)
        ->  Seq Scan on t1  (cost=0.00..431.00 rows=1 width=10)
        ->  Hash  (cost=431.00..431.00 rows=1 width=10)
              ->  Broadcast Motion 16:16  (slice1; segments: 16)  (cost=0.00..431.00 rows=2 width=10)
                    ->  Seq Scan on t3  (cost=0.00..431.00 rows=1 width=10)
Optimizer: GPORCA
(8 rows)
```

По сравнению с предыдущим примером, в выводе команды добавляется действие `Broadcast Motion`. Во время этой операции происходит копирование таблицы во все сегменты кластера – поскольку в них нет необходимых данных для выполнения JOIN. Такая поведение неприятно для таблиц среднего размера. Однако в случае соединения большой таблицы оно может вызвать серьезные проблемы с производительностью. Чтобы устранить `Broadcast Motion` в приведенном примере, достаточно изменить ключ распределения в таблице `t3` с `msg` на `id`.

Выбор неуникального поля в качестве ключа

1. Создайте дополнительную таблицу:

```
CREATE TABLE t4 (id INT, msg TEXT) DISTRIBUTED BY(id);
```

2. Добавьте данные в таблицу. Обратите внимание, что значение первого столбца (выбранного в качестве ключа распределения) неуникальны:

```
INSERT INTO t4 (id, msg) VALUES (1, 'test7'), (1, 'test8');
```

3. Получите список сегментов, в которых хранятся данные таблицы:

```
SELECT gp_segment_id, * FROM t4;
```

Поскольку в таблице используется хеш-распределение, то при получении на вход одинаковых значений столбца `id` хеш-функция возвращает одинаковые значения идентификаторов сегментов:

```
gp_segment_id | id | msg
---------------+----+-----
            4 |  1 | test7
            4 |  1 | test8

(2 rows)
```

Вывод команды наглядно демонстрирует, что обе записи таблицы хранятся на одном сегменте, поскольку для них задано одно и то же значение ключа распределения. На практике крайне важно избегать такую ситуацию: так как они могут вызвать увеличенную нагрузки на отдельные сегменты. Чтобы устранить проблему в приведенном примере, достаточно указать в качестве ключа распределения другой, уникальный столбец таблицы. Если это невозможно – изменить политику распределения на `DISTRIBUTED RANDOMLY`.

Табличные пространства

Илья Кузнецов

Содержание

Перейти к содержанию

Обзор
Просмотр существующих табличных пространств
Создание табличного пространства
Использование табличного пространства для работы с объектами БД
Создание объектов
Проверка корректности заполнения табличного пространства
Перемещение объектов
Перемещение временных объектов
Удаление табличного пространства

Обзор

Таблица (таблица) или **табличное пространство** — это логическая общая структура хранения объектов базы данных (таблицы, индексы, материализованные представления, последовательности и прочие объекты), для которых, табличное пространство определяет, где данные, физически для них отведенные, обитают. Каждый объект БД принадлежит одному табличному пространству. Для каждого табличного пространства, на уровне файловой системы, выделяется отдельная директория, в которой, для каждого объекта, принадлежащего данному табличному пространству, создается отдельный подкаталог, в котором, для каждого объекта, принадлежащего данному табличному пространству, создаются отдельные файлы (первый сегмент, последующие сегменты, файлы карт видимости, файлы карт свободных блоков, файлы инициализации).

Каждый кластер баз данных, при инициализации, создает в каталоге данных общедоступной подкаталог `pg_tblspc`, в котором, для каждого табличного пространства, определенного в кластере баз данных, создается символическая ссылка, имя которой соответствует идентификатору данного табличного пространства. Чтобы найти расположение файлов для определенного табличного пространства, необходимо, по его идентификатору, обратиться к каталогу `pg_tblspc`.

Кластер баз данных, при инициализации, создает два табличных пространства: `pg_global` — для хранения общесистемных таблиц и `pg_default` — для хранения всех остальных объектов, для которых, при их создании, не определено, в какое табличное пространство, они должны быть помещены. Чтобы посмотреть существующие табличные пространства, можно воспользоваться командой `\db` утилиты `psql`, или обратиться к системному каталогу `pg_tablespace`.

Просмотр существующих табличных пространств

Для просмотра существующих табличных пространств, можно воспользоваться командой `\db` утилиты `psql`, или обратиться к системному каталогу `pg_tablespace`:

```
postgres=# \db
List of tablespaces
    Name    |  Owner   | Location
------------+----------+----------
 pg_default | postgres |
 pg_global  | postgres |
(2 rows)
```

Или, для получения более подробной информации, можно обратиться к системному каталогу `pg_tablespace`:

```
postgres=# select * from pg_tablespace;
  oid  | spcname | spcowner | spcacl | spcoptions
-------+---------+----------+--------+------------
 1663  | pg_default |       10 |        |
 1664  | pg_global  |       10 |        |
(2 rows)
```

Где:

- `oid` — уникальный идентификатор табличного пространства;
- `spcname` — имя табличного пространства;
- `spcowner` — идентификатор роли, которая является владельцем табличного пространства;
- `spcacl` — права доступа, выданные на табличное пространство;
- `spcoptions` — параметры табличного пространства, заданные на уровне табличного пространства.

Для получения полного пути к каталогу, в котором, для каждого из табличных пространств, хранятся файлы, принадлежащие объектам, размещенным в данном табличном пространстве, можно воспользоваться следующим запросом:

```
postgres=# select oid, spcname, pg_tablespace_location(oid) from pg_tablespace;
  oid  | spcname | pg_tablespace_location
-------+---------+------------------------
 1663  | pg_default |
 1664  | pg_global  |
(2 rows)
```

Где:

- `pg_tablespace_location(oid)` — возвращает полный путь к каталогу, в котором, для каждого из табличных пространств, хранятся файлы, принадлежащие объектам, размещенным в данном табличном пространстве.

Создание табличного пространства

Чтобы создать новое табличное пространство, необходимо воспользоваться командой `CREATE TABLESPACE`, указав имя табличного пространства и путь к каталогу, в котором, для каждого из файлов, будет создаваться, при создании, в данном табличном пространстве, объектов БД:

```
postgres=# CREATE TABLESPACE test_space_dir LOCATION '/home/postgre/';
CREATE TABLESPACE
```

Чтобы посмотреть, какие, в данный момент, существуют табличные пространства, можно воспользоваться командой `\db` утилиты `psql`:

```
postgres=# \db
List of tablespaces
    Name    |  Owner   | Location
------------+----------+----------
 pg_default | postgres |
 pg_global  | postgres |
 test_space | postgres | /home/postgre
(3 rows)
```

Или, для получения более подробной информации, можно обратиться к системному каталогу `pg_tablespace`:

```
postgres=# select * from pg_tablespace;
  oid  | spcname | spcowner | spcacl | spcoptions
-------+---------+----------+--------+------------
 1663  | pg_default |       10 |        |
 1664  | pg_global  |       10 |        |
 16384 | test_space |       10 |        |
(3 rows)
```

Или, с использованием системного каталога:

```
postgres=# select oid, spcname, pg_tablespace_location(oid) from pg_tablespace;
  oid  | spcname | pg_tablespace_location
-------+---------+------------------------
 1663  | pg_default |
 1664  | pg_global  |
 16384 | test_space | /home/postgre
(3 rows)
```

Где:

- `oid` — уникальный идентификатор табличного пространства;
- `spcname` — имя табличного пространства;
- `spcowner` — идентификатор роли, которая является владельцем табличного пространства;
- `spcacl` — права доступа, выданные на табличное пространство;
- `spcoptions` — параметры табличного пространства, заданные на уровне табличного пространства.

Для получения полного пути к каталогу, в котором, для каждого из табличных пространств, хранятся файлы, принадлежащие объектам, размещенным в данном табличном пространстве, можно воспользоваться следующим запросом:

```
postgres=# select oid, spcname, pg_tablespace_location(oid) from pg_tablespace;
  oid  | spcname | pg_tablespace_location
-------+---------+------------------------
 1663  | pg_default |
 1664  | pg_global  |
 16384 | test_space | /home/postgre
(3 rows)
```

Где:

- `pg_tablespace_location(oid)` — возвращает полный путь к каталогу, в котором, для каждого из табличных пространств, хранятся файлы, принадлежащие объектам, размещенным в данном табличном пространстве.

Чтобы создать новое табличное пространство, необходимо воспользоваться командой `CREATE TABLESPACE`, указав имя табличного пространства и путь к каталогу, в котором, для каждого из файлов, будет создаваться, при создании, в данном табличном пространстве, объектов БД:

```
postgres=# CREATE TABLESPACE test_space_dir LOCATION '/home/postgre/';
CREATE TABLESPACE
```

Чтобы, при запуске, указать, проверяется ли каждый табличное пространство на наличие ошибок:

- `on` — ошибки проверяются;
- `off` — ошибки не проверяются.

```
postgres=# \dt pg_tablespace;
List of tablespaces
  oid  | spcname | spcowner | spcacl | spcoptions
-------+---------+----------+--------+------------
 1663  | pg_default |       10 |        |
 1664  | pg_global  |       10 |        |
 16384 | test_space |       10 |        |
(3 rows)
```

Или, с использованием системного каталога:

```
postgres=# select oid, spcname, pg_tablespace_location(oid) from pg_tablespace;
  oid  | spcname | pg_tablespace_location
-------+---------+------------------------
 1663  | pg_default |
 1664  | pg_global  |
 16384 | test_space | /home/postgre
(3 rows)
```

Где:

- `oid` — уникальный идентификатор табличного пространства;
- `spcname` — имя табличного пространства;
- `spcowner` — идентификатор роли, которая является владельцем табличного пространства;
- `spcacl` — права доступа, выданные на табличное пространство;
- `spcoptions` — параметры табличного пространства, заданные на уровне табличного пространства.

Для получения полного пути к каталогу, в котором, для каждого из табличных пространств, хранятся файлы, принадлежащие объектам, размещенным в данном табличном пространстве, можно воспользоваться следующим запросом:

```
postgres=# select oid, spcname, pg_tablespace_location(oid) from pg_tablespace;
  oid  | spcname | pg_tablespace_location
-------+---------+------------------------
 1663  | pg_default |
 1664  | pg_global  |
 16384 | test_space | /home/postgre
(3 rows)
```

Где:

- `pg_tablespace_location(oid)` — возвращает полный путь к каталогу, в котором, для каждого из табличных пространств, хранятся файлы, принадлежащие объектам, размещенным в данном табличном пространстве.

Использование табличного пространства для работы с объектами БД

Создание объектов

Для создания объектов, табличное пространство, для которого, нужно определить, для его использования, нужно указать `TABLESPACE`, при создании объекта, или, при отсутствии такого, объект будет создан в табличном пространстве, определенном, по умолчанию, для данного, пользователя БД:

```
postgres=# CREATE TABLESPACE test_space_dir LOCATION '/home/postgre/';
CREATE TABLESPACE
postgres=# \dt pg_tablespace;
List of tablespaces
  oid  | spcname | spcowner | spcacl | spcoptions
-------+---------+----------+--------+------------
 1663  | pg_default |       10 |        |
 1664  | pg_global  |       10 |        |
 16384 | test_space |       10 |        |
(3 rows)
postgres=# select oid, spcname, pg_tablespace_location(oid) from pg_tablespace;
  oid  | spcname | pg_tablespace_location
-------+---------+------------------------
 1663  | pg_default |
 1664  | pg_global  |
 16384 | test_space | /home/postgre
(3 rows)
postgres=# CREATE DATABASE test_space_db TABLESPACE test_space_dir;
CREATE DATABASE
postgres=# \l test_space_db
List of databases
   Name    |  Owner   | Encoding | Collate | Ctype | Locale | Icu Locale | Icu Rules | Access privileges
-----------+----------+----------+---------+-------+--------+------------+-----------+-------------------
 test_space_db | postgres | UTF8     | ru_UTF.UTF8 | ru_UTF.UTF8 |        |            |           |
(1 row)
postgres=# \c test_space_db
You are now connected to database "test_space_db" as user "postgre".
```

- **Таблицы**:
  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl (id, text);
ERROR: syntax error at or near "("
LINE 1: \dt test_space_tbl (id, text);
                           ^
postgres=# \dt test_space_tbl (id, text) TABLESPACE test_space_dir;
ERROR: syntax error at or near "("
LINE 1: \dt test_space_tbl (id, text) TABLESPACE test_space_dir;
                           ^
postgres=# \dt test_space_tbl;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl;
Table "public.test_tbl"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl2 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl2;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl2 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl2;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl2 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl2;
Table "public.test_tbl2"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl2;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl2 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl3 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl3;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl3 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl3;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl3 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl3;
Table "public.test_tbl3"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl3;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl3 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl4 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl4;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl4 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl4;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl4 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl4;
Table "public.test_tbl4"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl4;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl4 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl5 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl5;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl5 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl5;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl5 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl5;
Table "public.test_tbl5"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl5;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl5 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl6 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl6;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl6 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl6;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl6 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl6;
Table "public.test_tbl6"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl6;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl6 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl7 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl7;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl7 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl7;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl7 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl7;
Table "public.test_tbl7"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl7;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl7 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl8 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl8;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl8 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl8;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl8 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl8;
Table "public.test_tbl8"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl8;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl8 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl9 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl9;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl9 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl9;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl9 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl9;
Table "public.test_tbl9"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl9;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl9 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl10 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl10;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl10 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl10;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl10 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl10;
Table "public.test_tbl10"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl10;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl10 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl11 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl11;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl11 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl11;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl11 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl11;
Table "public.test_tbl11"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl11;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl11 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl12 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl12;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl12 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl12;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl12 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl12;
Table "public.test_tbl12"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl12;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl12 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl13 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl13;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl13 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl13;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl13 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl13;
Table "public.test_tbl13"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl13;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl13 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl14 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl14;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl14 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl14;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl14 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl14;
Table "public.test_tbl14"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl14;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl14 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl15 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl15;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl15 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl15;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl15 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl15;
Table "public.test_tbl15"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl15;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl15 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl16 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl16;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl16 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl16;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl16 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl16;
Table "public.test_tbl16"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl16;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl16 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl17 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl17;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | test_tbl17 | table | postgres
(1 row)
postgres=# \dt+ test_space_tbl17;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl17 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
postgres=# \d+ test_space_tbl17;
Table "public.test_tbl17"
 Column |  Type  | Collation | Nullable | Default | Storage | Compression | Stats target | Description
--------+--------+-----------+----------+---------+---------+-------------+--------------+-------------
 id     | text   |           |          |         | extended |             |              |
 text   | text   |           |          |         | extended |             |              |
Table size: 0 bytes
Table Access Method: heap
Table Size: 0 bytes
Table Description:
postgres=# \dt+ test_space_tbl17;
List of relations
 Schema |   Name   | Type  |  Owner   | Persistence | Access method |  Size  | Description
--------+----------+-------+----------+-------------+---------------+--------+-------------
 public | test_tbl17 | table | postgres | permanent   | heap          | 0 bytes |
(1 row)
```

  - Для того, чтобы, при создании, таблицы, в данном табличном пространстве, использовать `TABLESPACE`, при создании таблицы, для каждой таблицы, должно, в данном, табличном пространстве, существовать, свободное, место, для, хранения, данных, принадлежащих, данной, таблице:

```
postgres=# CREATE TABLE test_space_tbl18 (id, text) TABLESPACE test_space_dir;
CREATE TABLE
postgres=# \dt test_space_tbl18;
List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public |
```

Обзор Arenadata DB Control

Дарья Барышева

Содержание

Назад к содержанию

Функции

Архитектура

Функции

Arenadata DB Control (ADB Control) — это система мониторинга запросов Arenadata DB в режиме реального времени.

Основные функциональные возможности ADB Control перечислены ниже:

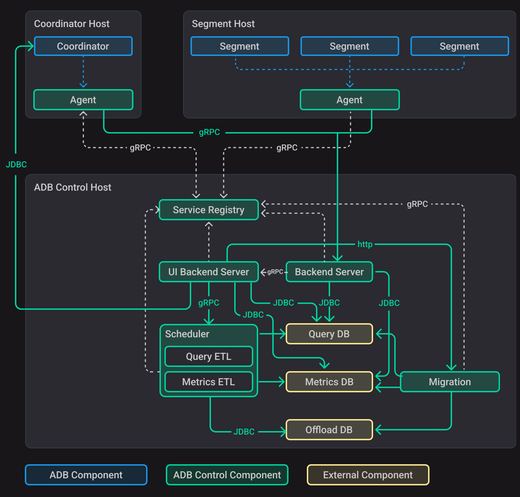
- Поддержка работы с несколькими кластерами ADB.
- Мониторинг SQL-запросов и транзакций, включая нижнеуровневые запросы (inner query).
- Мониторинг клиентских сессий ADB (как активных, так и неактивных) с возможностью их прерывания при необходимости.
- Сбор информации о потребляемых на уровне SQL-запросов и транзакций системных ресурсах.
- Построение сводной статистики по использованию системных ресурсов за выбранный период времени в виде графиков.
- Возможность отмены выполняющихся транзакций.
- Построение планов запросов с возможностью отслеживания прогресса их выполнения.
- Возможность выгрузки собираемых с кластеров метрик во внешнюю аналитическую базу данных (PostgreSQL/ADB).
- Аудит обращений к таблицам ADB (в кластерах, подключенных для мониторинга), а также авторизаций и других действий пользователей в ADB Control.

ПРИМЕЧАНИЕ

- ADB Control доступен в версии ADB Enterprise.
- В настоящий момент возможна только offline-установка ADB Control.
- Для использования ADB Control рекомендуется установить сервис Chrony. Синхронизация времени на хостах кластера необходима для корректной работы триггеров задач планировщика (Scheduler) и точных вычислений системных метрик.

Архитектура

Верхнеуровневая архитектурная схема ADB Control приведена ниже.



Архитектура ADB Control

Основные компоненты архитектуры ADB Control:

- **Agent**. Приложение для приема и фильтрации метрик с координатора и сегментов кластера ADB. Агенты устанавливаются в единственном экземпляре на каждый хост, то есть один агент принимает метрики со всех сегментов в рамках хоста.
- **Service Registry**. Каталог зарегистрированных в системе сервисов ADB Control. Отвечает за обнаружение сервисов (service discovery) в ADB Control. Обеспечивает обнаружение доступных агентов (преднастроенной карты агентов в системе нет) и поиск необходимых сервисов ADB Control со стороны агентов (для отправки метрик). Так как ADB Control может работать с несколькими кластерами ADB одновременно, при добавлении кластера новые агенты также регистрируются в Service Registry.
- **UI Backend Server**. Backend-сервер web-интерфейса ADB Control. Обрабатывает запросы со стороны Backend Server.
- **Backend Server**. Горизонтально масштабируемый сервис обработки и агрегации метрик, поступающих от агентов. Он же записывает информацию по запросам/метрикам в базы данных Query DB/Metrics DB и отправляет обновления на UI Backend Server.
- **Scheduler**. Планировщик, осуществляющий запуск задач по расписанию:
  - Автоматическое удаление из ADB Control метрик и иных данных, хранимых длительное время.
  - Выгрузка собираемых метрик и данных аудита во внешнюю базу данных (Offload DB).
  - Актуализация статусов запросов и транзакций.
  - Агрегация данных о потреблении системных ресурсов.
- **Migration**. Сервис автоматической миграции схем и данных в БД, с которыми взаимодействует ADB Control (Query DB, Metrics DB, Offload DB).
- **Query DB**. База данных PostgreSQL, выполняющая следующие функции:
  - Логирование запросов.
  - Логирование транзакций.
  - Хранение агрегатов системных метрик по транзакциям и запросам.
  - Аудит действий.
  - Ведение статистики по сущностям БД.
  - Хранение системных таблиц ADB Control с данными о пользователях, кластерах, настройках системы и так далее.
- **Metrics DB**. База данных ClickHouse, в которой хранятся:
  - Системные метрики, получаемые от агентов.
- **Offload DB**. Внешняя аналитическая база данных PostgreSQL/ADB для выгрузки собираемых метрик и данных аудита.

Использование внешних хранилищ Query DB и Metrics DB

- Базы данных Query DB и Metrics DB по умолчанию поднимаются в контейнерах. В обоих случаях существует возможность их подключения к ADB Control как внешних систем хранения. Для этого заполните параметры секции ***External database parameters*** (для Query DB) *и/или* ***External ADQM parameters*** (для Metrics DB) при конфигурировании сервиса ADB Control во время установки кластера ADB (см. Конфигурационные параметры → ADB Control).
- Начиная с версии ADB Control 4.7.5 создание пользователей во внешних БД и присвоение им необходимых прав производится вручную. Порядок настройки подробно описан в статье Настройка внешних БД для ADB Control.

ПРИМЕЧАНИЕ

Дополнительную информацию о том, как изменялась архитектура ADB Control по мере развития продукта, можно найти в статье технического блога Мониторинг запросов в Greenplum.

Настройка внешних БД для ADB Control

Дарья Барышева

Содержание

Назад к содержанию

PostgreSQL

ClickHouse

В production-средах рекомендуется использовать внешние базы данных PostgreSQL и ClickHouse для хранения данных ADB Control (см. Query DB и Metrics DB в статье Обзор Arenadata DB Control) по следующим причинам:

- У вас будет возможность настроить внешние БД на хостах с ресурсами, необходимыми для обеспечения высокой производительности, а также предусмотреть отказоустойчивость (High Availability, HA).
- Процесс восстановления ADB Control после сбоев будет проходить значительно проще и состоять по большей части из шагов по настройке сервисов.

Ниже приведены шаги, необходимые для настройки внешних БД начиная с версии ADB Control 4.7.5.

PostgreSQL

1. Выполните следующие запросы на стороне PostgreSQL. Запросы следует запускать под пользователем с правами superuser:

```
CREATE ROLE <user_name> LOGIN PASSWORD '<user_password>';
ALTER ROLE <user_name> SET search_path TO public, adcc;
CREATE DATABASE <database_name> WITH owner <user_name>;
```

2. Подключитесь к созданной базе данных `<database_name>` под суперпользователем и выполните:

```
CREATE SCHEMA adcc;
ALTER SCHEMA adcc OWNER TO <user_name>;

CREATE EXTENSION IF NOT EXISTS pgcrypto WITH schema public;
CREATE EXTENSION IF NOT EXISTS btree_gin WITH schema public;
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH schema public;

GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog TO <user_name>;

CREATE OPERATOR adcc.^| (PROCEDURE = jsonb_exists_any, LEFTARG = jsonb, RIGHTARG = _text, RESTRICT = contsel, JOIN = contjoinsel);
ALTER OPERATOR adcc.^|(jsonb, text[]) OWNER TO <user_name>;
CREATE OPERATOR CLASS adcc.jsonb_ops_custom
FOR TYPE jsonb USING gin AS
OPERATOR 7 pg_catalog.@>(jsonb, jsonb),
OPERATOR 10 adcc.^| (jsonb, _text),
FUNCTION 1  gin_compare_jsonb(text, text),
FUNCTION 2  gin_extract_jsonb(jsonb, internal, internal),
FUNCTION 3  gin_extract_jsonb_query(jsonb, internal, smallint, internal, internal, internal, internal),
FUNCTION 4  gin_consistent_jsonb(internal, smallint, jsonb, integer, internal, internal, internal, internal),
FUNCTION 6  gin_triconsistent_jsonb(internal, smallint, jsonb, integer, internal, internal, internal);
ALTER OPERATOR CLASS adcc.jsonb_ops_custom USING GIN OWNER TO <user_name>;
```

где `<user_name>` — имя созданного на предыдущем шаге пользователя.

Если есть возможность назначить права superuser пользователю в PostgreSQL (под которым будет выполняться подключение из ADB Control), достаточно создать роль и базу данных в PostgreSQL перед установкой или реконфигурацией ADB Control для использования внешней БД:

```
CREATE ROLE <user_name> LOGIN PASSWORD '<user_password>' SUPERUSER;
ALTER ROLE <user_name> SET search_path TO public, adcc;
CREATE DATABASE <database_name> WITH owner <user_name>;
```

3. Опционально можно запустить следующий запрос для отключения синхронной записи в WAL-файлы с целью увеличения производительности:

```
ALTER DATABASE <database_name> SET synchronous_commit = OFF;
```

где `<database_name>` — имя созданной выше БД.

4. Обеспечьте возможность доступа хоста с ADB Control к внешней базе данных PostgreSQL. Для этого на стороне сервера PostgreSQL требуется добавить запись следующего вида в файл *pg_hba.conf*.

Без SSL

```
host    <database_name>  <user_name>      <adbc_address>      trust
```

где:

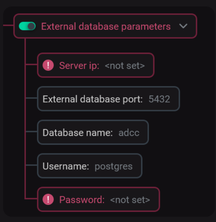
- `<database_name>` — имя созданной выше БД.
- `<user_name>` — имя созданного выше пользователя.
- `<adbc_address>` — IP-адрес ADB Control с подсетью.

С использованием SSL

```
hostssl    all  all       0.0.0.0/0      md5
```

Дополнительную информацию о предварительной настройке PostgreSQL для использования SSL можно получить в статье Шифрование SSL.

5. На странице конфигурирования сервиса ADB Control переведите в активное положение переключатель ***External database parameters*** и заполните параметры в открывшемся списке. Назначение параметров описано в разделе Конфигурационные параметры → ADB Control → External database parameters.



Параметры для подключения к внешней БД PostgreSQL

6. После заполнения параметров нажмите ***Save*** для сохранения конфигурации ADB Control.
7. Выполните установку сервиса ADB Control с помощью сервисного действия Install либо примените действие ***Reconfigure & Restart*** к ранее установленному сервису ADB Control.

ClickHouse

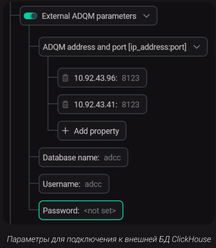
ВАЖНО

- Для использования кластеризованного ClickHouse в качестве внешней БД (вместо одиночной ноды) предварительно требуется создать логический кластер с именем `adcc_cluster` на стороне ClickHouse.
- В случае использования SSL-шифрования в ADB Control необходимо предварительно настроить SSL на стороне ClickHouse в соответствии со статьей Шифрование SSL и в параметрах подключения к хостам ClickHouse указывать порт `8443` (см. ***External ADQM parameters*** ниже).

1. Выполните следующие запросы для создания пользователя с необходимыми правами на стороне ClickHouse:

```
CREATE USER <user_name> NOT IDENTIFIED;
GRANT SELECT, INSERT, ALTER, CREATE, DROP, TRUNCATE ON *.* TO <user_name>
```

2. На странице конфигурирования сервиса ADB Control переведите в активное положение переключатель ***External ADQM parameters*** и заполните параметры в открывшемся списке. Назначение параметров описано в разделе Конфигурационные параметры → ADB Control → External ADQM parameters.



Параметры для подключения к внешней БД ClickHouse

3. После заполнения параметров нажмите ***Save*** для сохранения конфигурации ADB Control.
4. Выполните установку сервиса ADB Control с помощью сервисного действия Install либо примените действие ***Reconfigure & Restart*** к ранее установленному сервису ADB Control.

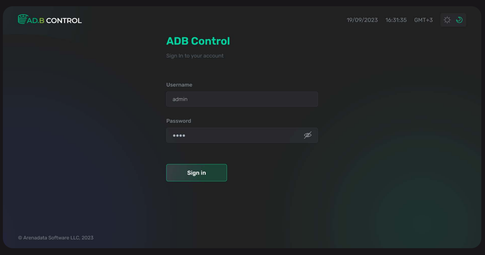
Подключение к ADB Control

Дарья Барышева

Назад к содержанию

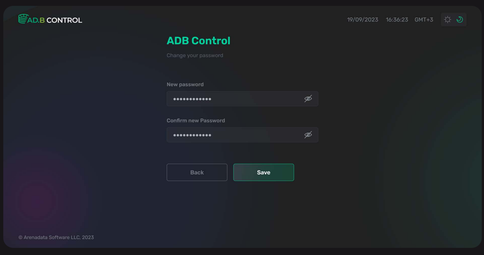
Для подключения к web-интерфейсу ADB Control выполните следующие шаги:

1. В адресной строке браузера введите адрес хоста, на котором развернут ADB Control. В качестве номера порта укажите значение параметра Listening port из раздела ***UI component parameters***, определенное на этапе конфигурирования сервиса ADB Control при установке кластера ADB (по умолчанию `81`).
2. В открывшемся окне введите учетные данные пользователя, который создается по умолчанию, и нажмите ***Sign in***:
   - ***Username:*** admin
   - ***Password:*** 1234



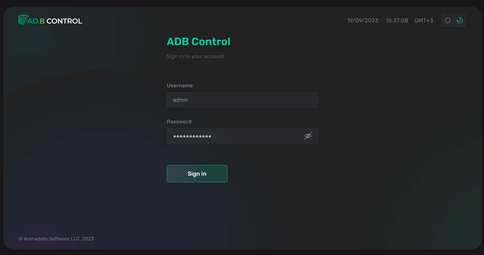
Подключение к ADB Control

3. При первом входе в систему необходимо изменить пароль для пользователя `admin`. Введите новый пароль в поле ***New password*** и подтвердите его в поле ***Confirm new Password***. Нажмите ***Save***. Обратите внимание, что длина пароля должна быть более либо равна 12 символам по умолчанию. К вводу допускаются любые символы кроме пробелов.



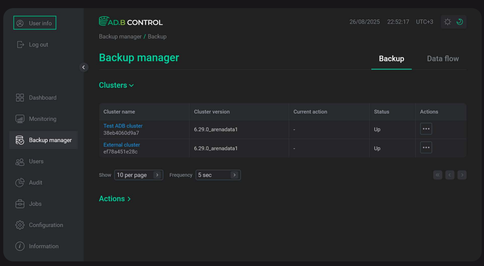
Смена пароля

4. После успешного сохранения нового пароля выполните вход под пользователем, повторно введя логин и новый пароль. Обратите внимание, что при достижении определенного числа неудачных попыток входа пользователь будет заблокирован — на постоянной или временной основе в зависимости от настроек.



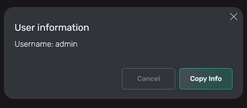
Вход с использованием измененного пароля

5. Если в дальнейшем потребуется получить информацию о пользователе, выполнившем вход в ADB Control — выберите вкладку ***User info*** в левом навигационном меню.



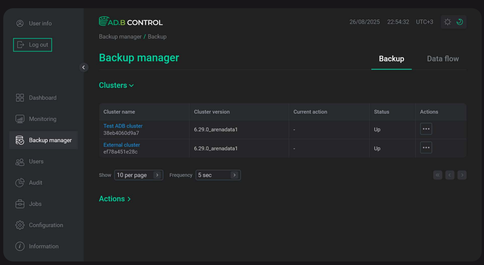
Получение информации о текущем пользователе

В появившемся модальном окне отображается имя текущего пользователя, которое можно скопировать нажав кнопку ***Copy info***.



Копирование информации о текущем пользователе

6. Для выхода из системы нажмите ***Log out*** в левом навигационном меню.



Выход из системы

ПРИМЕЧАНИЕ

- Информацию о том, как добавить новых пользователей и настроить права доступа для них, можно найти в следующих статьях:
  - Basic-аутентификация
  - Аутентификация с использованием LDAP
  - Авторизация
- Изменение политик безопасности (ограничения на длину пароля, максимальное количество неудачных попыток входа, правила блокировки пользователя) возможно на странице ***Configuration → Security***. За дополнительной информацией обратитесь к статье Настройка политик безопасности.

Basic-аутентификация

Версия документа

Содержание

Назад к содержанию

Просмотр списка пользователей
Добавление нового пользователя
Редактирование данных пользователя
Блокировка пользователя
Разблокировка пользователя
Удаление пользователя

При базовой аутентификации вход в ADB Control осуществляется под учетным записям, предварительно зарегистрированным в системе. Каждому пользователям при создании присваивается — роль, определяющая доступ к ресурсам в ADB Control.

Управление пользователями осуществляется на вкладке **Management** страницы **Users** в веб-интерфейсе ADB Control.

ВАЖНО

Операции меня операции доступны только пользователям с ролью **Owner**. Дополнительную информацию о роли можно получить в статье Авторизация.

Просмотр списка пользователей

Вкладка **Users** → **Management** выводит следующую информацию о пользователях:

Вкладка Users → Management

Поля вкладки Users → Management

Поле	Описание
Username	Имя пользователя. В зависимости от статуса и роли (см. ниже) в данном столбце дополнительно возможны следующие значки: 🟢 пользователь активен; 🔴 пользователь заблокирован; 🟡 пользователь временно заблокирован; 🔒 пользователь назначен роль **Owner**.
Status	Статус пользователя. Возможные значения: **Active** – активен; **Blocked** – заблокирован. Перевод в этот статус возможен при ручной блокировке пользователя либо автоматически согласно политике безопасности; **Temporarily blocked** – временно заблокирован согласно политике безопасности.
Last status change	Дата и время последнего изменения статуса (см. **Status** выше) в формате `DD/MM/YYYY HH:mm:ss`.
Role	Роль пользователя. Возможные значения: **ADB Administrator** – администратор ADM; **Administrator** – администратор ADB Control; **Advanced User** – пользователь с расширенными правами на доступ к определенным БД; **Owner** – владелец системы; **Regular User** – обычный пользователь.

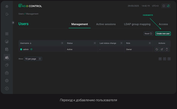
В заголовке столбцов **Role** и **Status** можно использовать фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо нажать на иконку ☰ для столбца, на основе значений которого требуется **Status**, в фильтре можно выбрать значения из выпадающего списка. Для применения фильтра нажмите **Apply**, для сброса фильтра нажмите **Reset**. Для отбора, пользователям для у каждого столбца **Last status change**, кроме фильтров диапазон можно выбрать за интервал.

Иконка 🔍 означает, что для столбца открыта фильтр. Для сброса всех фильтров нажмите **Reset**.

Добавление нового пользователя

Для добавления нового пользователя выполните следующие шаги:

1. Нажмите кнопку **Create new user** на вкладке **Users** → **Management**

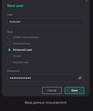


Переход к добавлению пользователя

2. В появившемся окне заполните следующие поля:
   - **User** – имя пользователя;
   - **Role** – роль пользователя;
   - **Password** – пароль пользователя.

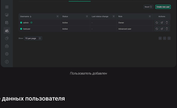
РЕКОМЕНДАЦИЯ

Для получения информации о требованиях к паролю и порядке начала нажмите курсор мыши на иконку ⓘ, которая появляется справа от названия полей **User** и **Password** во время их заполнения.



Ввод данных пользователя

3. Нажмите кнопку **Save**. В результате добавленный пользователь отобразится на вкладке **Users** → **Management**.

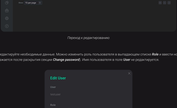


Пользователь добавлен

Редактирование данных пользователя

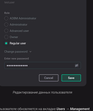
Для редактирования данных ранее созданного пользователя выполните следующие шаги:

1. Нажмите на иконку ✎ в столбце **Actions** на вкладке **Users** → **Management**



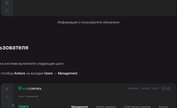
Переход к редактированию

2. В открывшемся окне отредактируйте необходимые данные. Можно изменить роль пользователя в выпадающем списке **Role** и ввести новый пароль в поле **Enter new password** (которое отображается после разрыва окна **Change password**). Имя пользователя в поле **User** не редактируется.



Редактирование данных пользователя

3. Нажмите кнопку **Save**. В результате информация о пользователе обновляется на вкладке **Users** → **Management**.

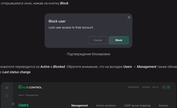


Информация о пользователе обновлена

Блокировка пользователя

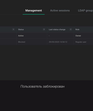
Для блокирования пользователя в системе выполните следующие шаги:

1. Нажмите на иконку 🔒 в столбце **Actions** на вкладке **Users** → **Management**



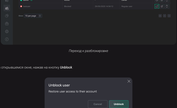
Переход к блокированию

2. Подтвердите операцию в открывшемся окне, нажав на кнопку **Block**



Подтверждение блокирования

В результате статус пользователя переходит в **Active** → **Blocked**. Обратите внимание, что на вкладке **Users** → **Management** также обновляется дата последнего изменения статуса в поле **Last status change**

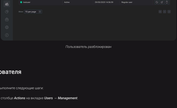


Пользователь заблокирован

Разблокировка пользователя

Для перевода ранее заблокированного пользователя в активное состояние выполните следующие шаги:

1. Нажмите на иконку 🔓 в столбце **Actions** на вкладке **Users** → **Management**



Переход к разблокированию

2. Подтвердите операцию в открывшемся окне, нажав на кнопку **Unblock**.

Подтверждение разблокирования

В результате пользователю возвращается доступ к системе, а его статус переходит из **Blocked** в **Active**

Пользователь разблокирован

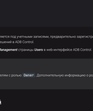
Удаление пользователя

Для удаления пользователя выполните следующие шаги:

1. Нажмите на иконку 🗑 в столбце **Actions** на вкладке **Users** → **Management**

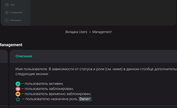
Переход к удалению

2. Подтвердите операцию удаления в открывшемся окне, нажав на кнопку **Delete**.



Подтверждение удаления

В результате данные о пользователе удаляются из системы и более не отображаются на вкладке **Users** → **Management**.



Пользователь удален из системы

Аутентификация с использованием LDAP

Дарья Барышева

AD.B CONTROL

Назад к содержанию

ADB Control поддерживает аутентификацию пользователей с использованием протокола Lightweight Directory Access Protocol (LDAP). При использовании LDAP нет необходимости в ручном добавлении пользователей, как это требуется при базовой аутентификации: проверка учетных записей пользователей и их паролей выполняется на выбранном LDAP-сервере.

Для настройки LDAP-аутентификации выполните следующие шаги:

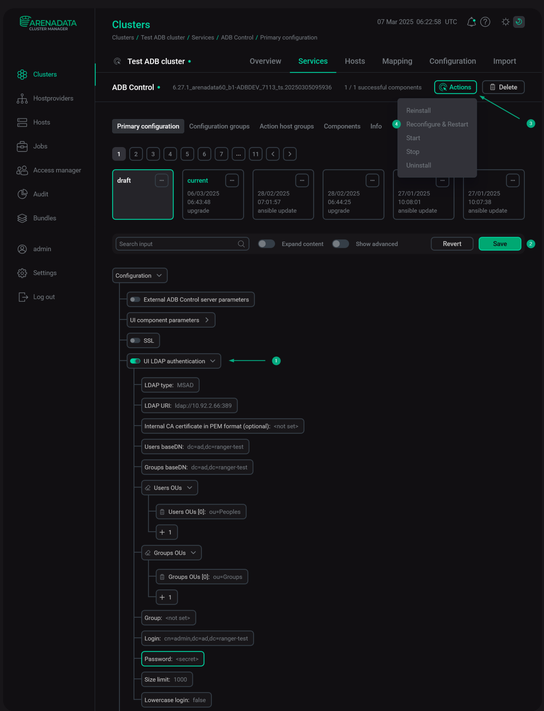
1. Убедитесь, что LDAP-сервер настроен и доступен. ADB Control поддерживает две реализации протокола LDAP: Microsoft Active Directory (MS AD) и 389 Directory server в составе FreeIPA.
2. Откройте конфигурационную страницу сервиса ADB Control в ADCM.
3. Переведите переключатель ***UI LDAP authentication*** в активное состояние.
4. Заполните следующие поля.

Поле	Описание	Пример в MS AD
LDAP type	Тип LDAP-сервера. Возможные значения: ▪ `MSAD` — Microsoft Active Directory; ▪ `FreeIPA` — 389 Directory server в составе FreeIPA.	MSAD
LDAP URI	URI для подключения к LDAP-серверу в формате `<схема>://<хост>:<порт>`. Например, `ldap://example.com:389` или `ldaps://example.com:636`. Если номер порта не указан, используются стандартные номера: `389` для LDAP и `636` для LDAPS	ldap://10.92.2.66:389
Internal CA certificate in PEM format (optional)	Внутренний сертификат центра сертификации (Certification Authority, CA) в формате PEM, используемый для подписи сертификата сервера LDAP	—
Users baseDN	Ограничение на область поиска объектов в каталоге LDAP, применяемое в запросах на поиск пользователей	dc=ad,dc=ranger-test
Groups baseDN	Ограничение на область поиска объектов в каталоге LDAP, применяемое в запросах на поиск групп пользователей	dc=ad,dc=ranger-test
Users OUs	Список `OU` для поиска пользователей. Если опция заполнена, поиск производится только в выбранных `OU` относительно ***Users baseDN***, иначе — относительно всего ***Users baseDN***. Каждое значение `OU` вводится в виде отдельной строки в формате `ou=<name>` (например, `ou=users1`)	ou=Peoples
Groups OUs	Список `OU` для поиска групп пользователей. Если опция заполнена, поиск производится только в выбранных `OU` относительно ***Groups baseDN***, иначе — относительно всего ***Groups baseDN***. Каждое значение `OU` вводится в виде отдельной строки в формате `ou=<name>` (например, `ou=groups1`)	ou=Groups
Group	Имя группы, используемое в сервисных запросах к LDAP-серверу	—
Login	Имя пользователя, используемое в сервисных запросах к LDAP-серверу	cn=admin,dc=ad,dc=ranger-test
Password	Пароль пользователя, используемый в сервисных запросах к LDAP-серверу	Пароль пользователя `admin`
Size limit	Максимальное количество записей, возвращаемое LDAP-сервером	1000
Lowercase login	Необходимо ли преобразование имен пользователей в нижний регистр	false

ВАЖНО

- Различные реализации LDAP используют разные имена для типов и идентификаторов объектов. Формат параметров для конкретной реализации LDAP рекомендуется уточнять у администратора LDAP-сервера.
- Для использования LDAP-аутентификации с SSL (LDAPS) начиная с версии ADB Control 4.10.3 необходимо:
  - Использовать протокол `ldaps` в поле ***LDAP URI***.
  - Заполнить секцию ***HTTPS parameters enable***.
  - Заполнить поле ***Internal CA certificate in PEM format (optional)***. Сертификат должен быть размещен на всех хостах, используемых ADB Control (импортирован в *truststore.jks*).

5. Нажмите ***Save***. Примените действие ***Reconfigure & Restart*** к сервису ADB Control.



Настройка LDAP-аутентификации для ADB Control в ADCM

6. В web-интерфейсе ADB Control выполните сопоставление групп пользователей LDAP-сервера и ролей ADB Control. Это необходимо для того, чтобы пользователи LDAP получили соответствующие разрешения на работу в ADB Control. Процесс настройки описан в статье Авторизация.

Если все шаги выполнены успешно, пользователи смогут подключаться к ADB Control, используя учетные записи, зарегистрированные для них на LDAP-сервере.

Содержание

Роли и уровни доступа

Сопоставление учетной записи с группами LDAP сервера
- Добавление сопоставления
- Редактирование сопоставления
- Удаление сопоставления

Назначение расширенных прав на доступ к БД
- Добавление расширения
- Редактирование расширения
- Удаление расширения

Развернуть все разделы

Роли и уровни доступа

Для разграничения прав доступа к функциям и данным в ПО Auth Service (далее Сервис) АО «Аксиома контроль» предусмотрены следующие роли и уровни доступа, которые могут быть назначены каждому пользователю:

- **Root** — суперпользователь;
- **Администратор** — администратор Сервиса;
- **Аудитор (только чтение)** — аудитор Сервиса;
- **Оператор** — оператор Сервиса;
- **Аутентификация через LDAP сервер** — пользователь, аутентификация которого осуществляется на сервере LDAP.

Назначение расширенных прав на доступ к БД:

- Добавление расширений
- Редактирование расширений
- Удаление расширений

Роли и уровни доступа

ОПИСАНИЕ

Встроенные роли доступа: **Root**, **Администратор**, **Аудитор** и **Оператор** в Сервисе АО «АКСИОМА КОНТРОЛЬ». Подробное описание ролей и уровней доступа приведено в таблице ниже.

ВАЖНО!

Для корректной работы сервиса необходимо в ОС Astra Linux SE 1.7.5 (Орел) с Мандатным МК ОС ОСКОМ ОС. Подробное описание ограничений приведено в документе «Руководство по установке».

Права доступа в АИС Control

Раздел	Root	Администратор	Аудитор (только чтение)	Оператор	Аутентификация через LDAP сервер
Меню «Учетные записи»	Да	Да	Да	Да	Нет
Просмотр всех учетных записей. Фильтрация по статусу, ролям и уровням доступа. Поиск учетной записи по имени пользователя, ФИО, email, телефону, комментарию. Сортировка по столбцам. Просмотр карточек учетных записей	Да	Да	Да	Да	Нет
Кнопка «Добавить»	Да	Да	Да	Да	Нет
Просмотр карточек учетных записей. Кнопка «Сохранить» доступна только при изменении данных. Кнопка «Отмена» доступна при изменении данных. Кнопка «Удалить» доступна для учетных записей, кроме учетной записи пользователя, под которой выполнен вход. Кнопка «Сброс пароля» доступна для учетных записей, кроме учетной записи пользователя, под которой выполнен вход. Кнопка «Разблокировать» доступна для заблокированных учетных записей	Да	Да	Да	Да	Нет
Кнопка «Сброс пароля»	Да	Да	Нет	Нет	Нет
Просмотр сопоставлений учетной записи с группами LDAP сервера. Кнопка «Добавить сопоставление». Кнопка «Редактировать». Кнопка «Удалить»	Да	Да	Да	Да	Нет
Меню «Роли и уровни доступа»	Да	Да	Да	Да	Нет
Просмотр всех ролей и уровней доступа. Поиск роли и уровня доступа по имени. Сортировка по столбцам. Просмотр карточек ролей и уровней доступа	Да	Да	Да	Да	Нет
Кнопка «Добавить»	Да	Да	Да	Да	Нет
Просмотр карточек ролей и уровней доступа. Кнопка «Сохранить» доступна только при изменении данных. Кнопка «Отмена» доступна при изменении данных. Кнопка «Удалить» доступна для ролей и уровней доступа, кроме роли и уровня доступа пользователя, под которой выполнен вход	Да	Да	Да	Да	Нет
Меню «Сопоставления»	Да	Да	Да	Да	Нет
Просмотр всех сопоставлений учетных записей с группами LDAP сервера. Поиск сопоставления по имени учетной записи, имени группы LDAP сервера. Сортировка по столбцам. Просмотр карточек сопоставлений	Да	Да	Да	Да	Нет
Кнопка «Добавить»	Да	Да	Да	Да	Нет
Просмотр карточек сопоставлений. Кнопка «Сохранить» доступна только при изменении данных. Кнопка «Отмена» доступна при изменении данных. Кнопка «Удалить» доступна для сопоставлений, кроме сопоставления пользователя, под которой выполнен вход	Да	Да	Да	Да	Нет
Меню «Расширения»	Да	Да	Нет	Нет	Нет
Просмотр всех расширений. Поиск расширения по имени. Сортировка по столбцам. Просмотр карточек расширений	Да	Да	Нет	Нет	Нет
Кнопка «Добавить»	Да	Да	Нет	Нет	Нет
Просмотр карточек расширений. Кнопка «Сохранить» доступна только при изменении данных. Кнопка «Отмена» доступна при изменении данных. Кнопка «Удалить» доступна для расширений, кроме расширения пользователя, под которой выполнен вход	Да	Да	Нет	Нет	Нет
Меню «Аудит»	Да	Да	Да	Нет	Нет
Просмотр всех записей аудита. Поиск записи аудита по имени пользователя, IP-адресу, ресурсу, действию, статусу. Сортировка по столбцам. Просмотр карточек записей аудита	Да	Да	Да	Нет	Нет
Кнопка «Экспорт»	Да	Да	Нет	Нет	Нет
Просмотр карточек записей аудита	Да	Да	Да	Нет	Нет
Меню «Настройки»	Да	Да	Нет	Нет	Нет
Просмотр настроек. Кнопка «Сохранить» доступна только при изменении данных. Кнопка «Отмена» доступна при изменении данных	Да	Да	Нет	Нет	Нет
Меню «Сервисы»	Да	Да	Нет	Нет	Нет
Просмотр всех сервисов. Поиск сервиса по имени. Сортировка по столбцам. Просмотр карточек сервисов	Да	Да	Нет	Нет	Нет
Кнопка «Добавить»	Да	Да	Нет	Нет	Нет
Просмотр карточек сервисов. Кнопка «Сохранить» доступна только при изменении данных. Кнопка «Отмена» доступна при изменении данных. Кнопка «Удалить» доступна для сервисов, кроме сервиса, под которым выполнен вход	Да	Да	Нет	Нет	Нет

Сопоставление учетной записи с группами LDAP сервера

Через сопоставление учетной записи с группами LDAP сервера пользователь авторизуется через LDAP сервер. Для этого в АИС Control создается сопоставление, в котором указывается имя группы LDAP сервера, которая будет использоваться для авторизации пользователя, и выбирается роль, которая будет назначена пользователю при авторизации через LDAP сервер.

Сопоставление учетной записи с группами LDAP сервера доступно в разделе **Меню** — **LDAP group mapping** и для создания в АИС Control.

Экран LDAP group mapping в Control

Добавление сопоставления

Для добавления сопоставления выполните следующие действия:

1. Перейдите в раздел **Меню** — **LDAP group mapping**.

Переход к добавлению сопоставления

2. Нажмите на кнопку **Добавить**.
3. В появившемся окне заполнить поля:
   - **Group** — укажите имя сопоставляемой LDAP группы. Для корректной работы с сопоставлением группа с таким именем должна уже существовать на LDAP сервере;
   - **Role** — роль, в АИС Control.

ВАЖНО!

Каждой группе сопоставляется LDAP сервер целиком. Если у сервера первая группа сопоставлена, то все остальные группы будут сопоставлены с LDAP сервером.

4. Нажмите на кнопку **Сохранить**.

Окно добавления

Сохранение добавления

Редактирование сопоставления

Для изменения сопоставления выполните следующие действия:

1. Перейдите в раздел **Меню** — **LDAP group mapping**.

Переход к редактированию

2. Нажмите на кнопку **Редактировать** у сопоставления, которое нужно изменить.

Окно редактирования сопоставления

3. В появившемся окне внесите изменения в поля **Group** и **Role**, которые, в случае необходимости, изменены ранее. Выполните кнопку **Сохранить** и нажмите **Да**.

Подтверждение сохранения

4. Нажмите на кнопку **Закрыть**, чтобы вернуться к списку сопоставлений.

Сохранение обновлений

Удаление сопоставления

Для удаления сопоставления выполните следующие действия:

1. Перейдите в раздел **Меню** — **LDAP group mapping**.

Переход к удалению

2. Нажмите на кнопку **Удалить** у сопоставления, которое нужно удалить.

Окно удаления

3. Подтвердите удаление сопоставления, нажмите на кнопку **Удалить**.

Подтверждение удаления

4. В появившемся окне подтвердите удаление, нажмите **Удалить**.

Подтверждение удаления

ОПИСАНИЕ

Встроенная учетная запись LDAP сервера — **LDAP server** — предназначена для авторизации через LDAP сервер. Для корректной работы с LDAP сервером необходимо в АИС Control создать сопоставление для учетной записи **LDAP server** с ролью и уровнем доступа, которые будут назначены пользователям при авторизации через LDAP сервер.

Учетная запись **LDAP server** — это системная **LDAP server** и не может быть удалена. Для корректной работы с LDAP сервером необходимо в АИС Control создать сопоставление для учетной записи **LDAP server** с ролью и уровнем доступа, которые будут назначены пользователям при авторизации через LDAP сервер.

Учетная запись **LDAP server** доступна в разделе **Меню** — **Учетные записи** — **LDAP server**. Для корректной работы с LDAP сервером необходимо в АИС Control создать сопоставление для учетной записи **LDAP server** с ролью и уровнем доступа, которые будут назначены пользователям при авторизации через LDAP сервер.

1. Перейдите в раздел **Меню** — **Учетные записи** — **LDAP server**.

Переход к редактированию учетной записи

2. Нажмите на кнопку **Редактировать**.

Окно редактирования учетной записи

3. Отредактируйте, в случае необходимости, поля **Email** и **Комментарий** и нажмите на кнопку **Сохранить**.

Подтверждение изменения

4. Подтвердите изменения, нажмите **Да**.

Подтверждение изменения учетной записи

Назначение расширенных прав на доступ к БД

Расширения для **Control DAT** позволяют выполнять запросы к БД, которые не предусмотрены в интерфейсе сервиса АИС. Каждое расширение привязывается к определенной роли и уровню доступа. Пользователи, которые имеют эту роль и уровень доступа, получают возможность выполнять запросы, которые были добавлены в расширении. Расширения доступны в разделе **Меню** — **Расширения**. Для корректной работы с расширениями необходимо в АИС Control создать расширение.

Управление расширениями доступно в разделе **Меню** — **Расширения** и для создания в АИС Control.

Экран расширений

Добавление расширения

Для добавления расширения выполните следующие действия:

1. Перейдите в раздел **Меню** — **Расширения**.

Переход к добавлению расширения

2. Нажмите на кнопку **Добавить расширение**.

Окно создания расширения

3. В появившемся окне заполнить поля:
   - **Role** — выберите роль, которая будет назначена пользователям, у которых есть доступ к расширению;
   - **Access** — выберите уровень доступа, который будет назначен пользователям, у которых есть доступ к расширению;
   - **Name** — укажите имя расширения;
   - **Description** — укажите описание расширения.
4. Нажмите на кнопку **Сохранить**.

Окно создания расширения

5. Подтвердите изменения, нажмите **Да**.

Подтверждение добавления

Редактирование расширения

Для изменения расширения выполните следующие действия:

1. Перейдите в раздел **Меню** — **Расширения**.

Переход к редактированию

2. Нажмите на кнопку **Редактировать** у расширения, которое нужно изменить.

Окно редактирования расширения

3. В появившемся окне внесите изменения в поля **Role**, **Access**, **Name** и **Description**, которые, в случае необходимости, изменены ранее. Выполните кнопку **Сохранить** и нажмите **Да**.

Подтверждение изменения

4. Нажмите на кнопку **Закрыть**, чтобы вернуться к списку расширений.

Сохранение обновления

Удаление расширения

Для удаления расширения выполните следующие действия:

1. Перейдите в раздел **Меню** — **Расширения**.

Переход к удалению

2. Нажмите на кнопку **Удалить** у расширения, которое нужно удалить.

Окно удаления

3. Подтвердите удаление расширения, нажмите на кнопку **Удалить**.

Подтверждение удаления

4. В появившемся окне подтвердите удаление, нажмите **Удалить**.

Подтверждение удаления

Сессии пользователей

Дарья Барышева

Содержание

Назад к содержанию

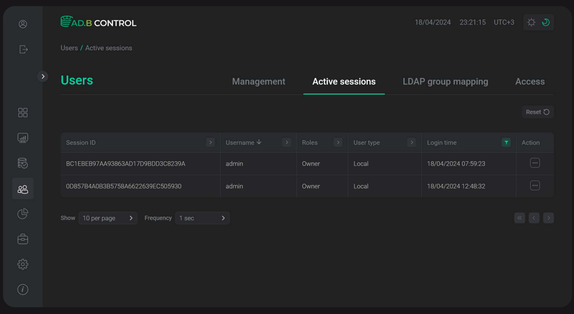
Просмотр списка сессий

Прерывание сессий

Вкладка ***Users → Active sessions*** предназначена для мониторинга активных сессий пользователей ADB Control. Функционал страницы позволяет также прерывать выбранные сессии.

Просмотр списка сессий

На вкладке ***Users → Active sessions*** выводится следующая информация о сессиях.



Список сессий

Поля вкладки Users → Active sessions

Поле	Описание
Session ID	Идентификатор сессии
Username	Имя пользователя, выполнившего подключение
Roles	Роль пользователя, выполнившего подключение
User type	Тип пользователя, выполнившего подключение. Возможные значения: - `AD` — пользователь LDAP; - `Local` — пользователь ADB Control, для которого настроена базовая аутентификация; - `System` — системный пользователь.
Login time	Время открытия сессии

В заголовках столбцов таблицы со списком сессий расположены фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо нажать на иконку > . Для тех столбцов, где набор возможных значений ограничен (например, ***Roles***), в фильтре можно выбрать значение из выпадающего списка. Для некоторых столбцов (например, ***Session ID***) требуется ввести искомое значение. Для столбцов, показывающих дату и время (например, ***Login time***), временной диапазон можно выбрать из календаря.

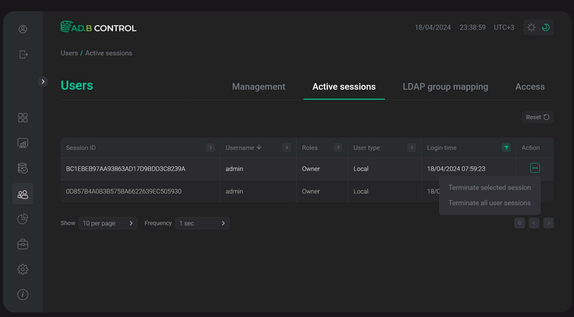
Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите ***Reset***.

Прерывание сессий

ПРИМЕЧАНИЕ

Доступ к прерыванию сессий есть только у пользователей ADB Control с соответствующими разрешениями (см. `Manage user sessions` в статье Авторизация).

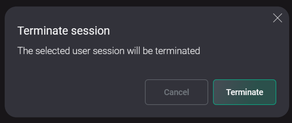
Чтобы прервать выполнение сессии, нажмите кнопку ••• в столбце ***Actions*** на вкладке ***Users → Active sessions***.



Переход к прерыванию сессии

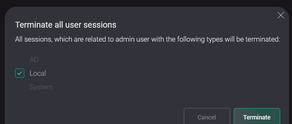
Выберите одной из следующих действий:

- ***Terminate selected session*** — прерывает только одну выбранную сессию. Для подтверждения действия нажмите ***Terminate*** в открывшемся окне.



Подтверждение прерывания одной сессии

- ***Terminate all user sessions*** — прерывает все сессии пользователя, запустившего выбранную сессию. При этом в открывшемся окне можно отметить тип пользователя (см. ***User type*** выше). Для подтверждения действия нажмите ***Terminate***.



Подтверждение прерывания всех сессий пользователя

Шифрование SSL

Дарья Барышева

AD.B CONTROL

Содержание

Назад к содержанию

ADB Control
Внешние базы данных
PostgreSQL
ClickHouse

ADB Control

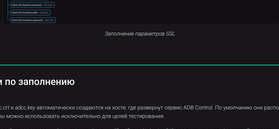
Для установки защищенного соединения между web-сервером ADB Control и клиентами можно использовать протокол HTTPS, поддерживающий технологию шифрования SSL. Для настройки доступа к ADB Control через HTTPS выполните следующие шаги:

1. Откройте конфигурационную страницу сервиса ADB Control в ADCM.
2. Установите флаг ***Show advanced***
3. Переведите переключатель ***SSL*** в активное состояние.
4. Заполните следующие поля, если вы планируете использовать собственные сертификаты.

ВАЖНО

Если настройки не будут заполнены, ADB Control создаст сертификаты самостоятельно.

Параметр	Описание	Значение по умолчанию
CA certificate	Содержимое сертификата центра сертификации (Certificate Authority, CA) в формате CRT (файл *.crt)	--
Server certificate	Содержимое серверного сертификата в формате CRT (файл *.crt)	--
Server private key	Содержимое серверного приватного ключа (файл *.key)	--
Verify system endpoints' certificates	Флаг, определяющий необходимость проверки сертификатов системных конечных точек. Перед использованием опции **Verify system endpoints' certificates** в конфигурации ADB Control с самоподписанным (self-signed) сертификатами убедитесь, что в хранилище доверенных корневых сертификатов ОС добавлены корневой и промежуточные CA-сертификаты Чтобы использовать опцию **Verify system endpoints' certificates** с вашими собственными сертификатами, для успешной верификации добавьте IP-адрес ADB Control в поле `CN` файла san.cnf (`SAN`), а также исключите DNS из секции `alt_names`: (оставьте только IP)	false
Server JKS keystore path	Путь к keystore-файлу в формате JKS для серверной части ADB Control. Например, /opt/adcc/ssl/keystore.jks	--
Server JKS keystore password	Пароль для доступа к keystore-файлу ***Server JKS keystore path***	--
Server JKS truststore path	Путь к truststore-файлу в формате JKS для серверной части ADB Control. Например, /opt/adcc/ssl/truststore.jks	--
Server JKS truststore password	Пароль для доступа к truststore-файлу ***Server JKS truststore path***	--
Client JKS keystore path	Путь к keystore-файлу в формате JKS для клиентской части ADB Control (агентов ADB Control/ADBM). Например, /opt/ssl/keystore.jks	--
Client JKS keystore password	Пароль для доступа к keystore-файлу ***Client JKS keystore path***	--
Client JKS truststore path	Путь к truststore-файлу в формате JKS для клиентской части ADB Control (агентов ADB Control/ADBM). Например, /opt/ssl/truststore.jks	--
Client JKS truststore password	Пароль для доступа к truststore-файлу ***Client JKS truststore path***	--



Заполнение параметров SSL

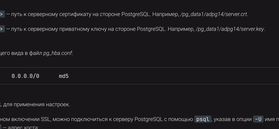
Рекомендации по заполнению

- Файлы adcc_ca.crt, adcc.crt и adcc.key автоматически создаются на хосте, где развернут сервис ADB Control. По умолчанию они расположены в директории /opt/adcc/ssl. Эти файлы можно использовать исключительно для целей тестирования.
- В production системах необходим доверенный центр сертификатов (Certificate Authority, CA) для подписи сертификатов. Для создания и регистрации SSL-сертификатов можно использовать утилиту `openssl`.
- В случае развертывания кластера ADB необходимо перегенерировать сертификаты, чтобы добавить соответствующие `ServerAltNames` для новых хостов.
- При подключении внешнего кластера ADB для мониторинга в ADB Control, где настроено использование SSL, необходимо указать сертификаты с `ServerAltNames` хостов обоих кластеров в настройках основного и внешнего кластера ADB.
- Для открытия пароля truststore в Vault присвойте параметру ***Use PEM CA for vault*** значение `true` в разделе ***Vault integration*** кластерных настроек ADB.
- В поле ***CA certificate*** для корректного формирования кластера необходимо указать цепочки сертификатов из корневого и всех промежуточных CA. Порядок CA-сертификатов в PEM-формате в цепочке сертификатов должен быть следующим:
  - Sub CA Certificate — начало.
  - Root CA Certificate — конец.

  Это также применимо к кастомным JKS, создаваемым с помощью `keytool`.

  В случае использования ADB Control как внешней системы убедитесь, что клиентский truststore содержит CA с сервера.

5. Нажмите ***Save***
6. Примените действие ***Reconfigure & Restart*** к сервису ADB Control.



Применение изменений

После успешного выполнения описанных выше шагов порт сервера ADB Control (по умолчанию `81`) будет использоваться для SSL-соединения. Адрес подключения будет выглядеть следующим образом: *https://<ADB Control IP address>:8*.

Внешние базы данных

Начиная с версии ADB Control 4.11.0 в случае включения SSL, описанным выше способом SSL-шифрование автоматически используется при подключении к внутренним базам данных ADB Control и ADBM. Если в ADB Control используется внешние базы данных Metrics DB (ClickHouse), Query DB (PostgreSQL) или Offload DB (PostgreSQL) и/или в ADBM используется внешняя PostgreSQL, необходимо дополнительно выполнить настройку SSL шифрования на их стороне:

PostgreSQL

1. Укажите следующие параметры в файле *postgresql.conf*:

```
ssl = on
ssl_cert_file = '<server_crt_path>'
ssl_key_file = '<server_key_path>'
```

   где:

   - `<server_crt_path>` — путь к серверному сертификату на стороне PostgreSQL. Например, /pg_data1/adpg14/server.crt.
   - `<server_key_path>` — путь к серверному приватному ключу на стороне PostgreSQL. Например, /pg_data1/adpg14/server.key.

2. Добавьте строку следующего вида в файл *pg_hba.conf*:

```
hostssl    all   all      0.0.0.0/0     md5
```

3. Перезагрузите PostgreSQL для применения настроек.
4. Чтобы убедиться в успешности включения SSL, можно подключиться к серверу PostgreSQL с помощью `psql`, указав в опции `-U` имя пользователя, в опции `-d` – название БД, в опции `-h` – адрес хоста:

```
$ psql -U adcc -d metrics -h 10.92.41.125
```

   После ввода пароля пользователя на экране должна отобразиться следующая информация:

```
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
```

5. Выполните стандартные действия по настройке внешней базы данных PostgreSQL в ADB Control или ADBM. Для настройки Offload DB обратитесь к разделу Настройка политик запуска задач — Metrics offload.

ВАЖНО

- Сертификат и ключ, указываемые в полях `<server_crt_path>` и `<server_key_path>` соответственно, необходимо сгенерировать самостоятельно либо воспользоваться теми, что формируются на стороне ADB Control (см. adcc.crt и adcc.key в разделе ADB Control). Но важно, чтобы в секции **SSL** для сервиса ADB Control были указаны те же сертификаты, что на стороне PostgreSQL в файле *postgresql.conf*.
- В текущей реализации не предусмотрена аутентификация по клиентскому сертификату, обеспечивается только SSL шифрование трафика. По этой причине в файле *pg_hba.conf* используется тип аутентификации `md5`, а не `cert`, а также не требуется ввод сертификата CA `ssl_ca_file` в файле *postgresql.conf*.
- Если в качестве внешней БД используется продукт ADPG, настройку SSL шифрования в нем (включая формирование сертификатов) можно выполнить согласно инструкции Включение SSL шифрования.

ClickHouse

С подробной инструкцией по настройке SSL можно ознакомиться в документации ClickHouse. Ниже приведены минимальные настройки:

1. Внесите следующие изменения в серверный конфигурационный файл config.xml на каждой ноде кластера ClickHouse:

   - Сделайте доступными HTTPS-порты и отключите HTTP-порты:

```
<https_port>8443</https_port>
<!--<http_port>8123</http_port>-->
<interserver_https_port>9010</interserver_https_port>
<!--<interserver_http_port>9009</interserver_http_port>-->
```

   - Настройте секцию `OpenSSL` следующим образом:

```
<openSSL>
    <server>
        <certificateFile>/etc/clickhouse-server/certs/server.crt</certificateFile>
        <privateKeyFile>/etc/clickhouse-server/certs/server.key</privateKeyFile>
        <verificationMode>relaxed</verificationMode>
        <caConfig>/etc/clickhouse-server/certs/ca.crt</caConfig>
        <cacheSessions>true</cacheSessions>
        <disableProtocols>sslv2,sslv3</disableProtocols>
        <preferServerCiphers>true</preferServerCiphers>
    </server>
    <client>
        <loadDefaultCAFile>false</loadDefaultCAFile>
        <caConfig>/etc/clickhouse-server/certs/ca.crt</caConfig>
        <cacheSessions>true</cacheSessions>
        <disableProtocols>sslv2,sslv3</disableProtocols>
        <preferServerCiphers>true</preferServerCiphers>
        <verificationMode>relaxed</verificationMode>
        <invalidCertificateHandler>
            <name>RejectCertificateHandler</name>
        </invalidCertificateHandler>
    </client>
</openSSL>
```

   где используются следующие пути, которые могут отличаться в вашем окружении:

   - /etc/clickhouse-server/certs/server.crt — путь к серверному сертификату на стороне ClickHouse.
   - /etc/clickhouse-server/certs/server.key — путь к серверному приватному ключу на стороне ClickHouse.
   - /etc/clickhouse-server/certs/ca.crt — путь к сертификату центра сертификации (Certificate Authority, CA) на стороне ClickHouse.

2. Перезагрузите ClickHouse для применения настроек.
3. Выполните стандартные действия по настройке внешней базы данных ClickHouse в ADB Control.

ВАЖНО

Сертификат и ключ, указываемые в полях `<certificateFile>` и `<privateKeyFile>`, соответственно, необходимо сгенерировать самостоятельно либо воспользоваться теми, что формируются на стороне ADB Control (см. adcc.crt и adcc.key в разделе ADB Control). Но важно, чтобы в секции **SSL** для сервиса ADB Control были указаны те же сертификаты, что на стороне ClickHouse в файле *config.xml*.

Настройка полетов запуска задач

Содержание

Docs cleanup
Metrics offload
Настройка
Сейчас выполняется запуск

Раздел / подраздел

Docs cleanup

Настройка **Сейчас выполняется запуск** — **Docs cleanup**. Задача для автоматической чистки устаревших данных. Удаляет старые записи из таблиц `Docs` и `Docs_Versions`, которые уже не нужны для работы системы.

Настройка **Сейчас выполняется запуск** — **Docs cleanup**. Задача для автоматической чистки устаревших данных. Удаляет старые записи из таблиц `Docs` и `Docs_Versions`, которые уже не нужны для работы системы.

Настройка **Сейчас выполняется запуск** — **Docs cleanup**

Поле	Описание	Допустимые значения
Name task	Имя задачи	string
Task type	Тип задачи. Для Docs cleanup — всегда `Docs cleanup`	`Docs cleanup`
Start date	Дата и время первого запуска задачи. Формат: `YYYY-MM-DD HH:MM:SS`. Для немедленного запуска — `now`	`now`, `YYYY-MM-DD HH:MM:SS`
Period	Период повторения задачи. Допустимые значения: `Minutes`, `Hours`, `Days`, `Weeks`, `Months`, `Years`	`Minutes`, `Hours`, `Days`, `Weeks`, `Months`, `Years`
Interval	Интервал повторения. Например, при `Period = Days` и `Interval = 1` задача запускается каждый день	`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`, `11`, `12`, `13`, `14`, `15`, `16`, `17`, `18`, `19`, `20`, `21`, `22`, `23`, `24`, `25`, `26`, `27`, `28`, `29`, `30`, `31`, `32`, `33`, `34`, `35`, `36`, `37`, `38`, `39`, `40`, `41`, `42`, `43`, `44`, `45`, `46`, `47`, `48`, `49`, `50`, `51`, `52`, `53`, `54`, `55`, `56`, `57`, `58`, `59`, `60`
Days of the week	Дни недели, в которые задача может выполняться. Для `Period = Weeks`	`Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, `Saturday`, `Sunday`
Days of the month	Дни месяца, в которые задача может выполняться. Для `Period = Months`	`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`, `11`, `12`, `13`, `14`, `15`, `16`, `17`, `18`, `19`, `20`, `21`, `22`, `23`, `24`, `25`, `26`, `27`, `28`, `29`, `30`, `31`, `Last day of month`

Настройка **Сейчас выполняется запуск** — **Docs cleanup**. Задача для автоматической чистки устаревших данных. Удаляет старые записи из таблиц `Docs` и `Docs_Versions`, которые уже не нужны для работы системы.

Metrics offload

Настройка

Настройка **Сейчас выполняется запуск** — **Metrics offload**. Задача для выгрузки метрик из таблицы `Metrics` в внешнюю систему мониторинга. Поддерживает два способа выгрузки: по HTTP API и через файлы в формате Prometheus.

Настройка **Сейчас выполняется запуск** — **Metrics offload**. Задача для выгрузки метрик из таблицы `Metrics` в внешнюю систему мониторинга. Поддерживает два способа выгрузки: по HTTP API и через файлы в формате Prometheus.

Шаги

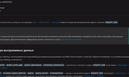
- **CONFIGURE** — Создание или изменение конфигурации выгрузки
- **DELETE** — Удаление конфигурации выгрузки

Статусы

- **SUCCESS** — Задача выполнена успешно
- **FAILED** — Задача завершилась с ошибкой

Значения поля Type

- **CONFIGURE** — Создание или изменение конфигурации выгрузки
- **DELETE** — Удаление конфигурации выгрузки



Настройка **Сейчас выполняется запуск** — **Metrics offload**

Поле	Описание	Допустимые значения
Name task	Имя задачи	string
Task type	Тип задачи. Для Metrics offload — всегда `Metrics offload`	`Metrics offload`
Start date	Дата и время первого запуска задачи. Формат: `YYYY-MM-DD HH:MM:SS`. Для немедленного запуска — `now`	`now`, `YYYY-MM-DD HH:MM:SS`
Period	Период повторения задачи. Допустимые значения: `Minutes`, `Hours`, `Days`, `Weeks`, `Months`, `Years`	`Minutes`, `Hours`, `Days`, `Weeks`, `Months`, `Years`
Interval	Интервал повторения. Например, при `Period = Days` и `Interval = 1` задача запускается каждый день	`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`, `11`, `12`, `13`, `14`, `15`, `16`, `17`, `18`, `19`, `20`, `21`, `22`, `23`, `24`, `25`, `26`, `27`, `28`, `29`, `30`, `31`, `32`, `33`, `34`, `35`, `36`, `37`, `38`, `39`, `40`, `41`, `42`, `43`, `44`, `45`, `46`, `47`, `48`, `49`, `50`, `51`, `52`, `53`, `54`, `55`, `56`, `57`, `58`, `59`, `60`
Days of the week	Дни недели, в которые задача может выполняться. Для `Period = Weeks`	`Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, `Saturday`, `Sunday`
Days of the month	Дни месяца, в которые задача может выполняться. Для `Period = Months`	`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`, `11`, `12`, `13`, `14`, `15`, `16`, `17`, `18`, `19`, `20`, `21`, `22`, `23`, `24`, `25`, `26`, `27`, `28`, `29`, `30`, `31`, `Last day of month`

Настройка **Сейчас выполняется запуск** — **Metrics offload**. Задача для выгрузки метрик из таблицы `Metrics` в внешнюю систему мониторинга. Поддерживает два способа выгрузки: по HTTP API и через файлы в формате Prometheus.

Управление кластерами

А. Ибрагимов

Содержание

Назад к содержанию

- Просмотр списка кластеров
- Подключение нового кластера
- Редактирование подключения к кластеру
- Установка кластера по умолчанию
- Архивация кластера
- Настройка мониторинга выбранного кластера
  - Databases monitoring
  - GUC management
  - Metric offload

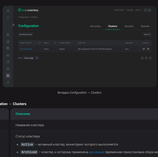
Управление кластерами в ADB Control.

В ADB Control есть возможность мониторинга множества кластеров ADB одновременно. При этом отображается только один из кластеров в выбранный пользователем. ADB Control, в зависимости, которым будет отправлять и для каждого кластера, индивидуальные данные, например о здоровье ADB Control.

Управление кластерами доступно в разделе меню Configuration, в подразделе Clusters.

Просмотр списка кластеров

Вкладка **Configuration** → **Clusters** выводит следующую информацию о кластерах:



Вкладка Configuration → Clusters

Поля вкладки Configuration → Clusters

Поле	Описание
Cluster name	Название кластера
Status	Статус кластера - **Active** - активный кластер, мониторинг которого выполняется - **Inactive** - кластер отключен, мониторинг выполняться не будет
Default cluster	Для кластера, используемого по умолчанию, выводится текст **Default**. Для прочих кластеров - **No**
JDBC URL	URL JDBC соединения с кластером
Username	Имя пользователя, используемого для соединения с кластером

Управление кластерами доступно для пользователя с ролью admin. Для остальных пользователей доступен только просмотр, для роли operator доступна только кнопка Refresh, для роли viewer - ничего. Для роли operator доступна кнопка **Refresh**, а фильтры можно выбрать, значения не изменяются. Для роли viewer - ничего. Для роли operator доступна кнопка **Cluster name** - клик по названию кластера приводит к переходу на вкладку Monitoring.

Кнопка **Refresh** - аналогичен, что для роли оператора. Фильтры на фильтры доступны **Refresh**.

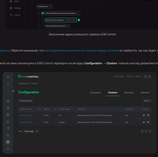
Подключение нового кластера

ВАЖНО!

- Для подсоединения ADB должны быть развернуты и запущены ADB Control. Зарегистрировать несколько кластеров с одним именем (например, создавать в разных ADB Control).
- При подключении кластера в ADB по первым клавиш следует обновить, тот кластер, ADB, где установлен основной сервер ADB Control (см. выполнить).

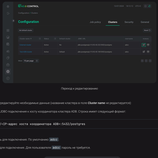
Подключение нового кластера ADB к ADB Control выполняется следующим образом:

1. Используя ADB, добавьте сервер ADB Control в кластер, который необходимо подключить к существующей системе мониторинга.
2. На сервере ADB Control выполнить следующие действия:
   - Перейдите в переменную **Control Cluster** для кластера.
   - В поле **ADB Control IP address** укажите IP-адрес хоста, где развернут целевой сервер ADB Control, web-интерфейс которого планируется использовать для мониторинга нового кластера.
   - В поле **External ADB Control port** укажите номер порта, по которому доступен целевой сервер ADB Control. Значение по умолчанию - **80**.
   - Нажмите **Save**.



Регистрация кластера в мониторинге ADB Control

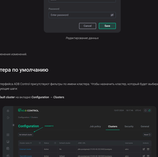
3. Выполнить установку кластера. Обратите внимание, что для выполнения скрипта установки кластера потребуется, так как будет использоваться внешний ADB Control.
4. В веб-интерфейсе целевой системы мониторинга ADB Control перейдите на вкладку **Configuration** → **Clusters**. Новый кластер добавляется в список автоматически.



Кластер добавлен

5. Для того чтобы убедиться, что подключение кластера выполнено успешно, перейдите на вкладку **Monitoring** → **Clusters**. Новый кластер добавляется в список, выбранным уровнем для мониторинга, в противном случае в выпадающем меню отсутствует. Выберите **Monitoring** → **Clusters**, и отображается кластер, выбираем предварительно выбрать в фильтрах по параметрам или по значению - **Cluster name** и **Cluster**. Полная таблица кластеров:

```
SELECT pg_clone(dbid);
```

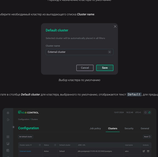


Кластер доступен для мониторинга

Редактирование подключения к кластеру

Для редактирования данных для добавленного кластера выполняется следующая цепочка:

1. Нажмите на иконку в столбце **Actions** на вкладке **Configuration** → **Clusters**

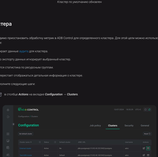


Переход к редактированию

2. В открывшемся окне отредактируйте необходимые данные (название кластера в поле **Cluster name** не редактируется):
   - **JDBC URL** - строка JDBC подключения к кластеру ADB. Пример строки JDBC ADB Control - следующий формат:

```
jdbc:postgresql://IP:port host=postgres DB=DB2/pgbfpro
```

   - **User** - пользователь для подключения. Это значение - **gpadmin**.
   - **Password** - пароль для подключения. Для пользователя **gpadmin** пароль не требуется.



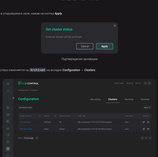
Редактирование данных

3. Нажмите **Save**.

Установка кластера по умолчанию

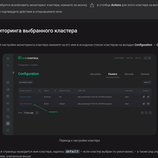
Для каждого пользователя ADB Control выполняется фильтр по умолчанию кластера. Чтобы изменить значение, которое будет выбрано в этом фильтре по умолчанию, выполняется следующая цепочка:

1. Нажмите иконку **Set default cluster** на вкладке **Configuration** → **Clusters**



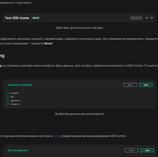
Переход к изменению статуса кластера

2. В открывшемся окне выберите необходимый для него из выпадающего списка **Cluster name**



Выбор кластера по умолчанию

3. Нажмите **Save**. В результате в столбце **Default cluster** для кластера, выбранного по умолчанию, отображается текст **Default**, для прочих кластеров - пусто. Для переходящего дефолтного кластера - пусто.



Кластер по умолчанию обновлен

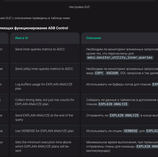
Архивация кластера

В некоторых случаях необходимо прекратить обработку метрик в ADB Control для определенного кластера. Для этой цели можно использовать **архивация** кластера, которая позволяет следующее:

- ADB Control полностью не собирает данные по кластеру для метрик.
- Системные задачи по чистке данных и создания исходных выборочных метрик.
- Для кластера не отображается статистика по загрузке кластера.
- Но страница **Information** продолжает отображать информацию о кластере.

Для архивации кластера выполняется следующая цепочка:

1. Нажмите на иконку в столбце **Actions** на вкладке **Configuration** → **Clusters**



Переход к архивации кластера

2. Подтвердите операцию в появившемся окне, нажав на кнопку **Apply**

Подтверждение архивации

В результате статус кластера изменится на **Archivated** на вкладке **Configuration** → **Clusters**

Обновленный статус кластера отображается

СТРАНЕ НОЖЕЙ

Если в дальнейшем потребуется возобновить мониторинг кластера, нажмите на иконку в столбце **Actions** для этого кластера на вкладке **Configuration** → **Clusters** и подтверждение действия в появившемся окне.

Настройка мониторинга выбранного кластера

Для перехода к расширенной настройке мониторинга кластера нажмите на его имя в каждом списке кластеров на вкладке **Configuration** → **Clusters**

Переход к настройке кластера

В верхней части отображаются 2 страницы выбранного кластера, первая - **DATLIST** - пусть пустую выбрано по умолчанию, и 1 пункт меню отображающих элементы для быстрого перехода к другим пунктам, отображаемый формой:

- **Databases monitoring** - для перехода к выбору для мониторинга.
- **GUC management** - для перехода к настройке.
- **Metric offload** - для перехода к выгрузке.

Для перехода к другим страницам GUC и отображение страницы и таблицы метрик:

- **Refresh** - для перехода к выгрузке.
- **Refresh** - для перехода к отображению.

Для перехода к другим страницам GUC и отображение страницы и таблицы метрик:

- **Refresh** - для перехода к выгрузке.
- **Refresh** - для перехода к отображению.

Databases monitoring

Для выбора баз данных для мониторинга нажмите на иконку в столбце **Actions** на вкладке **Configuration** → **Clusters**

Выбор баз данных для мониторинга

GUC management

В меню **GUC management** на странице этой страницы настроек, GUC, определяющие функционирование ADB Control:

Настройка GUC

Доступные для изменения параметры GUC и описанием приведены в таблице ниже:

Перечень GUC, определяющие функционирование ADB Control

Имя GUC	Имя в UI	Описание
adcc.monitor_enter_quantiles	Send enter quantiles metrics to ADCC	Необходимо ли передавать квантили выборок для всех метрик, которые ADB Control отправляет в ADCC
adcc.monitor_utility_stmt_quantiles	Send utility stmt quantiles metrics to ADCC	Необходимо ли передавать квантили выборок для утилиты между этапы COPY, REFLOW, ETC. которые ADB Control
adcc.explain_log_buffers	Log buffers usage for EXPLAIN ANALYZE	Использовать ли буферы логов для плана EXPLAIN ANALYZE
adcc.explain_log_timing	Collect timing data, not just row counts for EXPLAIN ANALYZE	Собирать ли данные о времени и количестве в результате и метку плана для EXPLAIN ANALYZE
adcc.explain_log_analyze	Log EXPLAIN ANALYZE of the end of the query	Отправлять ли EXPLAIN ANALYZE в лог-вывод выполнения запросов
adcc.explain_log_verbose	Log VERBOSE for EXPLAIN ANALYZE plans	Использовать ли опцию VERBOSE для EXPLAIN ANALYZE
adcc.explain_log_min_dur ation	Sets the minimum execution time above which EXPLAIN ANALYZE plans will be logged	Минимальный время выполнения, при превышении которого будут отправлены планы для команды EXPLAIN ANALYZE. -1 - не логгировать

Metric offload

В меню **Metric offload** - страница кластера можно выбрать базы данных, для которых необходимо выгрузить метрики на внешнюю БД (которая настроена, для использования на сервере **metrics-offload**, см. в разделе «Metrics offload»). Для отображения на выбрать на форме в таблицу:

Выбор баз данных для выгрузки метрик

Настройка политик безопасности

Дарья Барышева

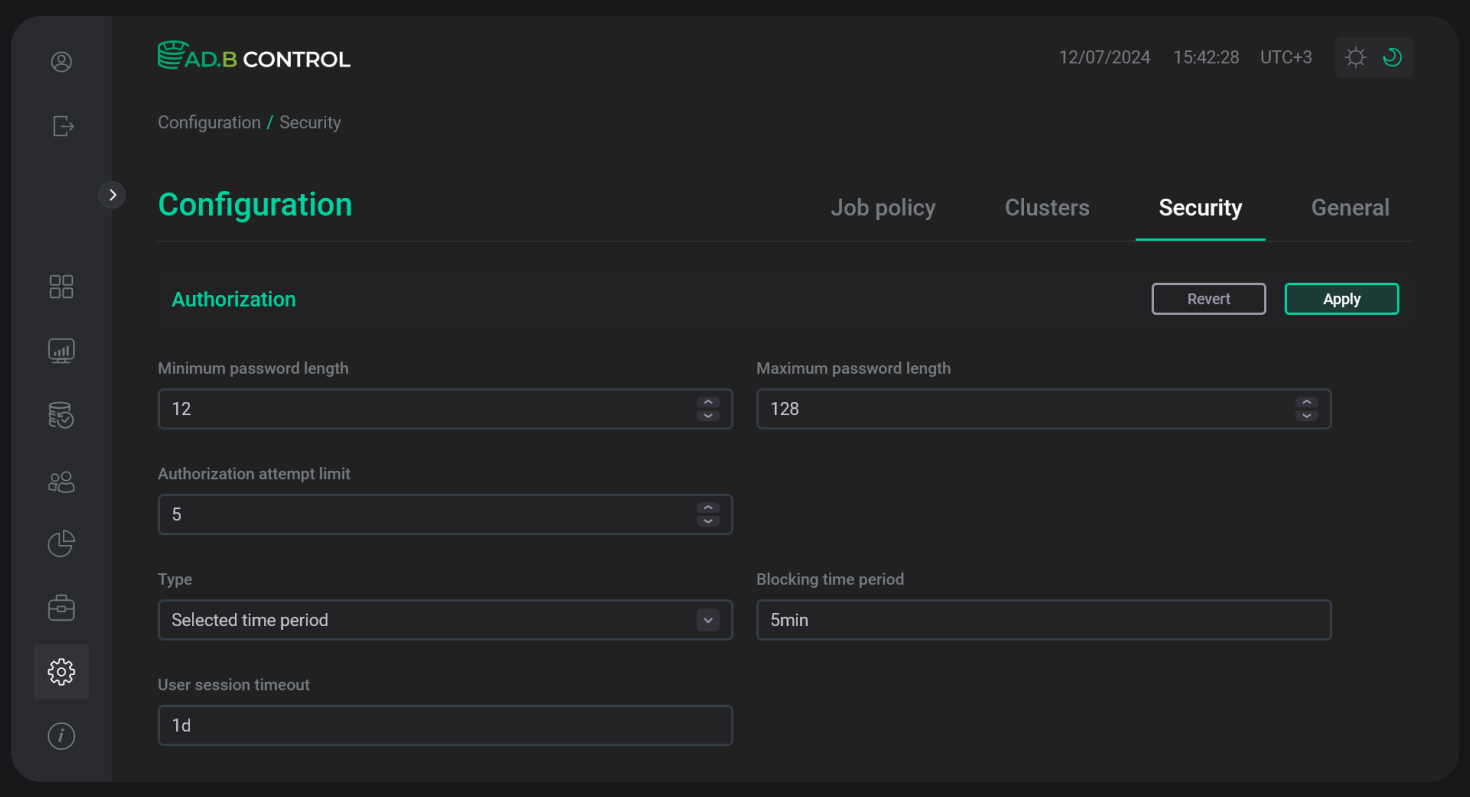
Назад к содержанию

Вкладка ***Security*** на странице ***Configuration*** в web-интерфейсе ADB Control предназначена для настройки политик безопасности. На ней можно определить следующие ограничения, которые будут проверяться при создании и авторизации пользователей.

Поля вкладки Configuration → Security

Поле	Описание	Значение по умолчанию
Minimum password length	Минимальная длина пароля, заполняемого при создании пользователя	12
Maximum password length	Максимальная длина пароля, заполняемого при создании пользователя	128
Authorization attempt limit	Максимальное количество неудачных попыток входа в систему, при достижении которого пользователь блокируется на постоянной или временной основе (в зависимости от значения поля ***Type***)	5
Type	Тип блокировки пользователя при достижении максимального числа неудачных попыток входа (указанного в поле ***Authorization attempt limit***). Возможные значения: ▪ `Permanently` — блокировка на постоянной основе (до разблокировки пользователя владельцем системы). ▪ `Selected time period` — временная блокировка в течение указанного в поле ***Blocking time period*** времени (либо до разблокировки пользователя владельцем системы).	Selected time period
Blocking time period	Период временной блокировки пользователя в формате `<value><unit>` (без пробела), где `<value>` — количество; `<unit>` — единица измерения, принимающая следующие возможные значения: ▪ `min` — минуты; ▪ `hr` — часы; ▪ `d` — дни. Примеры: `5min`, `1hr`, `1d`	5min
User session timeout	Время, по истечении которого сессии пользователей ADB Control необходимо прерывать. Заполняется в формате, аналогичном полю ***Blocking time period***	1d

Если необходимо изменить настройки по умолчанию — отредактируйте нужные поля и нажмите кнопку ***Apply***. Кнопка ***Revert*** предназначена для отмены внесенных в форме изменений (еще не сохраненных путем нажатия ***Apply***).



Вкладка Configuration → Security

Общие настройки

Дарья Барышева

Содержание

Назад к содержанию

General

Sessions

Metrics

Вкладка ***General*** на странице ***Configuration*** в web-интерфейсе ADB Control предназначена для ввода конфигурационных параметров общего назначения. Страница состоит из трех вкладок, каждая из которых детально описана ниже.

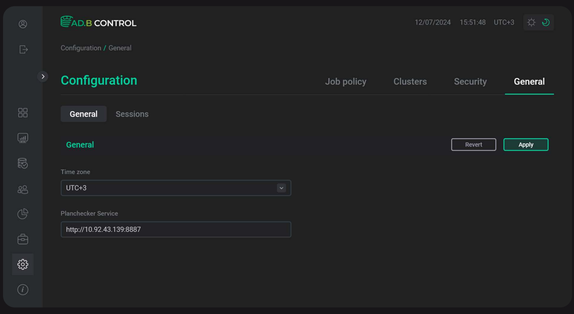
Если необходимо изменить настройки по умолчанию — отредактируйте нужные поля и нажмите кнопку ***Apply***. Кнопка ***Revert*** предназначена для отмены внесенных в форме изменений (еще не сохраненных путем нажатия ***Apply***).

General

На вкладке ***General*** можно настроить общие параметры.

Поля вкладки Configuration → General → General

Поле	Описание	Значение по умолчанию
Time zone	Часовой пояс пользователя	UTC+3
Planchecker service	Путь к сервису Planchecker. Параметр не применим в ADB 7	—



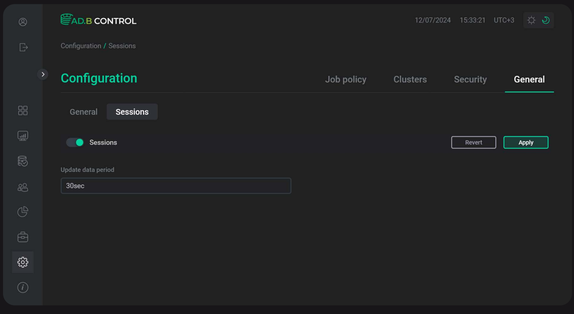
Вкладка Configuration → General → General

Sessions

Вкладка ***Sessions*** предназначена для настройки мониторинга сессий.

Поля вкладки Configuration → General → Sessions

Поле	Описание	Значение по умолчанию
Sessions	Переключатель, определяющий необходимость мониторинга сессий в ADB Control	True
Update data period	Периодичность получения информации о сессиях	30sec



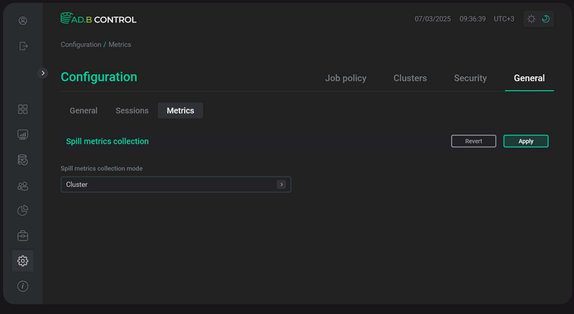
Вкладка Configuration → General → Sessions

Metrics

Вкладка ***Metrics*** предназначена для настройки сбора различных метрик. Вкладка доступна начиная с версии ADB Control 4.12.0.

Поля вкладки Configuration → General → Metrics

Поле	Описание	Значение по умолчанию
Spill metrics collection mode	Режим сбора метрик spill-файлов: ▪ `Cluster` — только по кластеру в целом; ▪ `Cluster and segment` — по кластеру и каждому сегменту. При выборе значения `Cluster and segment` на странице с детальной информацией о команде в разделе Performance будут заполняться поля с метриками ***Spill size*** и ***Spill files number*** по каждому сегменту	Cluster



Вкладка Configuration → General → Metrics

Dashboard

Дарья Барышева

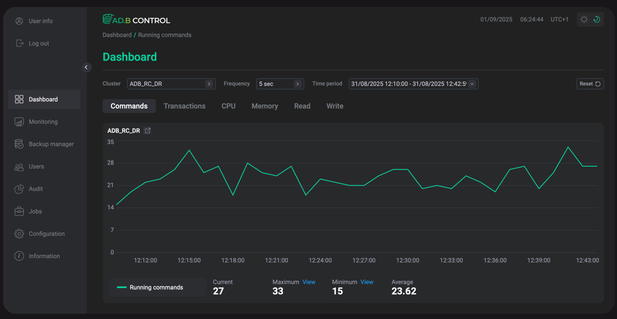
Назад к содержанию

Страница ***Dashboard*** в web-интерфейсе ADB Control отображает графики для отслеживания динамики изменения различных метрик за выбранный период времени в кластерах ADB, подключенных для мониторинга.

Каждая вкладка на странице ***Dashboard*** выводит отдельный вид графиков:

- ***Commands*** — число SQL-команд в статусе `Running`.
- ***Transactions*** — число транзакций в статусе `Active`.
- ***CPU*** — общий объем потребляемого CPU в кластере (в процентах).
- ***Memory*** — общий объем выделенной виртуальной памяти в кластере (в байтах).
- ***Read*** — общий объем прочитанных данных в кластере (в байтах).
- ***Write*** — общий объем записанных данных в кластере (в байтах).

Для перехода к полноэкранному режиму просмотра любого из графиков нажмите на иконку справа от имени кластера в верхней части графика.

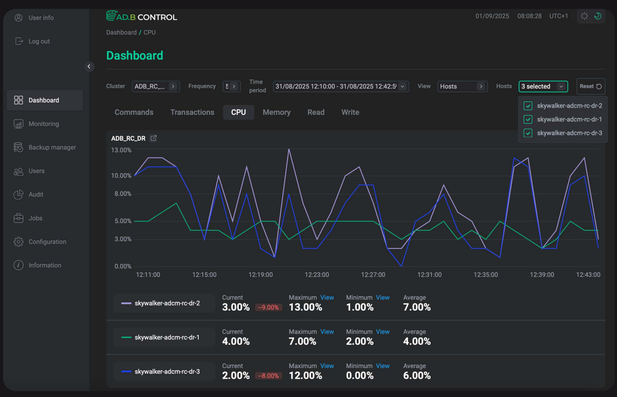


Страница Dashboard

Для конфигурирования графиков используются следующие поля в верхней части страницы:

- ***Cluster*** — кластер ADB, для которого строятся графики. По умолчанию выбран кластер, отмеченный как дефолтный в настройках ADB Control.
- ***Frequency*** — частота обновления информации на графиках.
- ***Time period*** — временной период, за который требуется построить график. При нажатии на поле открывается окно, в котором можно выбрать интервал из предложенных вариантов либо самостоятельно установить начальную и конечную границы временного диапазона.
- ***View*** — способ отображения графиков. Поле не выводится на вкладках ***Commands*** и ***Transactions***. Возможные значения:
  - `Summary` — вывод сводной информации в виде общего графика по всему кластеру в целом.
  - `Hosts` — вывод информации по каждому хосту кластера в виде отдельных графиков.
- ***Hosts*** — выпадающий список, в котором можно выбрать хосты кластера. Становится доступен, если для кластера выбран режим отображения `Hosts` в поле ***View***.

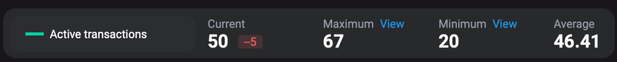
На рисунке ниже показан вариант отображения информации в разрезе хостов кластера.



Построение графиков в разрезе хостов

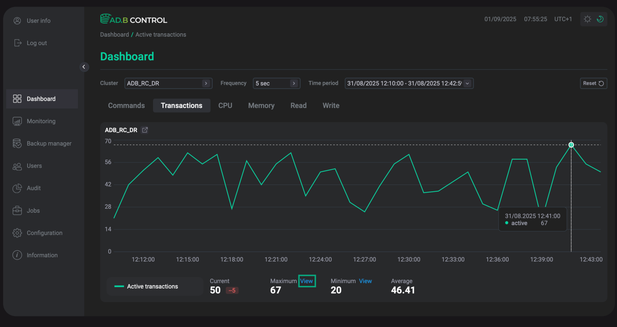
На каждой вкладке под графиками выводится следующая сводная информация:

- ***Current*** — последнее значение метрики за выбранный период времени. Справа от значения может выводиться специальный индикатор, показывающий разницу между прошлым и текущим значениями (в случае изменения).
- ***Maximum*** — максимальное значение метрики за выбранный период времени. Нажав ***View***, можно выделить соответствующую точку на графике.
- ***Minimum*** — минимальное значение метрики за выбранный период времени. Нажав ***View***, можно выделить соответствующую точку на графике.
- ***Average*** — среднее значение, рассчитанное на основе всех значений метрики, полученных за выбранный период времени.



Сводная информация по метрике

Ниже показано, как отобразить максимальное или минимальное значение метрики на графике, нажав ***View***.



Отображение максимального значения метрики на графике

При наведении курсора мыши на график можно получить значение отслеживаемой метрики для любой выбранной временной точки.

Мониторинг сокета

Реклама · Скрыть рекламу

Содержание

- Info
- Sockets
- Socket
- Socket Stats
- Socket Stats Summary
- Socket Stats Summary Statistics
- Socket Stats Summary Statistics Tcp
- Socket Stats Summary Statistics Udp
- Socket Stats Summary Statistics Raw
- Socket Stats Summary Statistics Frag
- Socket Stats Summary Statistics Total
- Socket Stats Summary Statistics Tcp6
- Socket Stats Summary Statistics Udp6
- Socket Stats Summary Statistics Raw6
- Socket Stats Summary Statistics Frag6
- Socket Stats Summary Statistics Total6
- Socket Stats Summary Statistics Tcp Tw
- Socket Stats Summary Statistics Tcp Orphan
- Socket Stats Summary Statistics Tcp Mem
- Socket Stats Summary Statistics Tcp Mem6
- Socket Stats Summary Statistics Tcp Mem Kb
- Socket Stats Summary Statistics Tcp Mem Kb6

Info

Info — Информация о сокетах.

Sockets

Sockets — Сокеты.

Socket

Socket — Сокет.

Socket Stats

Socket Stats — Статистика сокета.

Socket Stats Summary

Socket Stats Summary — Сводка статистики сокета.

Socket Stats Summary Statistics

Socket Stats Summary Statistics — Статистика сводки статистики сокета.

Socket Stats Summary Statistics Tcp

Socket Stats Summary Statistics Tcp — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Udp

Socket Stats Summary Statistics Udp — Статистика UDP сводки статистики сокета.

Socket Stats Summary Statistics Raw

Socket Stats Summary Statistics Raw — Статистика RAW сводки статистики сокета.

Socket Stats Summary Statistics Frag

Socket Stats Summary Statistics Frag — Статистика фрагментов сводки статистики сокета.

Socket Stats Summary Statistics Total

Socket Stats Summary Statistics Total — Общая статистика сводки статистики сокета.

Socket Stats Summary Statistics Tcp6

Socket Stats Summary Statistics Tcp6 — Статистика TCP6 сводки статистики сокета.

Socket Stats Summary Statistics Udp6

Socket Stats Summary Statistics Udp6 — Статистика UDP6 сводки статистики сокета.

Socket Stats Summary Statistics Raw6

Socket Stats Summary Statistics Raw6 — Статистика RAW6 сводки статистики сокета.

Socket Stats Summary Statistics Frag6

Socket Stats Summary Statistics Frag6 — Статистика фрагментов сводки статистики сокета.

Socket Stats Summary Statistics Total6

Socket Stats Summary Statistics Total6 — Общая статистика сводки статистики сокета.

Socket Stats Summary Statistics Tcp Tw

Socket Stats Summary Statistics Tcp Tw — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Orphan

Socket Stats Summary Statistics Tcp Orphan — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem

Socket Stats Summary Statistics Tcp Mem — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem6

Socket Stats Summary Statistics Tcp Mem6 — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem Kb

Socket Stats Summary Statistics Tcp Mem Kb — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem Kb6

Socket Stats Summary Statistics Tcp Mem Kb6 — Статистика TCP сводки статистики сокета.

Info

Info — Информация о сокетах.

Sockets

Sockets — Сокеты.

Socket

Socket — Сокет.

Socket Stats

Socket Stats — Статистика сокета.

Socket Stats Summary

Socket Stats Summary — Сводка статистики сокета.

Socket Stats Summary Statistics

Socket Stats Summary Statistics — Статистика сводки статистики сокета.

Socket Stats Summary Statistics Tcp

Socket Stats Summary Statistics Tcp — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Udp

Socket Stats Summary Statistics Udp — Статистика UDP сводки статистики сокета.

Socket Stats Summary Statistics Raw

Socket Stats Summary Statistics Raw — Статистика RAW сводки статистики сокета.

Socket Stats Summary Statistics Frag

Socket Stats Summary Statistics Frag — Статистика фрагментов сводки статистики сокета.

Socket Stats Summary Statistics Total

Socket Stats Summary Statistics Total — Общая статистика сводки статистики сокета.

Socket Stats Summary Statistics Tcp6

Socket Stats Summary Statistics Tcp6 — Статистика TCP6 сводки статистики сокета.

Socket Stats Summary Statistics Udp6

Socket Stats Summary Statistics Udp6 — Статистика UDP6 сводки статистики сокета.

Socket Stats Summary Statistics Raw6

Socket Stats Summary Statistics Raw6 — Статистика RAW6 сводки статистики сокета.

Socket Stats Summary Statistics Frag6

Socket Stats Summary Statistics Frag6 — Статистика фрагментов сводки статистики сокета.

Socket Stats Summary Statistics Total6

Socket Stats Summary Statistics Total6 — Общая статистика сводки статистики сокета.

Socket Stats Summary Statistics Tcp Tw

Socket Stats Summary Statistics Tcp Tw — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Orphan

Socket Stats Summary Statistics Tcp Orphan — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem

Socket Stats Summary Statistics Tcp Mem — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem6

Socket Stats Summary Statistics Tcp Mem6 — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem Kb

Socket Stats Summary Statistics Tcp Mem Kb — Статистика TCP сводки статистики сокета.

Socket Stats Summary Statistics Tcp Mem Kb6

Socket Stats Summary Statistics Tcp Mem Kb6 — Статистика TCP сводки статистики сокета.

Info

Info — Информация о сокетах.

Info	Info
Sockets	Sockets — Сокеты.
Socket	Socket — Сокет.
Socket Stats	Socket Stats — Статистика сокета.
Socket Stats Summary	Socket Stats Summary — Сводка статистики сокета.
Socket Stats Summary Statistics	Socket Stats Summary Statistics — Статистика сводки статистики сокета.
Socket Stats Summary Statistics Tcp	Socket Stats Summary Statistics Tcp — Статистика TCP сводки статистики сокета.
Socket Stats Summary Statistics Udp	Socket Stats Summary Statistics Udp — Статистика UDP сводки статистики сокета.
Socket Stats Summary Statistics Raw	Socket Stats Summary Statistics Raw — Статистика RAW сводки статистики сокета.
Socket Stats Summary Statistics Frag	Socket Stats Summary Statistics Frag — Статистика фрагментов сводки статистики сокета.
Socket Stats Summary Statistics Total	Socket Stats Summary Statistics Total — Общая статистика сводки статистики сокета.
Socket Stats Summary Statistics Tcp6	Socket Stats Summary Statistics Tcp6 — Статистика TCP6 сводки статистики сокета.
Socket Stats Summary Statistics Udp6	Socket Stats Summary Statistics Udp6 — Статистика UDP6 сводки статистики сокета.
Socket Stats Summary Statistics Raw6	Socket Stats Summary Statistics Raw6 — Статистика RAW6 сводки статистики сокета.
Socket Stats Summary Statistics Frag6	Socket Stats Summary Statistics Frag6 — Статистика фрагментов сводки статистики сокета.
Socket Stats Summary Statistics Total6	Socket Stats Summary Statistics Total6 — Общая статистика сводки статистики сокета.
Socket Stats Summary Statistics Tcp Tw	Socket Stats Summary Statistics Tcp Tw — Статистика TCP сводки статистики сокета.
Socket Stats Summary Statistics Tcp Orphan	Socket Stats Summary Statistics Tcp Orphan — Статистика TCP сводки статистики сокета.
Socket Stats Summary Statistics Tcp Mem	Socket Stats Summary Statistics Tcp Mem — Статистика TCP сводки статистики сокета.
Socket Stats Summary Statistics Tcp Mem6	Socket Stats Summary Statistics Tcp Mem6 — Статистика TCP сводки статистики сокета.
Socket Stats Summary Statistics Tcp Mem Kb	Socket Stats Summary Statistics Tcp Mem Kb — Статистика TCP сводки статистики сокета.
Socket Stats Summary Statistics Tcp Mem Kb6	Socket Stats Summary Statistics Tcp Mem Kb6 — Статистика TCP сводки статистики сокета.

Мониторинг транзакций

Ю. Ковальченко

Содержание

- Содержание
- Online
  - Назад
- Детали транзакции
  - Заголовок
  - Overview
  - Performance
  - Рабочая нагрузка
  - Commands
- Метрики, собираемые для транзакций

Назад к содержанию

Страница **Monitoring** → **Transactions** → **Online** позволяет контролировать для каждой транзакции, которая выполняется в настоящее время, характеристики для каждого сервера в разрезе вкладок **Online** → **History** за последние сутки.

Список транзакций

На странице **Monitoring** → **Transactions** → **Online** выводится транзакции, которые выполняются в текущий момент времени по пулам **SUTUAT**.

В верхней части выводится статистика по выполняющимся транзакциям общая по всем серверам, ниже по каждому серверу отдельно для каждого пула.

1 servers
Monitoring
Active Users
Sessions
Transactions
Transactions
Online
SUTUAT
Active
Idle
Total
Active
Idle
Total
Active
Idle
Total

Мониторинг сессий

Дарья Барышева

Содержание

Назад к содержанию

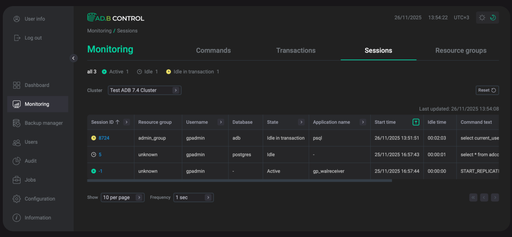
Просмотр списка сессий

Прерывание сессии

Вкладка ***Monitoring → Sessions*** предназначена для мониторинга клиентских сессий ADB — как активных, так и неактивных ("зависших"). Функционал страницы позволяет также прерывать выбранные сессии для освобождения пула соединений.

Просмотр списка сессий

На вкладке ***Monitoring → Sessions*** выводится следующая информация о сессиях.



Список сессий

Поля вкладки Monitoring → Sessions

Поле	Описание
Session ID	Идентификатор сессии
Resource group	Имя ресурсной группы
Username	Имя пользователя, выполнившего подключение
Database	Имя базы данных
State	Текущий статус сессии. Возможные значения: ▪ `Active` — запрос выполняется на стороне бэкенда. ▪ `Idle` — бэкенд ожидает новой команды со стороны клиента. ▪ `Idle in transaction` — на стороне бэкенда запущена транзакция, однако в текущий момент в ней нет активных команд.
Application name	Имя приложения, в рамках которого запущена сессия
Start time	Время начала процесса на стороне бэкенда
Idle time	Время неактивности сессии (в секундах)
Command text	Текст последней запущенной в рамках сессии команды. Если `state = Active`, в поле показывается текст текущей команды
Tags	Теги сессии

ПРИМЕЧАНИЕ

Чтобы перейти к просмотру деталей последней команды, запущенной в рамках текущей сессии, нажмите на идентификатор сессии (***Session ID***). Обратите внимание, что перейти к просмотру команды можно только в том случае, если она зарегистрирована в ADB Control. Узнать о необходимых для этого условиях можно из статьи Мониторинг команд.

В верхней части вкладки содержатся поля ***Active***, ***Idle*** и ***Idle in transaction***, показывающие общее число сессий в соответствующих статусах.

Над таблицей со списком сессий расположен фильтр ***Cluster***, в котором можно выбрать кластер ADB, для которого требуется вывести данные в таблице. По умолчанию выбран кластер, отмеченный как дефолтный в настройках ADB Control.

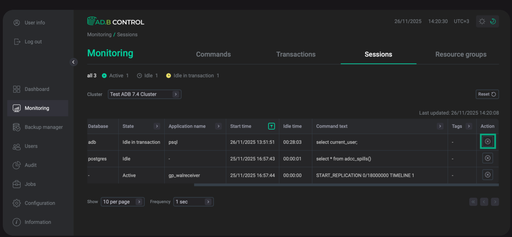
В заголовках столбцов таблицы со списком сессий, расположены фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо нажать на иконку > . Для тех столбцов, где набор возможных значений ограничен (например, ***State***), в фильтре можно выбрать значение из выпадающего списка. Для некоторых столбцов (например, ***Session ID***) требуется ввести искомое значение. Для столбцов, показывающих дату и время (например, ***Start time***), временной диапазон можно выбрать из календаря.

Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите ***Reset***.

Прерывание сессии

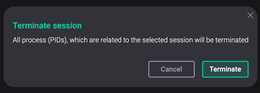
Чтобы прервать выполнение сессии, выполните следующие шаги:

1. Нажмите на иконку ⊗ в столбце ***Actions*** на вкладке ***Monitoring → Sessions***.



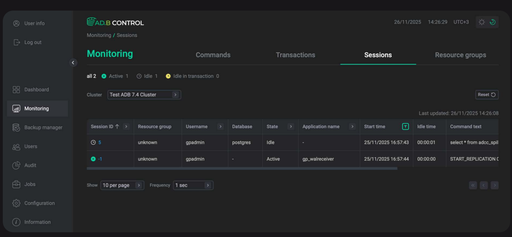
Переход к прерыванию сессии

2. Подтвердите операцию в открывшемся окне, нажав на кнопку ***Terminate***.



Подтверждения операции

В результате происходит прерывание клиентской сессии, и она более не отображается на вкладке ***Monitoring → Sessions***.



Данные о сессии более не отображаются

Работа с тегами запросов

Дарья Барышева

Назад к содержанию

Начиная с версии 4.7.5 в ADB Control поддерживается **тегирование запросов** — назначение SQL-запросам в рамках ADB-сессии либо транзакции кастомных метаданных в виде пар `<ключ>: <значение>` с последующим их отображением (с возможностью фильтрации) на страницах ***Monitoring → Commands***, ***Monitoring → Transactions*** и ***Monitoring → Sessions*** в web-интерфейсе ADB Control. Установка тегов осуществляется в ADB с помощью GUC `query_band`. Этот функционал может быть полезен для мониторинга групп запросов, объединенных, например, в рамках общего процесса ETL.

Ниже приведен пример работы с тегами запросов:

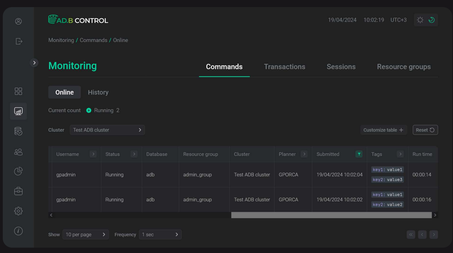
1. Откройте первую сессию в ADB (подключившись к базе данных `adb` на хосте координатора, например, через `psql`).
2. Установите значение двух тегов на уровне первой сессии:

```
SET query_band = 'key1: value1,key2: value2';
```

3. Запустите любой SQL-запрос, продолжительности которого будет достаточно для отображения в ADB Control.
4. Во второй сессии ADB установите иное значение тега `key2`:

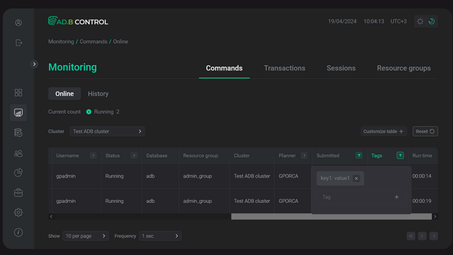
```
SET query_band = 'key1: value1,key2: value3';
```

5. Запустите любой продолжительный запрос во второй сессии.
6. Перейдите в web-интерфейс ADB Control на страницу ***Monitoring → Commands → Queries → Online***. Значения тегов двух запущенных запросов отображаются в столбце ***Tags***. Аналогичный столбец присутствует на страницах *Monitoring → Commands → Queries → History*, *Monitoring → Transactions* (*Online* и *History*), *Monitoring → Sessions* и *Monitoring → Resource groups → Statistics → <ресурсная группа> → Commands → Queueing*.



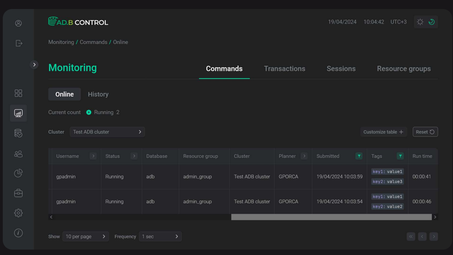
Теги в интерфейсе ADB Control

7. Чтобы отобразить на странице запросы с определенным значением одного из тегов, раскройте выпадающий список в заголовке столбца ***Tags***, укажите значение тега в поле ***Tag*** в формате `<ключ>: <значение>` и нажмите на иконку `+`. Если впоследствии потребуется удалить тег из фильтра, нажмите на иконку `×`.



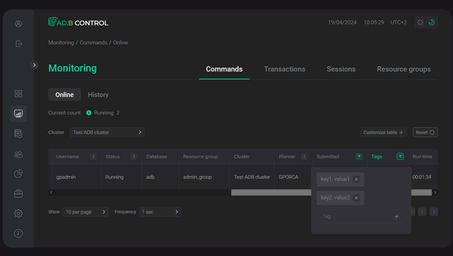
Установка значения первого тега в фильтре

В результате на странице остаются оба запроса, т.к. для обоих было указано значение `value1` для `key1`.



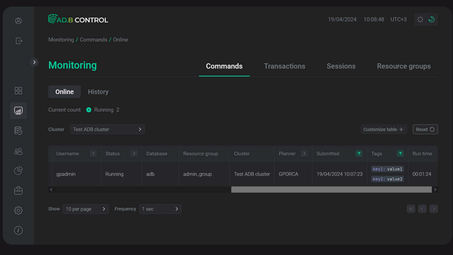
Результат фильтрации по значению одного тега

8. Ниже показано добавление фильтра по второму тегу. Значение каждого тега заполняется отдельно.



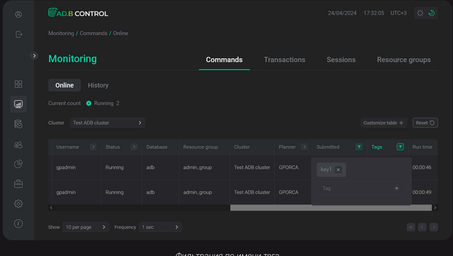
Установка значения второго тега в фильтре

В результате применения фильтра на странице остается только запрос из первой сессии.



Результат фильтрации по значениям двух тегов

9. Также вы можете фильтровать данные по имени тега, не выбирая его значение. В результате все запросы, для которых определен выбранный тег, будут отображены на странице – независимо от значения тега.



Фильтрация по имени тега

Аудит

Раздел **Аудит** собирает операции в ADB Control и позволяет отслеживать и расследовать изменения в системе.

Страница **Аудит** состоит из вкладок в ADB Control и вкладок в ADBM.

ADB Control

На вкладке **ADB Control** накапливаются следующие операции в ADB Control:

- Relations
- Operations
- Authorizations

Страница Аудит

Relations

Список отношений

Вкладка **ADB Control** → **Relations** → **Relations** Аудит показывает часто обращающиеся отношения в последовательности, отсортированной по последнему использованному времени. Для каждого отношения показывается, сколько раз оно было использовано, и когда последний раз оно было использовано. Выберите отношение, чтобы увидеть его детали, перейдите на вкладку Relations.

Вкладка Relations

Таблица отображается на рисунке выше, выводятся следующие информации:

Поля вкладки Relations

Поле	Описание
Database	Название базы данных
Schema	Название схемы
Relation	Название отношения
Last accessed	Дата и время последнего доступа к отношению в формате `DD.MM.YYYY HH:MM:SS`

Детали каждого обращения к выбранному отношению

Для каждого отношения в Relations показывается, сколько раз оно было использовано, и последнее время использования.

Вкладка Relations

Отображает поля следующих данных:

- **Time stamp** — время обращения;
- **Security self** — статистика обращений к выбранному отношению, которое пользователем ADB Control было сделано, указанной в фильтре **Select time period**.

Вкладка Relations

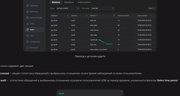
Отображает поля следующих данных:

Поля вкладки Time stamp

Поле	Описание
Today	Число обращений к отношению за текущие сутки
This week	Число обращений к отношению за последние недели
This month	Число обращений к отношению за последний месяц
Last accessed	Дата и время последнего доступа к отношению в формате `DD.MM.YYYY HH:MM:SS`

Поля вкладки Security self

Поле	Описание
Total	Общее число обращений пользователя к отношению за все время наблюдения
Today accessed	Число обращений пользователя к отношению в текущие сутки
Last accessed	Дата и время последнего доступа пользователя к отношению в формате `DD.MM.YYYY HH:MM:SS`
Total accessed	Общее число обращений пользователя к отношению за все время наблюдения



Вкладка Relations

Для каждого отношения показывается, сколько раз оно было использовано, и последнее время использования. Для каждого вида операции показывается:

- **Select count** — фактическое число выполнений `Select` для каждой таблицы;
- **Select time period** — фактическое время выполнения `Select` для каждой таблицы;
- **Last accessed** — последнее время использования таблицы.

Для каждого отношения показывается, сколько раз оно было использовано, и последнее время использования. Для каждого вида операции показывается:

- **Insert** — `INSERT`;
- **Update** — `UPDATE`;
- **Delete** — `DELETE`;
- **Select** — `SELECT`.

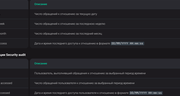
Вкладка Relations

Для каждого отношения показывается, сколько раз оно было использовано, и последнее время использования. Для каждого вида операции показывается:

- **Select count** — фактическое число выполнений `Select` для каждой таблицы;
- **Select time period** — фактическое время выполнения `Select` для каждой таблицы;
- **Last accessed** — последнее время использования таблицы.

Для каждого отношения показывается, сколько раз оно было использовано, и последнее время использования. Для каждого вида операции показывается:

- **Insert** — `INSERT`;
- **Update** — `UPDATE`;
- **Delete** — `DELETE`;
- **Select** — `SELECT`.



Вкладка Relations

Operations

Вкладка **ADB Control** → **Operations** → **Operations** показывает список операций пользователей в ADB Control за выбранный период времени. По умолчанию данные выводятся за последний час.

Вкладка Operations

Таблица отображается на рисунке выше, выводятся следующие информации:

Поля вкладки Operations

Поле	Описание
Object name	Название объекта
Object type	Объекты, над которыми осуществляется операция. Возможные значения: - `User` — пользователь; - `Role` — роль пользователя; - `Profile` — профиль пользователя, определяющий лимиты и ограничения на ресурсы; - `Tablespace` — табличное пространство; - `Schema` — схема; - `Table` — таблица; - `Index` — индекс; - `Sequence` — последовательность; - `Synonym` — синоним; - `Procedure` — процедура; - `Function` — функция; - `Package` — пакет; - `Trigger` — триггер; - `Materialized view` — материализованный просмотр; - `Materialized view log` — журнал материализованных представлений; - `Dblink` — база данных; - `Directory` — каталог; - `Cluster` — кластер; - `Type` — тип; - `Operator` — оператор; - `Java source` — исходный код Java; - `Java class` — класс Java; - `Java resource` — ресурс Java; - `Cluster relationship` — отношение между кластерами в ADB Control.
Operation type	Тип операции, которая была произведена с объектом. **Общие типы**: Возможные значения: - `Create` — создание; - `Change` — изменение; - `Drop` — удаление; - `Manual drop` — ручное удаление; - `Recycle` — помещение в корзину; - `Rename` — переименование; - `Grant` — выдача привилегии; - `Revoke` — отзыв привилегии; - `Autotask` — автозадача. **Только для ролей**: - `Set default role` — изменение роли по умолчанию; - `Set password` — изменение пароля; - `Change auth` — изменение способа аутентификации; - `Change expiry` — изменение срока действия пароля; - `Change lock` — изменение блокировки; - `Change password` — изменение пароля.
Result	Результат операции. Возможные значения: - `Successful` — операция завершена; - `Failed` — операция завершилась с ошибкой; - `Unknown` — статус неизвестен.
Operation time	Дата и время выполнения операции в формате `DD.MM.YYYY HH:MM:SS`
Client hostname	Имя пользователя ОС
Session	Идентификатор сеанса ADB Control, в рамках которого выполнена операция
Node	Имя узла, на котором выполнена операция

В правом углу **Open table** кнопку, нажмите чтобы открыть окно `Table` для просмотра над текущей строкой таблицы. Нажмите снова, чтобы свернуть окно `Table`.

Список операций пользователей

Authorizations

Вкладка **ADB Control** → **Authorizations** → **Authorizations** показывает список операций пользователей ADB Control за выбранный период времени. По умолчанию данные выводятся за последний час. Для отображения на странице операции, которые были совершены с ролями, нажмите на кнопку **Roles**. Для отображения на странице операций, которые были совершены с ролями, нажмите на кнопку **Roles**.

Вкладка Authorizations

Таблица отображается на рисунке выше, выводятся следующие информации:

Поля вкладки Authorizations

Поле	Описание
Operation time	Дата и время выполнения операции в формате `DD.MM.YYYY HH:MM:SS`
Client hostname	Имя пользователя ОС
Operation	Идентификатор операции в ADB Control
Message	Сообщение о статусе ошибки. Например, сообщение вида `ORA-01031:INSUFFICIENT PRIVILEGES` говорит о недостаточных привилегиях для выполнения операции. Для успешных операций — пустое
Sub type	Тип операции: - `ADD` — пользователь или ролью в ADB Control; - `MOD_COMP` — пользователь или роль в ADB Control; - `DEL_COMP` — пользователь или роль в ADB Control; - `ADD_GRANT` — выдача привилегии или роли в ADB Control; - `DEL_GRANT` — отзывание привилегии или роли в ADB Control.
Type	Тип объекта. Возможные значения: - `ADB` — вход в систему; - `Connect` — вход в ограниченном режиме; - `Internal maintenance` — внутренние задачи ядра системы.
Result	Результат операции. Возможные значения: - `Successful` — операция завершена; - `Failed` — операция завершилась с ошибкой.
Node	Имя узла, на котором выполнена операция. Столбцы для этого поля: `Node` и `Instance`. Нажмите на `Instance` для отображения на странице дополнительных столбцов.
Session Time	Дата и время выполнения операции в формате `DD.MM.YYYY HH:MM:SS`

Для отображения всего доступного, для каждого отношения, нажмите на кнопку **Customize table**. В появившемся окне выберите нужные поля и нажмите **Apply**. Чтобы вернуть обратно первоначальный вид, нажмите **Reset**. Для поиска нажмите **Find**. Для сохранения отчёта нажмите **Save**.

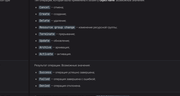
Настройка таблицы с данными

ADBM

Вкладка **ADBM** показывает информацию о кластерах ADB Control и о их группах и репликациях.

Clusters

Вкладка **ADBM** → **Clusters** показывает информацию о создании изменениях и удалении кластеров.



Вкладка Clusters

Поля вкладки Clusters

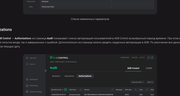
Поле	Описание
Object name	Название объекта
Object type	Объекты, над которыми осуществляется операция. Для всех видов **Cluster** или всегда `Cluster`
Operation type	Тип операции, которая была произведена с кластером. Возможные значения: - `Create` - `Change` - `Drop`
Result	Результат операции. Возможные значения: - `Successful` — операция завершена; - `Failed` — операция завершилась с ошибкой.
Operation time	Дата и время выполнения операции в формате `DD.MM.YYYY HH:MM:SS`
Client hostname	Имя пользователя ОС
Session	Идентификатор сеанса ADB Control, в рамках которого выполнена операция
Node	Имя узла, на котором выполнена операция

В правом углу **Open table** кнопку, нажмите чтобы открыть окно `Table` для просмотра над текущей строкой таблицы. Нажмите снова, чтобы свернуть окно `Table`.

Список операций пользователей

Storages

Вкладка **ADBM** → **Storages** показывает информацию о создании, изменении и удалении хранилищ.

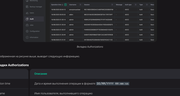


Вкладка Storages

Поля вкладки Storages

Поле	Описание
Object name	Название объекта
Object type	Объекты, над которыми осуществляется операция. Для всех видов **Storages** или всегда `Storages`
Operation type	Тип операции, которая была произведена с хранилищем. Возможные значения: - `Create` - `Change` - `Drop`
Result	Результат операции. Возможные значения: - `Successful` — операция завершена; - `Failed` — операция завершилась с ошибкой.
Operation time	Дата и время выполнения операции в формате `DD.MM.YYYY HH:MM:SS`
Client hostname	Имя пользователя ОС
Session	Идентификатор сеанса ADB Control, в рамках которого выполнена операция
Node	Имя узла, на котором выполнена операция

В правом углу **Open table** кнопку, нажмите чтобы открыть окно `Table` для просмотра над текущей строкой таблицы. Нажмите снова, чтобы свернуть окно `Table`.



Список операций пользователей

Табличные фильтры

Для удобства поиска и анализа данных в таблицах можно задать несколько фильтров, которые можно использовать для отбора необходимых данных. Для каждого поля можно задать свой фильтр. Для задания фильтра нажмите на кнопку **Filter**. Для того, чтобы задать несколько условий сразу, нажмите на кнопку **Advanced filter**. Для сохранения отчёта нажмите **Save**. Для сброса всех фильтров нажмите **Reset**. Для поиска нажмите **Find**. Для того, чтобы свернуть окно фильтров, нажмите на кнопку **Hide filters**.

Кнопка **Find** позволяет задавать следующие фильтры:

- **Equal** — фильтрующие по заданному значению. Возможно задать несколько значений сразу.
- **Not equal** — фильтрующие по заданному значению. Возможно задать несколько значений сразу.
- **Contains** — фильтрующие по заданной подстроке. Возможно задать несколько подстрок сразу.
- **Not contains** — фильтрующие по заданной подстроке. Возможно задать несколько подстрок сразу.
- **Is null** — фильтрующие по пустым полям.
- **Is not null** — фильтрующие по непустым полям.
- **Greater** — фильтрующие по заданному значению. Для дат, времени и чисел.
- **Greater or equal** — фильтрующие по заданному значению. Для дат, времени и чисел.
- **Less** — фильтрующие по заданному значению. Для дат, времени и чисел.
- **Less or equal** — фильтрующие по заданному значению. Для дат, времени и чисел.
- **Between** — фильтрующие по заданному диапазону. Для дат, времени и чисел.
- **Not between** — фильтрующие по заданному диапазону. Для дат, времени и чисел.
- **Last** — фильтрующие по последнему периоду. Возможно задать несколько значений сразу.
- **Not last** — фильтрующие по последнему периоду. Возможно задать несколько значений сразу.
- **Current** — фильтрующие по текущему периоду. Возможно задать несколько значений сразу.
- **Not current** — фильтрующие по текущему периоду. Возможно задать несколько значений сразу.
- **Next** — фильтрующие по следующему периоду. Возможно задать несколько значений сразу.
- **Not next** — фильтрующие по следующему периоду. Возможно задать несколько значений сразу.
- **Exists** — фильтрующие по заданному списку. Возможно задать несколько значений сразу.
- **Not exists** — фильтрующие по заданному списку. Возможно задать несколько значений сразу.
- **Regexp** — фильтрующие по регулярному выражению.
- **Not regexp** — фильтрующие по регулярному выражению.
- **Contains** — фильтрующие по заданному значению. Для всех видов данных.
- **Not contains** — фильтрующие по заданному значению. Для всех видов данных.
- **Is null** — фильтрующие по пустым полям. Для всех видов данных.
- **Is not null** — фильтрующие по непустым полям. Для всех видов данных.
- **Greater** — фильтрующие по заданному значению. Для всех видов данных.
- **Greater or equal** — фильтрующие по заданному значению. Для всех видов данных.
- **Less** — фильтрующие по заданному значению. Для всех видов данных.
- **Less or equal** — фильтрующие по заданному значению. Для всех видов данных.
- **Between** — фильтрующие по заданному диапазону. Для всех видов данных.
- **Not between** — фильтрующие по заданному диапазону. Для всех видов данных.
- **Exists** — фильтрующие по заданному списку. Для всех видов данных.
- **Not exists** — фильтрующие по заданному списку. Для всех видов данных.
- **Regexp** — фильтрующие по регулярному выражению. Для всех видов данных.
- **Not regexp** — фильтрующие по регулярному выражению. Для всех видов данных.

Просмотр jobs

Д. Дальк Карышева

AD.B CONTROL

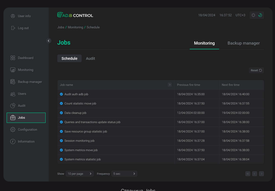
Содержание

Назад к содержанию

Monitoring
- Schedule
- Audit

Backup manager
- Schedule
- Audit

Страница **Jobs** в web-интерфейсе ADB Control предназначена для вывода информации по запланированным и выполненным системным задачам (job) в ADB Control/ADB Backup Manager. Страница включает несколько вкладок, каждая из которых детально описана ниже.

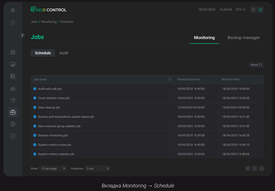


Страница Jobs

Monitoring

Schedule

Вкладка **Monitoring** → **Schedule** на странице **Jobs** отображает запланированные системные задачи (job) в ADB Control.



Вкладка Monitoring → Schedule

Таблица, изображенная на рисунке выше, выводит следующую информацию.

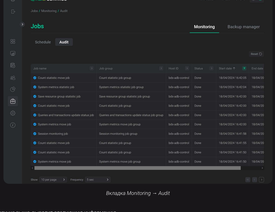
Поля вкладки Monitoring → Schedule

Поле	Описание
Job name	Название задачи. Полное описание задачи можно получить навяда курсор на ее название
Previous fire time	Дата и время предыдущего запуска в формате `DD/MM/YYYY HH:mm:ss`
Next fire time	Дата и время следующего запланированного запуска в формате `DD/MM/YYYY HH:mm:ss`

В заголовке столбца **Job name** таблицы со списком запланированных задач расположен фильтр, который можно использовать для отбора данных по названию задачи. Поиск возможен по первым введенным символам. Для открытия фильтра нажмите на иконку ⌕. Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

Audit

Вкладка **Monitoring** → **Audit** на странице **Jobs** отображает все запуски системных задач (job) в ADB Control за выбранный период времени. По умолчанию данные выводятся за текущую дату.



Вкладка Monitoring → Audit

Таблица, изображенная на рисунке выше, выводит следующую информацию.

Поля вкладки Monitoring → Audit

Поле	Описание
Job name	Название задачи. В случае если задача завершена с ошибкой (в статусе `Error`), в столбце **Job name** становится доступна кнопка ⋯. При ее нажатии под текущей строкой таблицы появляется блок с сообщением об исходной ошибке, которое можно скопировать нажав на иконку `Copy` в верхнем правом углу появившегося блока
Job group	Название группы, к которой относится задача
Host ID	Идентификатор хоста, на котором выполнялась задача, в формате `<IP-адрес>:<порт>`
Status	Статус выполнения задачи. Возможные значения: - `Done` — задача выполнена успешно; - `Error` — задача завершена с ошибкой; - `Unknown` — статус неизвестен; - `Running` — задача выполняется в текущий момент времени.
Start date	Дата и время начала выполнения задачи в формате `DD/MM/YYYY HH:mm:ss`
End date	Дата и время завершения выполнения задачи в формате `DD/MM/YYYY HH:mm:ss`

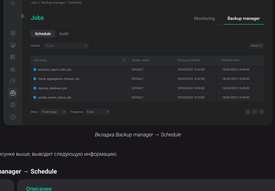
В заголовках столбцов таблицы со списком запусков задач расположены фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо нажать на иконку ⌕. Для тех столбцов, где набор возможных значений ограничен (например, **Job name**), в фильтре можно выбрать значение из выпадающего списка. Для некоторых столбцов (например, **Host ID**) требуется ввести искомое значение. Для столбцов, показывающих дату и время (например, **Start date**), временной диапазон можно выбрать из календаря.

Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

Backup manager

Schedule

Вкладка **Backup manager** → **Schedule** на странице **Jobs** отображает запланированные системные задачи (job) в ADB Backup Manager.



Вкладка Backup manager → Schedule

Таблица, изображенная на рисунке выше, выводит следующую информацию.

Поля вкладки Backup manager → Schedule

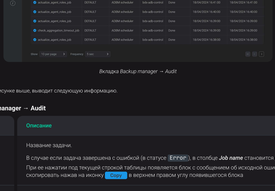
Поле	Описание
Job name	Название задачи. Полное описание задачи можно получить навяда курсор на ее название
Cluster name	Название кластера, для которого запланирована задача. Значение `DEFAULT`, используется для общих задач, не относящихся к конкретному кластеру
Previous fire time	Дата и время предыдущего запуска в формате `DD/MM/YYYY HH:mm:ss`
Next fire time	Дата и время следующего запланированного запуска в формате `DD/MM/YYYY HH:mm:ss`

Над таблицей со списком запланированных задач расположен фильтр **Cluster**, в котором можно выбрать кластер ADB, для которого требуется вывести данные в таблице. По умолчанию выбраны все кластеры.

В заголовке столбца **Job name** таблицы со списком запланированных задач расположен фильтр, который можно использовать для отбора данных по названию задачи. Поиск возможен по первым введенным символам. Для открытия фильтра нажмите на иконку ⌕. Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

Audit

Вкладка **Backup manager** → **Audit** на странице **Jobs** отображает все запуски системных задач (job) в ADB Backup Manager за выбранный период времени. По умолчанию данные выводятся за текущую дату.



Вкладка Backup manager → Audit

Таблица, изображенная на рисунке выше, выводит следующую информацию.

Поля вкладки Backup manager → Audit

Поле	Описание
Job name	Название задачи. В случае если задача завершена с ошибкой (в статусе `Error`), в столбце **Job name** становится доступна кнопка ⋯. При ее нажатии под текущей строкой таблицы появляется блок с сообщением об исходной ошибке, которое можно скопировать нажав на иконку `Copy` в верхнем правом углу появившегося блока
Cluster name	Название кластера, для которого выполнена задача. Значение `DEFAULT`, используется для общих задач, не относящихся к конкретному кластеру
Scheduler name	Название планировщика
Host ID	Имя хоста, на котором выполнялась задача
Status	Статус выполнения задачи. Возможные значения: - `Done` — задача выполнена успешно; - `Error` — задача завершена с ошибкой; - `Unknown` — статус неизвестен; - `Running` — задача выполняется в текущий момент времени.
Start date	Дата и время начала выполнения задачи в формате `DD/MM/YYYY HH:mm:ss`
End date	Дата и время завершения выполнения задачи в формате `DD/MM/YYYY HH:mm:ss`

Над таблицей со списком запусков задач расположен фильтр **Cluster**, в котором можно выбрать кластер ADB, для которого требуется вывести данные в таблице. По умолчанию выбраны все кластеры.

В заголовках столбцов таблицы со списком запусков задач расположены фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо нажать на иконку ⌕. Для тех столбцов, где набор возможных значений ограничен (например, **Job name**), в фильтре можно выбрать значение из выпадающего списка. Для некоторых столбцов (например, **Host ID**) требуется ввести искомое значение. Для столбцов, показывающих дату и время (например, **Start date**), временной диапазон можно выбрать из календаря.

Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

Просмотр информации по компонентам

Версия Enterprise

Содержание

Назад к содержанию

ADB Control
- UI backend server
- Agent
- Backend server
- Scheduler

Backup manager

Страница **Information** в веб-интерфейсе ADB Control отображает статус работы компонентов, входящих в состав ADB Control и ADBM. Страница состоит из нескольких вкладок, которые подробно описаны ниже.

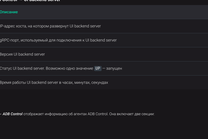
ADB Control

На странице **Information** → **ADB Control** фиксируется текущее состояние компонентов ADB Control.

> **ПРИМЕЧАНИЕ**
>
> Более подробную информацию об архитектуре ADB Control можно получить в статье Обзор Arenadata DB Control.

UI backend server

Вкладка **UI backend server** на странице **Information** → **ADB Control** содержит информацию о backend-сервере веб-интерфейса ADB Control (UI backend server).



Вкладка Information → ADB Control → UI backend server

Поля вкладки Information → ADB Control → UI backend server

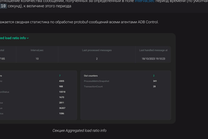
Поле	Описание
Host Name	IP-адрес хоста, на котором развернут UI backend server
GRPC Port	gRPC-порт, используемый для подключения к UI backend server
Version	Версия UI backend server
Status	Статус UI backend server. Возможно одно значение: **UP** – запущен
Up Time	Время работы UI backend server в часах, минутах, секундах

Agent

Вкладка **Agent** на странице **Information** → **ADB Control** отображает информацию об агентах ADB Control. Она включает две секции:

- *Overview*
- *Aggregated load ratio info*

Секция **Overview** предназначена для вывода общей информации по всем агентам, которые установлены на хосты кластера ADB, подключенных к системе мониторинга.

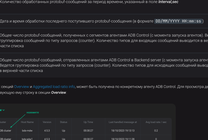


Секция Overview

Поля секции Overview

Поле	Описание
Cluster	Название кластера ADB
Host Name	Название хоста, на котором установлен агент ADB Control
Version	Версия агента ADB Control
Status	Статус агента ADB Control. Возможно только одно значение: **UP** – запущен. Остановленные агенты (например, при отключении хоста) не отображаются в таблице
Up Time	Время с момента последнего запуска агента в часах, минутах, секундах
Last handled message at	Дата и время обработки агентом последнего поступившего protobuf сообщения с сегментом (в формате: **DD/MM/YYYY HH:MM:SS**)
Avg load rate / sec	Средняя количественная скорость protobuf сообщений на агент с сегментом в секунду. Рассчитывается как отношение количества сообщений, полученных за определенный в поле interval.sec период времени (по умолчанию **10**) секунд, к величине этого периода

В секции **Aggregated load ratio info** отображается сводная статистика по обработке protobuf сообщений всеми агентами ADB Control.

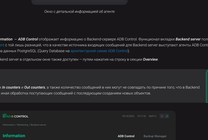


Секция Aggregated load ratio info

Поля секции Aggregated load ratio info

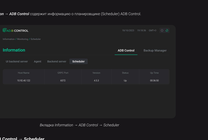
Поле	Описание
Total	Общее количество protobuf сообщений, полученных с сегментов агентами ADB Control (с момента запуска агентов)
Interval.sec	Период времени между измерениями статистики (counter) на основе поступления protobuf сообщений (в секундах). По умолчанию **10** секунд. Для изменения можно переопределить в конфиг-файлах параметром **eureka.update.counter.interval**
Last processed messages	Количество обработанных protobuf сообщений за период времени, указанный в поле **Interval.sec**
Last handled message at	Дата и время обработки последнего поступившего protobuf сообщения (в формате: **DD/MM/YYYY HH:MM:SS**)
In counters	Общее число protobuf сообщений, отправленных агентами ADB Control в Backend server (с момента запуска агентов). Водится суммарное количество сообщений по типу запросов (counter). Количество типов для входящих сообщений выводится в верхней части списка
Out counters	Общее число protobuf сообщений, отправленных агентами ADB Control в Backend server (с момента запуска агентов). Водится суммарное количество сообщений по типу запросов (counter). Количество типов для исходящих сообщений выводится в верхней части списка

Информация, описанная выше в разных секциях (Overview и Aggregated load ratio info), может быть получена по конкретному агенту ADB Control. Для просмотра детальных данных по агенту нажмите на соответствующую строку в секции **Overview**.



Переход к просмотру детальной информации по агенту

Все поля, отображаемые для выбранного агента, практически идентичны описанным выше. Добавляется поле **Failed counters** – количество счетчиков исходящих сообщений, при отправке которых возникала ошибка. Данный счетчик учитывается, что все счетчики в секциях обработки сообщений, отображаемые в индивидуальном окне, вычисляются исключительно для выбранного агента.



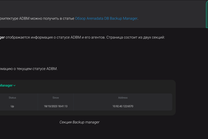
Окно с детальной информацией об агенте

Backend server

Вкладка **Backend server** на странице **Information** → **ADB Control** отображает информацию о Backend-сервере ADB Control. Функциональ вкладки **Backend server** полностью идентичен описанной выше вкладке Agent с той лишь разницей, что в качестве агента выступает исходящая сообщения для Backend server выступает агента ADB Control, а в роли получателя входящих сообщений – Backend server. Подробнее см. раздел данной инструкции «Agent ADB Control».

Просмотр детальной информации о Backend server в отдельном окне также доступен – нужно нажать на строку в секции **Overview**.

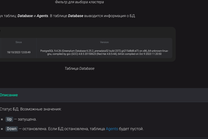
> **ПРИМЕЧАНИЕ**
>
> Наименования счетчиков в разделах **In counters** и **Out counters**, а также количество сообщений в них могут не совпадать. По причине того, что в Backend server выполняется агрегирование и иная обработка поступающих сообщений – поступающие сообщения имеют обработчик.



Вкладка Information → ADB Control → Backend server

Scheduler

Вкладка **Scheduler** на странице **Information** → **ADB Control** содержит информацию о планировщике (Scheduler) ADB Control.



Вкладка Information → ADB Control → Scheduler

Поля вкладки Information → ADB Control → Scheduler

Поле	Описание
Host Name	IP-адрес хоста, на котором развернут Scheduler
GRPC Port	gRPC-порт, используемый для подключения к Scheduler
Version	Версия Scheduler
Status	Статус Scheduler. Возможно одно значение: **UP** – запущен
Up Time	Время работы Scheduler в часах, минутах, секундах

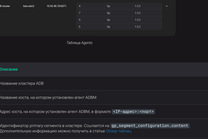
Backup manager

> **ПРИМЕЧАНИЕ**
>
> Более подробную информацию об архитектуре ADBM можно получить в статье Обзор Arenadata DB Backup Manager.

На странице **Information** → **Backup manager** отображаются информация о статусе ADBM и его агентах. Страница состоит из двух секций:

- *Backup manager*
- *Database & Agents*

Секция **Backup manager** содержит информацию о текущем статусе ADBM.



Секция Backup manager

Поля секции Backup manager

Поле	Описание
Status	Статус ADBM. Возможно одно значение: **Up** – сервер запущен. Если ADBM остановлен, страница **Backup manager** будет пустой
Since	Дата и время последнего запуска ADBM в формате: **DD/MM/YYYY HH:MM:SS**
Address	Адрес хоста, на котором развернут ADBM, в формате: **IIP:порт** или **host:**

В секциях **Database & Agents** отображаются агентские БД и базы данных ADB, подключенные для бэкапирования, и агенты ADBM, установленные на хосты кластера. Данные выводятся для каждого ADB, который выбран в фильтре непосредственно под заголовком **Database & Agents**.

Фильтр для выбора кластера

Секция **Database & Agents** состоит из двух таблиц, **Database** и **Agents**. В таблице **Database** выводятся информация о БД.

Таблица Database

Поля таблицы Database

Поле	Описание
Status	Статус БД. Возможные значения: - **Up** – запущена; - **Down** – остановлена. Если БД остановлена, таблица Agents будет пустой
Since	Дата и время последнего запуска БД в формате: **DD/MM/YYYY HH:MM:SS**
Version	Подробное описание сборки ADB, включая версию PostgreSQL и Greenplum DB

В таблице **Agents** выводятся информация об агентах ADBM.

Таблица Agents

Поля таблицы Agents

Поле	Описание
Cluster	Название кластера ADB
Host name	Название хоста, на котором установлен агент ADBM
Agent address	Адрес хоста, на котором установлен агент ADBM, в формате: **IIP:порт**
Segment number	Идентификатор сегмента в кластере. Ссылается на gp_segment_configuration.content. Дополнительную информацию можно получить в статье Сегмент кластера
Status	Статус агента ADBM. Возможные значения: - **Up** – агент запущен; - **Down** – агент в процессе остановки (например, в случае отключения хоста). Статус отображается непродолжительное время, после чего будет отправлен запрос в Service Registry на удаление неработающих агентов и запись об агенте удаляется из таблицы
Version	Версия агента ADBM

Обзор Arenadata DB Backup Manager

Дарья Барышева

Содержание

Назад к содержанию

Функции
Архитектура
Концепции
- Бэкап
- WAL
- Restore point
- Timeline

Начало работы

Функции

Arenadata DB Backup Manager (ADBM) – это построенная на основе pgbackrest отказоустойчивая система для управления бинарными бэкапами ADB.

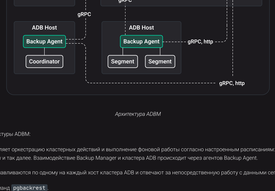
Основные функциональные возможности ADBM перечислены ниже:

- Гибкая настройка политик формирования бэкапов, включая отдельные расписания для различных их видов.
- Формирование бэкапов согласно настроенному расписанию и вручную.
- Ведение списка бэкапов с возможностью их поиска и просмотра деталей, в том числе кластерной топологии.
- Удаление бэкапов в соответствии с настроенными политиками по расписанию и вручную.
- Создание точек восстановления (restore point) согласно установленному расписанию и вручную.
- Восстановление баз данных на момент выбранной restore point.
- Логирование всех производимых в системе операций с возможностью детального просмотра действий, завершившихся сбоем.
- Поддержка работы с несколькими кластерами ADB различных типов (с/без standby и зеркалирования).
- Поддержка S3- и Posix-совместимых хранилищ для бэкапов.

> **ПРИМЕЧАНИЕ**
>
> - ADBM доступен в версии ADB Enterprise.
> - В настоящий момент возможна только offline-установка ADBM.

Архитектура

Верхнеуровневая архитектурная схема ADBM приведена ниже:



Архитектура ADBM

Основные компоненты архитектуры ADBM.

- **Backup Manager**. Осуществляет оркестрацию кластерных действий и выполнение фоновой работы согласно настроенным расписаниям: создание бэкапов и точек восстановления, их очистку и так далее. Взаимодействие Backup Manager и кластера ADB происходит через агентов Backup Agent.
- **Backup Agent**. Агенты устанавливаются по одному на каждый хост кластера ADB и отвечают за непосредственную работу с данными сегментов:
  - Запуск необходимых команд `pgbackrest`.
  - Управление конфигурацией `pgbackrest`.
  - На хосте координатора – операции по управлению кластером ADB (например, перезапуск).
- **Service Registry**. Отвечает за обнаружение сервисов (service discovery) в ADBM. Обеспечивает обнаружение доступных агентов (предыдущей карты агентов в системе нет) и поиск необходимых сервисов ADBM со стороны агентов (для отправки ответов). Так как Backup Manager может работать с несколькими кластерами ADB одновременно, при добавлении кластера новые агенты также регистрируются в Service Registry.
- **PostgreSQL**. Используется для хранения следующих данных:
  - Блокировки на уровне кластера для обеспечения эксклюзивности запускаемых действий.
  - Информация о состоянии текущего активного действия. В случае если Backup Manager останавливается или выходит из строя – метаданные о текущем действии берутся из PostgreSQL.
  - Исторические данные по бэкапам и конфигурациям.

  По умолчанию поднимается в контейнере. Также существует возможность подключения внешней базы данных PostgreSQL.

> **ПРИМЕЧАНИЕ**
>
> - Чтобы использовать внешний PostgreSQL, запомните параметры секции *External database parameters* при конфигурировании сервиса ADBM во время установки кластера ADB (см. Конфигурационные параметры → ADBM → External database parameters).
> - Начиная с версии ADBM 1.1.0 создание пользователей во внешней БД PostgreSQL и присвоение им необходимых прав производится вручную. Порядок настройки подробно описан в статье Настройка внешней БД для ADBM.

Концепции

ADBM реализует **point-in-time recovery (PITR)** – возможность восстановления баз данных на выбранный момент времени в прошлом. В основе этого подхода используются несколько концепций, кратко описанных ниже.

Бэкап

Бэкап (backup) – это конкретная копия данных кластера ADB, которая может быть использована для восстановления БД в случае аппаратных и других сбоев. ADBM поддерживает следующие типы бэкапов:

- **Полный (full)**. При формировании полной резервной копии все содержимое базы данных помещается в бэкап. Первый запущенный в рамках текущего timeline бэкап всегда будет иметь тип full в ADBM, даже если согласно расписанию (или выбору пользователя – в случае ручного запуска) первыми следуют дифференциальные либо инкрементные резервные копии. Восстановление полного бэкапа возможно напрямую, так как он не зависит от каких либо внешних файлов из других бэкапов. Преимуществом полного резервного копирования является быстрое восстановление всех файлов (по сравнению с другими видами). Однако полные копии не рекомендуются для регулярного резервного копирования (ежночасно или ежедневно), поскольку требуют много времени на формирование и занимают значительное место на диске.
- **Дифференциальный (differential)**. В дифференциальный бэкап помещаются только те файлы базы данных, которые изменились после запуска последнего полного бэкапа. По сравнению с полной копией дифференциальный бэкап формируется быстрее и занимает меньшее дискового пространства. Однако для восстановления бэкапа этого типа потребуется копирование не только его содержимого, но и файлов из последнего полного бэкапа.
- **Инкрементный (incremental)**. В инкрементный бэкап помещаются только файлы, изменившиеся с момента запуска последнего бэкапа любого типа (полного, дифференциального либо инкрементного). Из всех видов резервных копий инкрементные формируются быстрее всего и занимают минимальное место на диске. Однако их восстановление происходит дольше, поскольку необходимо извлечь файлы из последней полной и, при наличии, дифференциальной копии, а затем последовательно применить все последующие инкрементные бэкапы.

WAL

Write-ahead log (WAL) – это стандартный механизм, гарантирующий, что закоммиченные изменения не будут потеряны. Все изменения файлов с данными (хранящих таблицы, индексы и так далее) записываются последовательно в WAL. По прошествии некоторого времени фоновый процесс записывает изменения в основные файлы с данными. В случае сбоев запись WAL можно воспроизвести для возврата БД в консистентное состояние.

WAL разбивается на индивидуальные файлы по 64 Мб, называемые **сегментами (segment)**. Каждая запись в сегменте имеет 16-значный логический последовательный номер (Logical Sequence Number, LSN), используемый для поиска записи по смещению в текущем сегменте. Название сегмента, в свою очередь, формируется из номера timeline и LSN первой записи сегмента.

Использование WAL существенно сокращает количество операций записи на диск, так как нет необходимости записывать файлы данных после каждого коммита транзакции. Только WAL-файлы должны быть записаны на диск, чтобы гарантировать фиксацию транзакции.

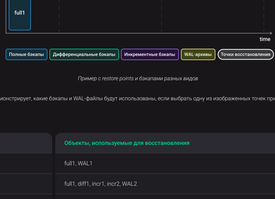
WAL играет важную роль в механизме Point-in-Time Recovery (PITR), используемом в ADBM. Архивируя WAL-файлы, ADBM позволяет восстанавливать БД на момент любой точки, покрываемой имеющимися данными WAL. Фактическое процесс восстановления включает два основных шага:

- Установить один или несколько предыдущих бэкапов.
- Воспроизвести записи WAL, сделанные после последнего загруженного бэкапа, до нужной именованной точки.

Restore point

Restore point – это именованная точка восстановления, которая является минимальной единицей гранулярности согласования данных в кластере ADB. После добавления restore point (по расписанию или вручную) ADBM позволяет восстановить базы данных кластера на момент создания этой точки. Для каждой созданной restore point ADBM хранит информацию о том, какие бэкапы потребуется восстановить и какие WAL-сегменты воспроизвести, чтобы вернуть БД в состояние на момент создания этой точки.

Для лучшего понимания механизма работы restore points можно обратиться к рисунку ниже:



Пример с restore points и бэкапами разных видов

Приведенная ниже таблица демонстрирует, какие бэкапы и WAL-файлы будут использованы, если выбрать одну из изображенных точек при выполнении восстановления данных.

Логика restore points

Выбранная restore point	Объекты, используемые для восстановления
rp1	full1, WAL1
rp2	full1, diff1, incr1, incr2, WAL2
rp3	full1, diff2, incr3, incr4, WAL3

Timeline

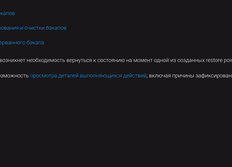
Timeline – это механизм, используемый в ADBM для того, чтобы отличать серии WAL, сгенерированные после восстановления БД на определенный момент времени, от тех, которые были созданы в исходной истории базы данных (до применения восстановления).

Предположим, что для восстановления БД на момент создания некоторой restore point после загрузки одного или нескольких бэкапов требуется воспроизведение `WAL1` (номер использован с целью упрощения). После успешного восстановления архива заполнение WAL продолжится и, если архивы WAL будут помещаться в ту же директорию, последующие записи WAL перезапишут те, что могли существовать в исходной истории БД после выбранной точки восстановления (`WAL2`, `WAL3` и так далее). А это, в свою очередь, приведет к невозможности восстановления БД для других точек, добавленных перед восстановлением.

Чтобы избежать описанных коллизий, в ADBM после каждого восстановления данных инициализируется новый номер timeline для идентификации архивов WAL, созданных после восстановления. Нумерация timeline в системе ведется с `0`; при каждом успешном восстановлении номер увеличивается на единицу. Этот номер фиксируется в названии директорий, хранящих архивы WAL и бэкапы (с разделением по сегментам Greenplum называемым **shards**). При каждом восстановлении данных создается все необходимые директории с новым номером timeline в названии, и все последующие записи WAL помещаются в новые директории, не перезаписывая данные в директориях с предыдущим номером.

> **ВАЖНО**
>
> Поскольку вновь создаваемые директории изначально не содержат бэкапов, первый бэкап после каждого восстановления данных будет полным (`full`), даже если согласно расписанию (или выбору пользователя – в случае ручного запуска) первыми следуют дифференциальная либо инкрементная резервные копии.

Следующий рисунок иллюстрирует описанный процесс в упрощенном виде. После восстановления на момент точки `rp1` воспроизводится `WAL1` и создается `timeline1`. Последующие записи WAL помещаются в новые директории. При следующем восстановлении создается `timeline2` и так далее. Обратите внимание, что после создания `timeline1` и `timeline2` по-прежнему сохраняется возможность восстановиться на момент точки `rp2` – без применения концепции timeline это было бы невозможно.



Механизм работы timeline

Начало работы

Типовая последовательность действий при работе с ADBM через web-интерфейс включает следующие шаги:

1. Подключение к ADBM.
2. Создание конфигурации, которая будет использоваться для формирования бэкапов и других связанных действий.
3. Запуск бэкапов вручную либо ожидание их автоматического формирования по расписанию. При необходимости – просмотр сведений о сформированных бэкапах.
4. Создание точек восстановления вручную либо ожидание их автоматического формирования по расписанию.
5. Выполнение эксплуатационных действий при необходимости:
   - Очистка бэкапов
   - Проверка сформированных бэкапов
   - Прерывание операций формирования и очистки бэкапов
   - Возобновление последнего прерванного бэкапа
6. Восстановление баз данных, если возникнет необходимость вернуться к состоянию на момент одной из созданных restore points.

На каждом из описанных шагов есть возможность просмотра деталей выполняющихся действий, включая причины зафиксированных сбоев (при их наличии).

Настройка внешней БД для ADBM

Дарья Барышева

Назад к содержанию

В production-средах рекомендуется использовать внешнюю базу данных PostgreSQL для хранения данных ADBM в целях обеспечения высокой производительности и отказоустойчивости (High Availability, HA). Ниже приведены шаги, необходимые для настройки внешней БД начиная с версии ADBM 1.7.3:

1. Выполните следующие запросы на стороне PostgreSQL. Запросы следует запускать под дефолтным пользователем `postgres` с правами `superuser`:

```
CREATE ROLE <user_name> LOGIN PASSWORD '<user_password>'; ❶
ALTER ROLE <user_name> SET search_path TO public, adbm;
CREATE DATABASE <database_name> WITH owner <user_name>; ❷
```

❶ Создание пользователя `<user_name>` с необходимыми правами для подключения к внешней БД.

❷ Создание базы данных `<database_name>` для миграции данных ADBM.

2. Подключитесь к созданной базе данных `<database_name>` под суперпользователем и выполните:

```
CREATE SCHEMA adbm; ❶
GRANT USAGE, CREATE ON SCHEMA adbm TO <user_name>; ❷
ALTER DEFAULT PRIVILEGES IN SCHEMA adbm GRANT UPDATE, INSERT, SELECT, DELETE ON TABLES TO <user_name>;
CREATE EXTENSION btree_gin WITH schema adbm;
CREATE EXTENSION pg_trgm WITH schema adbm;
GRANT SELECT ON pg_extension TO <user_name>;
```

❶ Создание схемы `adbm` в базе данных `<database_name>` для миграции данных ADBM

❷ Создание расширений (extension) в схеме `adbm` и предоставление созданному пользователю `<user_name>` прав для их использования.

Если есть возможность назначить права superuser пользователю в PostgreSQL (под которым будет выполняться подключение из ADBM), достаточно создать роль и базу данных в PostgreSQL перед установкой или реконфигурацией ADBM для использования внешней БД:

```
CREATE ROLE <user_name> LOGIN PASSWORD '<user_password>' SUPERUSER;
ALTER ROLE <user_name> SET search_path TO public, adbm;
CREATE DATABASE <database_name> WITH owner <user_name>;
```

3. Обеспечьте возможность доступа хоста с ADBM к внешней базе данных PostgreSQL. Для этого на стороне сервера PostgreSQL требуется добавить запись следующего вида в файл *pg_hba.conf*.

Без SSL

```
host    <database_name>  <user_name>      <adbm_address>     trust
```

где:

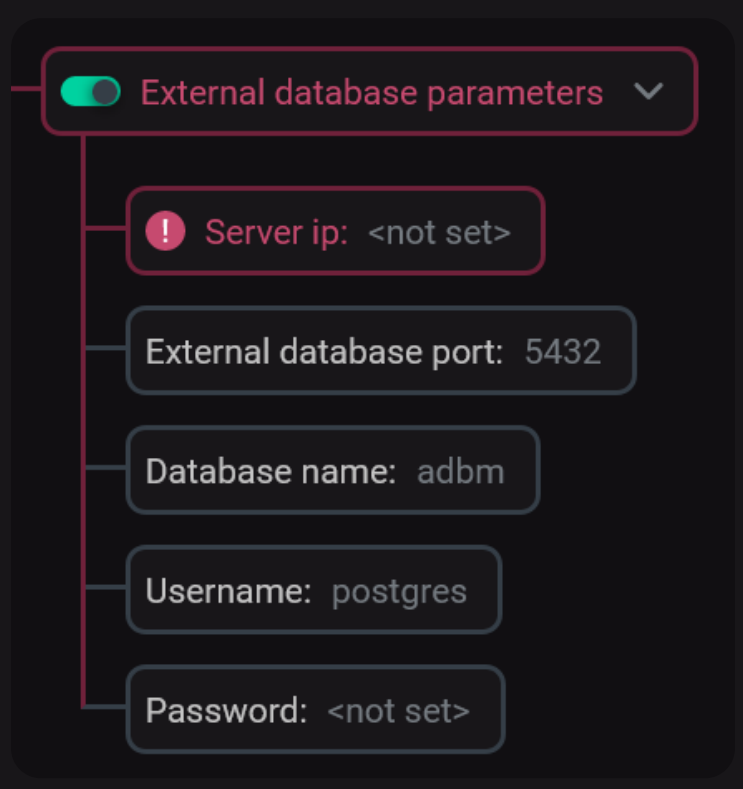
- `<database_name>` — имя созданной выше БД.
- `<user_name>` — имя созданного выше пользователя.
- `<adbm_address>` — IP-адрес ADBM с подсетью.

С использованием SSL

```
hostssl    all  all       0.0.0.0/0      md5
```

Дополнительную информацию о предварительной настройке PostgreSQL для использования SSL можно получить в статье Шифрование SSL.

4. На странице конфигурирования сервиса ADBM переведите в активное положение переключатель ***External database parameters*** и заполните параметры в открывшемся списке. Назначение параметров описано в разделе Конфигурационные параметры → ADBM → External database parameters.



Параметры для подключения к внешней БД PostgreSQL

5. После заполнения параметров нажмите ***Save*** для сохранения конфигурации ADBM.
6. Выполните установку сервиса ADBM с помощью сервисного действия Install либо примените действие ***Reconfigure & Restart*** к ранее установленному сервису ADBM.

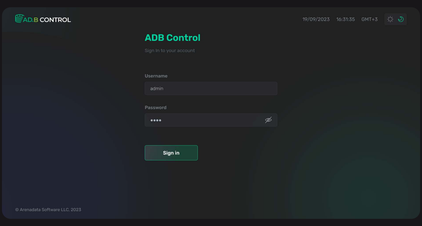
Подключение к ADBM

Дарья Барышева

Назад к содержанию

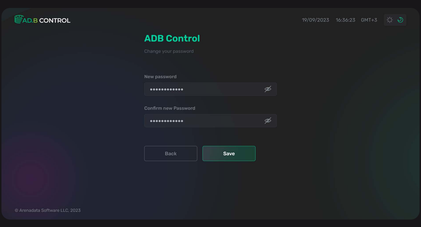
В настоящее время работа с ADBM осуществляется через web-интерфейс Arenadata DB Control (ADB Control). Для подключения к ADBM выполните следующие шаги:

1. В адресной строке браузера введите адрес хоста, на котором развернут ADB Control. В качестве номера порта укажите значение параметра Listening port из раздела ***UI component parameters***, определенное на этапе конфигурирования сервиса ADB Control при установке кластера ADB (по умолчанию `81`).
2. В открывшемся окне введите учетные данные пользователя, который создается по умолчанию, и нажмите ***Sign in***.
   - ***Username:*** admin
   - ***Password:*** 1234



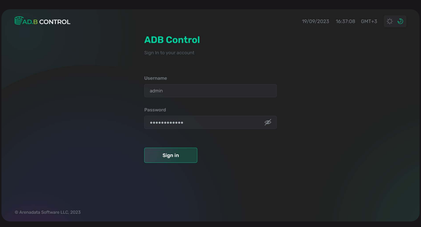
Подключение к ADB Control

3. При первом входе в систему необходимо изменить пароль для пользователя `admin`. Введите новый пароль в поле ***New password*** и подтвердите его в поле ***Confirm new Password***. Нажмите ***Save***. Обратите внимание, что длина пароля должна быть более либо равна 12 символам по умолчанию. К вводу допускаются любые символы кроме пробелов.



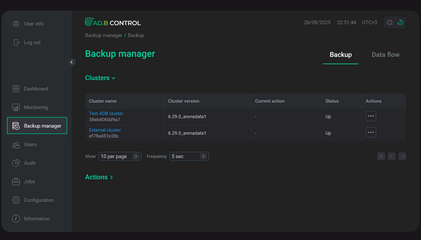
Смена пароля

4. После успешного сохранения нового пароля выполните вход под пользователем, повторно введя логин и новый пароль. Обратите внимание, что при достижении определенного числа неудачных попыток входа пользователь будет заблокирован — на постоянной или временной основе в зависимости от настроек.



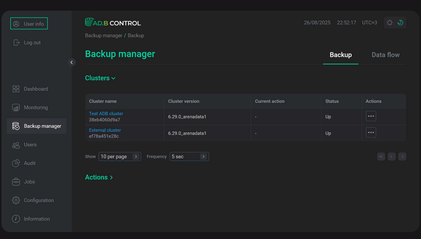
Вход с использованием измененного пароля

5. На следующей странице выберите пункт ***Backup manager*** в левом навигационном меню.



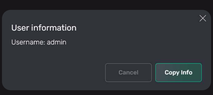
Страница Backup manager

6. Если в дальнейшем потребуется получить информацию о пользователе, выполнившем вход в ADB Control — выберите вкладку ***User info*** в левом навигационном меню.



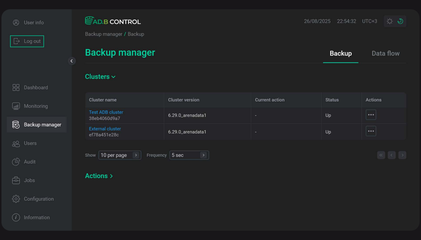
Получение информации о текущем пользователе

В появившемся модальном окне отображается имя текущего пользователя, которое можно скопировать нажав кнопку ***Copy info***.



Копирование информации о текущем пользователе

7. Для выхода из системы нажмите ***Log out*** в левом навигационном меню.



Выход из системы

- Все действия, описанные в последующих разделах документации ADBM, выполняются на странице ***Backup manager*** в web-интерфейсе ADB Control.
- Страница ***Backup manager*** доступна только пользователям с ролью `Administrator`, `ADBM Administrator` или `Owner`.
- Информацию о том, как добавить новых пользователей и настроить права доступа для них, можно найти в следующих статьях:
  - Basic-аутентификация
  - Аутентификация с использованием LDAP
  - Авторизация
- Изменение политик безопасности (ограничения на длину пароля, максимальное количество неудачных попыток входа, правила блокировки пользователя) возможно на странице ***Configuration → Security***. За дополнительной информацией обратитесь к статье Настройка политик безопасности.

Управление конфигурациями в АСВК

Содержание

Конфигурация basic

Общие

Управление

Добавление новой конфигурации

Редактирование конфигурации

Удаление конфигурации

Частные конфигурации

Расписание

Кружающие

Дополнительные конфигурации

Мониторинг заявления

Содержание

1. Конфигурация basic
   1.1. Общие
   1.2. Управление
   1.3. Добавление новой конфигурации
   1.4. Редактирование конфигурации
   1.5. Удаление конфигурации
   1.6. Частные конфигурации
   1.7. Расписание
   1.8. Кружающие
   1.9. Дополнительные конфигурации
   1.10. Мониторинг заявления

1. Конфигурация basic

1.1. Общие

Для работы с общими настройками необходимо перейти в раздел **Общие** в меню **Конфигурация basic**.

Рисунок 1.1.1 – Раздел «Общие»

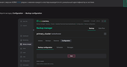


Рисунок 1.1.2 – Раздел «Общие» в разрешении 1920x1080

1.2. Управление

Для работы с конфигурациями необходимо перейти в раздел **Управление** в меню **Конфигурация basic**.

Рисунок 1.2.1 – Раздел «Управление»

Рисунок 1.2.2 – Раздел «Управление» в разрешении 1920x1080

1.3. Добавление новой конфигурации

Для добавления новой конфигурации:

1. Перейдите в раздел **Управление**, нажмите кнопку **Добавить**.
2. Заполните поля в открывшемся окне:
   - **Название** – краткое название конфигурации (обязательное поле);
   - **Описание** – подробное описание конфигурации (обязательное поле);
   - **Тип** – выберите тип конфигурации из списка (обязательное поле);
   - **Значение** – введите значение конфигурации (обязательное поле);
   - **Доступ к конфигурации** – выберите роли пользователей, которым будет доступна конфигурация (обязательное поле);
   - **Доступ к изменению** – выберите роли пользователей, которым будет доступно изменение конфигурации (обязательное поле);
   - **Доступ к удалению** – выберите роли пользователей, которым будет доступно удаление конфигурации (обязательное поле);
   - **Доступ к просмотру** – выберите роли пользователей, которым будет доступен просмотр конфигурации (обязательное поле);
   - **Доступ к изменению значения** – выберите роли пользователей, которым будет доступно изменение значения конфигурации (обязательное поле);
   - **Доступ к просмотру значения** – выберите роли пользователей, которым будет доступен просмотр значения конфигурации (обязательное поле);
   - **Доступ к изменению доступа** – выберите роли пользователей, которым будет доступно изменение доступа к конфигурации (обязательное поле);
   - **Доступ к просмотру доступа** – выберите роли пользователей, которым будет доступен просмотр доступа к конфигурации (обязательное поле);
   - **Доступ к изменению доступа к изменению** – выберите роли пользователей, которым будет доступно изменение доступа к изменению конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к изменению** – выберите роли пользователей, которым будет доступен просмотр доступа к изменению конфигурации (обязательное поле);
   - **Доступ к изменению доступа к удалению** – выберите роли пользователей, которым будет доступно изменение доступа к удалению конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к удалению** – выберите роли пользователей, которым будет доступен просмотр доступа к удалению конфигурации (обязательное поле);
   - **Доступ к изменению доступа к просмотру** – выберите роли пользователей, которым будет доступно изменение доступа к просмотру конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к просмотру** – выберите роли пользователей, которым будет доступен просмотр доступа к просмотру конфигурации (обязательное поле);
   - **Доступ к изменению доступа к изменению значения** – выберите роли пользователей, которым будет доступно изменение доступа к изменению значения конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к изменению значения** – выберите роли пользователей, которым будет доступен просмотр доступа к изменению значения конфигурации (обязательное поле);
   - **Доступ к изменению доступа к удалению значения** – выберите роли пользователей, которым будет доступно изменение доступа к удалению значения конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к удалению значения** – выберите роли пользователей, которым будет доступен просмотр доступа к удалению значения конфигурации (обязательное поле);
   - **Доступ к изменению доступа к просмотру значения** – выберите роли пользователей, которым будет доступно изменение доступа к просмотру значения конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к просмотру значения** – выберите роли пользователей, которым будет доступен просмотр доступа к просмотру значения конфигурации (обязательное поле);
   - **Доступ к изменению доступа к изменению доступа к изменению** – выберите роли пользователей, которым будет доступно изменение доступа к изменению доступа к изменению конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к изменению доступа к изменению** – выберите роли пользователей, которым будет доступен просмотр доступа к изменению доступа к изменению конфигурации (обязательное поле);
   - **Доступ к изменению доступа к удалению доступа к изменению** – выберите роли пользователей, которым будет доступно изменение доступа к удалению доступа к изменению конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к удалению доступа к изменению** – выберите роли пользователей, которым будет доступен просмотр доступа к удалению доступа к изменению конфигурации (обязательное поле);
   - **Доступ к изменению доступа к просмотру доступа к изменению** – выберите роли пользователей, которым будет доступно изменение доступа к просмотру доступа к изменению конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к просмотру доступа к изменению** – выберите роли пользователей, которым будет доступен просмотр доступа к просмотру доступа к изменению конфигурации (обязательное поле);
   - **Доступ к изменению доступа к изменению доступа к удалению** – выберите роли пользователей, которым будет доступно изменение доступа к изменению доступа к удалению конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к изменению доступа к удалению** – выберите роли пользователей, которым будет доступен просмотр доступа к изменению доступа к удалению конфигурации (обязательное поле);
   - **Доступ к изменению доступа к просмотру доступа к удалению** – выберите роли пользователей, которым будет доступно изменение доступа к просмотру доступа к удалению конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к просмотру доступа к удалению** – выберите роли пользователей, которым будет доступен просмотр доступа к просмотру доступа к удалению конфигурации (обязательное поле);
   - **Доступ к изменению доступа к изменению доступа к просмотру** – выберите роли пользователей, которым будет доступно изменение доступа к изменению доступа к просмотру конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к изменению доступа к просмотру** – выберите роли пользователей, которым будет доступен просмотр доступа к изменению доступа к просмотру конфигурации (обязательное поле);
   - **Доступ к изменению доступа к удалению доступа к просмотру** – выберите роли пользователей, которым будет доступно изменение доступа к удалению доступа к просмотру конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к удалению доступа к просмотру** – выберите роли пользователей, которым будет доступен просмотр доступа к удалению доступа к просмотру конфигурации (обязательное поле);
   - **Доступ к изменению доступа к просмотру доступа к просмотру** – выберите роли пользователей, которым будет доступно изменение доступа к просмотру доступа к просмотру конфигурации (обязательное поле);
   - **Доступ к просмотру доступа к просмотру доступа к просмотру** – выберите роли пользователей, которым будет доступен просмотр доступа к просмотру доступа к просмотру

Запуск работы через API

Docs

Содержание

Обзор
Backup
Restore
Clone
Cancel
Delete
Status
Schedule
Configs
Error codes

Перейти к разделу

Обзор

Для запуска задач резервного копирования, восстановления, клонирования и управления ими используется REST API. Все запросы выполняются по адресу `https://cloud.ru/api/v1/backup/...`. Ответы приходят в формате JSON. Для каждого метода ниже показаны пример запроса и пример ответа.

Все запросы требуют заголовка `Authorization: Bearer <access_token>`. Подробнее о получении токена см. раздел Backup.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/backup/tasks/{task_id}/status
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "name": "backup-prod-db-01",
  "type": "backup",
  "status": "running",
  "progress": 42,
  "started_at": "2026-04-12T08:15:00Z",
  "finished_at": null,
  "duration_seconds": null,
  "source": {
    "resource_id": "vm-prod-db-01",
    "resource_type": "virtual_machine",
    "region": "ru-moscow-1"
  },
  "destination": {
    "storage_id": "s3-backup-ru-moscow-1",
    "path": "/backup/prod/db-01/2026-04-12/"
  }
}
```

Для запуска задачи резервного копирования используется `POST /backup/tasks/{task_id}/start`. Тело запроса может быть пустым — все параметры задачи уже заданы при создании.

Пример запуска

```
curl -X POST https://cloud.ru/api/v1/backup/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c/start
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 202 Accepted
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "status": "queued",
  "queued_at": "2026-04-12T08:15:00Z"
}
```

Backup

Для запуска задачи резервного копирования используется `POST /backup/tasks/{task_id}/start`. Тело запроса может быть пустым — все параметры задачи уже заданы при создании.

Пример запуска

```
curl -X POST https://cloud.ru/api/v1/backup/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c/start
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 202 Accepted
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "status": "queued",
  "queued_at": "2026-04-12T08:15:00Z"
}
```

Для получения статуса задачи используется `GET /backup/tasks/{task_id}/status`. Ответ содержит текущий статус, прогресс в процентах и время последнего обновления.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/backup/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c/status
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "status": "running",
  "progress": 42,
  "updated_at": "2026-04-12T08:15:00Z"
}
```

Для отмены задачи используется `POST /backup/tasks/{task_id}/cancel`. Тело запроса может быть пустым. Ответ приходит сразу, но фактическая остановка может занять несколько минут — статус задачи будет переходить в `cancelling` до завершения.

Пример запроса

```
curl -X POST https://cloud.ru/api/v1/backup/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c/cancel
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 202 Accepted
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "status": "cancelling",
  "requested_at": "2026-04-12T08:20:00Z"
}
```

Для удаления задачи используется `DELETE /backup/tasks/{task_id}`. Удаление возможно только для задач в статусе `completed`, `failed` или `cancelled`. Для активных задач сначала нужно отменить задачу.

Пример запроса

```
curl -X DELETE https://cloud.ru/api/v1/backup/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 204 No Content
```

Restore

Для запуска задачи восстановления используется `POST /restore/tasks/{task_id}/start`. Тело запроса может быть пустым — все параметры задачи уже заданы при создании.

Пример запуска

```
curl -X POST https://cloud.ru/api/v1/restore/tasks/rs-2b8e5d1a-9c4f-7a3b-6e0d-1f8a2c5b9e7d/start
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 202 Accepted
Content-Type: application/json
{
  "task_id": "rs-2b8e5d1a-9c4f-7a3b-6e0d-1f8a2c5b9e7d",
  "status": "queued",
  "queued_at": "2026-04-12T09:30:00Z"
}
```

Для получения статуса задачи используется `GET /restore/tasks/{task_id}/status`. Ответ содержит текущий статус, прогресс в процентах и время последнего обновления.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/restore/tasks/rs-2b8e5d1a-9c4f-7a3b-6e0d-1f8a2c5b9e7d/status
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "rs-2b8e5d1a-9c4f-7a3b-6e0d-1f8a2c5b9e7d",
  "status": "running",
  "progress": 67,
  "updated_at": "2026-04-12T09:45:00Z"
}
```

Для отмены задачи используется `POST /restore/tasks/{task_id}/cancel`. Тело запроса может быть пустым. Ответ приходит сразу, но фактическая остановка может занять несколько минут — статус задачи будет переходить в `cancelling` до завершения.

Пример запроса

```
curl -X POST https://cloud.ru/api/v1/restore/tasks/rs-2b8e5d1a-9c4f-7a3b-6e0d-1f8a2c5b9e7d/cancel
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 202 Accepted
Content-Type: application/json
{
  "task_id": "rs-2b8e5d1a-9c4f-7a3b-6e0d-1f8a2c5b9e7d",
  "status": "cancelling",
  "requested_at": "2026-04-12T09:50:00Z"
}
```

Для удаления задачи используется `DELETE /restore/tasks/{task_id}`. Удаление возможно только для задач в статусе `completed`, `failed` или `cancelled`. Для активных задач сначала нужно отменить задачу.

Пример запроса

```
curl -X DELETE https://cloud.ru/api/v1/restore/tasks/rs-2b8e5d1a-9c4f-7a3b-6e0d-1f8a2c5b9e7d
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 204 No Content
```

Clone

Для запуска задачи клонирования используется `POST /clone/tasks/{task_id}/start`. Тело запроса может быть пустым — все параметры задачи уже заданы при создании.

Пример запуска

```
curl -X POST https://cloud.ru/api/v1/clone/tasks/cl-4d9f2a7c-1e8b-5c6d-3a0f-7b4e9d2c1a6f/start
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 202 Accepted
Content-Type: application/json
{
  "task_id": "cl-4d9f2a7c-1e8b-5c6d-3a0f-7b4e9d2c1a6f",
  "status": "queued",
  "queued_at": "2026-04-12T10:00:00Z"
}
```

Для получения статуса задачи используется `GET /clone/tasks/{task_id}/status`. Ответ содержит текущий статус, прогресс в процентах и время последнего обновления.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/clone/tasks/cl-4d9f2a7c-1e8b-5c6d-3a0f-7b4e9d2c1a6f/status
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "cl-4d9f2a7c-1e8b-5c6d-3a0f-7b4e9d2c1a6f",
  "status": "running",
  "progress": 12,
  "updated_at": "2026-04-12T10:05:00Z"
}
```

Для отмены задачи используется `POST /clone/tasks/{task_id}/cancel`. Тело запроса может быть пустым. Ответ приходит сразу, но фактическая остановка может занять несколько минут — статус задачи будет переходить в `cancelling` до завершения.

Пример запроса

```
curl -X POST https://cloud.ru/api/v1/clone/tasks/cl-4d9f2a7c-1e8b-5c6d-3a0f-7b4e9d2c1a6f/cancel
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 202 Accepted
Content-Type: application/json
{
  "task_id": "cl-4d9f2a7c-1e8b-5c6d-3a0f-7b4e9d2c1a6f",
  "status": "cancelling",
  "requested_at": "2026-04-12T10:10:00Z"
}
```

Для удаления задачи используется `DELETE /clone/tasks/{task_id}`. Удаление возможно только для задач в статусе `completed`, `failed` или `cancelled`. Для активных задач сначала нужно отменить задачу.

Пример запроса

```
curl -X DELETE https://cloud.ru/api/v1/clone/tasks/cl-4d9f2a7c-1e8b-5c6d-3a0f-7b4e9d2c1a6f
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 204 No Content
```

Status

Для получения статуса задачи используется `GET /status/tasks/{task_id}`. Ответ содержит полную информацию о задаче: тип, имя, статус, прогресс, время начала и завершения, длительность, источник и назначение.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/status/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "name": "backup-prod-db-01",
  "type": "backup",
  "status": "running",
  "progress": 42,
  "started_at": "2026-04-12T08:15:00Z",
  "finished_at": null,
  "duration_seconds": null,
  "source": {
    "resource_id": "vm-prod-db-01",
    "resource_type": "virtual_machine",
    "region": "ru-moscow-1"
  },
  "destination": {
    "storage_id": "s3-backup-ru-moscow-1",
    "path": "/backup/prod/db-01/2026-04-12/"
  }
}
```

Для получения расписания задачи используется `GET /status/tasks/{task_id}/schedule`. Ответ содержит тип расписания, cron-выражение, часовой пояс и время следующего запуска.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/status/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c/schedule
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "schedule_type": "cron",
  "cron": "0 2 * * 1-5",
  "timezone": "Europe/Moscow",
  "next_run_at": "2026-04-13T02:00:00Z"
}
```

Для получения конфигурации задачи используется `GET /status/tasks/{task_id}/configs`. Ответ содержит параметры источника и назначения, политику ретраев, шифрование и сжатие.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/status/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c/configs
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "source": {
    "resource_id": "vm-prod-db-01",
    "resource_type": "virtual_machine",
    "region": "ru-moscow-1"
  },
  "destination": {
    "storage_id": "s3-backup-ru-moscow-1",
    "path": "/backup/prod/db-01/2026-04-12/"
  }
}
```

Schedule

Для получения расписания задачи используется `GET /schedule/tasks/{task_id}`. Ответ содержит тип расписания, cron-выражение, часовой пояс и время следующего запуска.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/schedule/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "schedule_type": "cron",
  "cron": "0 2 * * 1-5",
  "timezone": "Europe/Moscow",
  "next_run_at": "2026-04-13T02:00:00Z"
}
```

Для изменения расписания задачи используется `PUT /schedule/tasks/{task_id}`. Тело запроса содержит новые параметры расписания. Ответ приходит с обновлённым расписанием.

Пример запроса

```
curl -X PUT https://cloud.ru/api/v1/schedule/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
  -H "Content-Type: application/json"
  -d '{
    "schedule_type": "cron",
    "cron": "0 3 * * 1-5",
    "timezone": "Europe/Moscow"
  }'
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "schedule_type": "cron",
  "cron": "0 3 * * 1-5",
  "timezone": "Europe/Moscow",
  "next_run_at": "2026-04-13T03:00:00Z"
}
```

Для удаления расписания задачи используется `DELETE /schedule/tasks/{task_id}`. После удаления задача больше не будет запускаться по расписанию.

Пример запроса

```
curl -X DELETE https://cloud.ru/api/v1/schedule/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 204 No Content
```

Configs

Для получения конфигурации задачи используется `GET /configs/tasks/{task_id}`. Ответ содержит параметры источника и назначения, политику ретраев, шифрование и сжатие.

Пример запроса

```
curl -X GET https://cloud.ru/api/v1/configs/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "source": {
    "resource_id": "vm-prod-db-01",
    "resource_type": "virtual_machine",
    "region": "ru-moscow-1"
  },
  "destination": {
    "storage_id": "s3-backup-ru-moscow-1",
    "path": "/backup/prod/db-01/2026-04-12/"
  }
}
```

Для изменения конфигурации задачи используется `PUT /configs/tasks/{task_id}`. Тело запроса содержит новые параметры. Ответ приходит с обновлённой конфигурацией.

Пример запроса

```
curl -X PUT https://cloud.ru/api/v1/configs/tasks/bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsIkpXVCJ..."
  -H "Accept: application/json"
  -H "Content-Type: application/json"
  -d '{
    "source": {
      "resource_id": "vm-prod-db-01",
      "resource_type": "virtual_machine",
      "region": "ru-moscow-1"
    },
    "destination": {
      "storage_id": "s3-backup-ru-moscow-1",
      "path": "/backup/prod/db-01/2026-04-12/"
    }
  }'
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "task_id": "bk-7f3a9c2e-4d1b-8e6f-0a5c-9b2d7e4f1a8c",
  "source": {
    "resource_id": "vm-prod-db-01",
    "resource_type": "virtual_machine",
    "region": "ru-moscow-1"
  },
  "destination": {
    "storage_id": "s3-backup-ru-moscow-1",
    "path": "/backup/prod/db-01/2026-04-12/"
  }
}
```

Просмотр бэкапов в ADBM

Д. Дарья Барышева

Содержание

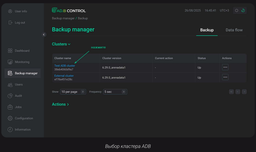
Назад к содержанию

Список бэкапов
Детали бэкапа

Список бэкапов

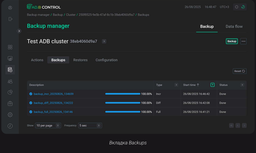
Независимо от способа создания (вручную или по расписанию), все бэкапы отображаются на вкладке **Backups**. Чтобы открыть эту вкладку, выполните шаги:

1. Откройте страницу Backup manager в веб-интерфейсе ADB Control.
2. Нажмите на имя кластера ADB в таблице, расположенной в секции **Clusters** открывшейся страницы.



Выбор кластера ADB

3. Выберите вкладку **Backups**.

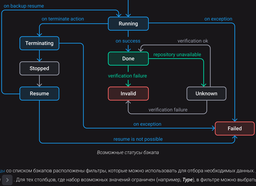


Вкладка Backups

Открывшаяся страница выводит следующую информацию о бэкапах.

Поля списка бэкапов

Поле	Описание
Description	Название бэкапа. Заполняется автоматически при формировании бэкапа по следующему шаблону: `backup_<type>_<start_date>_<start_time>`, где: • `<type>` — тип бэкапа (см. *Type* ниже); • `<start_date>` — дата начала формирования бэкапа в формате `YYYYMMDD`; • `<start_time>` — время начала формирования бэкапа в формате `HHmmss`. Начиная с версии ADBM 4.4.0 справа от названия выводится индикатор времени выполнения бэкапа (в % от рассчитанной продолжительности операции на всех сегментах кластера). Обратите внимание, что расчет прогресса выполнения бэкапа осуществляется только для этапа физического резервирования кластера и не включает в себя остальные действия
Type	Тип бэкапа, выбранный при его формировании. Возможные значения: • `Full` — полный бэкап; • `Diff` — дифференциальный бэкап; • `Incr` — инкрементный бэкап.
Start time	Время начала формирования бэкапа в формате `DD/MM/YYYY HH:mm:ss`
Status	Статус бэкапа. Возможные значения: • `Init` — инициализируется запуск бэкапа. • `Running` — бэкап формируется. • `Terminating` — запущено прерывание (terminate) формирования бэкапа. • `Invalid` — после выполненной проверки (verify) бэкап признан некорректным. Подобные бэкапы будут удалены при следующей процедуре очистки (cleanup). • `Stopped` — запуск бэкапа остановлен после успешного прерывания (terminate). • `Resume` — формирование бэкапа возобновлено (resume). • `Failed` — запуск бэкапа завершен с ошибкой. • `Done` — бэкап успешно сформирован. • `Unknown` — ретроспект, указанный в конфигурации бэкапа, недоступен. Дополнительную информацию о возможных переходах между статусами можно получить из приведенной ниже диаграммы



Возможные статусы бэкапа

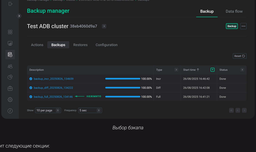
В заголовках столбцов таблицы со списком бэкапов расположены фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо навести на иконку ▽. Для тех столбцов, где набор возможных значений ограничен (например, **Type**), в фильтре можно выбрать значение из выпадающего списка. Для столбцов, показывающих дату и время (например, **Start time**), временной диапазон можно выбрать из календаря.

Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

> **ВАЖНО!**
> В настоящее время таблица, расположенная на вкладке **Backups**, по умолчанию показывает бэкапы, снятые за текущую дату. Чтобы вывести все бэкапы, воспользуйтесь фильтрами.

Детали бэкапа

Чтобы просмотреть детальную информацию о бэкапе, нажмите на описание бэкапа (**Description**) в таблице на вкладке **Backups**.

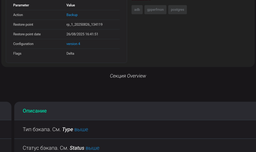


Выбор бэкапа

Следующая страница содержит следующие секции:

- **Overview**
- **Topology**
- **Statuses**

Секция **Overview** показывает базовую информацию о бэкапе. В верхней части секции выводится имя бэкапа. Начиная с версии ADBM 4.4.0 справа от имени отображается индикатор времени выполнения бэкапа (в % от рассчитанной продолжительности операции на всех сегментах кластера).

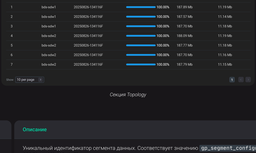


Секция Overview

Поля секции Overview

Поле	Описание
Type	Тип бэкапа. См. *Type* выше
Status	Статус бэкапа. См. *Status* выше
Date	Время начала формирования бэкапа в формате `DD/MM/YYYY HH:mm:ss`
Database size	Суммарный объем файлов данных, которые были заблокированы
Backup size	Суммарный объем бэкапа
Action	Название действия, в рамках которого был запущен выбранный бэкап. Можно нажать на название для перехода к просмотру детали действия
Restore point	Название точки восстановления (restore point), созданной после формирования бэкапа
Restore point date	Время создания точки восстановления (restore point) в формате `DD/MM/YYYY HH:mm:ss`
Configuration	Версия конфигурации. Для перехода к просмотру конфигурации нажмите на номер версии
Flags	Флаги, выбранные при создании бэкапа (см. *Delta* в описании действия Backup)
Databases	Список баз данных, которые были в СУБД на момент снятия бэкапа

Секция **Topology** содержит информацию о топологии.



Секция Topology

Поля секции Topology

Поле	Описание
Segment	Уникальный идентификатор сегмента данных. Соответствует значению `gp_segment_configuration.content` для текущего сегмента
Host	Имя хоста
Label	Название бэкапа в соответствии с внутренней логикой `pgbackrest`
Progress	Индикатор прогресса формирования резервной копии данных для текущего сегмента (в % от рассчитанной продолжительности операции). Отображается начиная с версии ADBM 2.4.0
Database size	Полный объем несжатых данных БД на сегменте
Backup size	Объем сжатых данных БД на сегменте

Секция **Statuses** содержит историю изменения статуса бэкапа.

Statuses

Date ↓	Status	Action
26/08/2025 16:41:21	Init	Backup
26/08/2025 16:41:46	Running	Backup
26/08/2025 16:41:46	Done	Backup

Секция Statuses

Поля секции Statuses

Поле	Описание
Date	Время изменения статуса бэкапа в формате `DD/MM/YYYY HH:mm:ss`
Status	Статус бэкапа. См. *Status* выше
Action	Название действия, в рамках которого изменен статус бэкапа. Для перехода к просмотру деталей действия можно нажать на его имя

Просмотр восстановлений БД в ADBM

© Дарья Борисова

Содержание

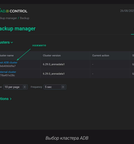
Назад к содержанию

Список восстановлений БД
Детали восстановления БД

Список восстановлений БД

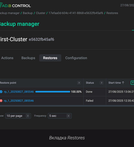
Все операции восстановления БД, запущенные пользователем в ADBM (в рамках действий Common restore, Partial restore или Disaster recovery), отображаются на вкладке **Restore**. Чтобы открыть эту вкладку, выполните шаги:

1. Открыть страницу Backup manager в web-интерфейсе ADB Control.
2. Нажмите на имя кластера ADB в таблице, расположенной в секции **Clusters** открывшейся страницы.



Выбор кластера ADB

3. Выберите вкладку **Restores**.



Вкладка Restores

Открывшаяся страница выводит следующую информацию об операциях восстановления данных.

Поля списка восстановлений БД

Поле	Описание
Restore point	Выбранная для восстановления данных restore point (точка восстановления). Начиная с версии ADBM 2.5.0 отрезка от названия точек восстановления выводится индикатор времени выполнения восстановления данных (в % от рассчитанной продолжительности операции на всех сегментах кластера). Индикатор выводится только для операций восстановления с типом `Self-restore`
Status	Статус восстановления данных. Возможные значения: • `Init` – запуск восстановления инициализирован; • `Running` – восстановление данных выполняется; • `Stopped` – восстановление данных остановлено; • `Failed` – восстановление данных завершено с ошибкой; • `Done` – восстановление данных завершено успешно.
Start time	Время запуска восстановления в формате `DD/MM/YYYY HH:mm:ss`
Update time	Время последнего изменения статуса восстановления в формате `DD/MM/YYYY HH:mm:ss`
Restore type	Тип восстановления данных. Возможные значения: • `Self-restore` – стандартное восстановление данных после запуска действия Common restore; • `Table-restore` – частичное восстановление данных (на уровне таблиц) после запуска действия Partial restore; • `Copy` – восстановление при копировании данных между кластерами в рамках Disaster recovery с типом `Copy` (cold standby); • `Streaming` – восстановление при копировании данных между кластерами в рамках Disaster recovery с типом `Streaming` (warm standby).

В зеленовом столбце таблицы по списком восстановлений расположены фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо нажать на иконку . Для тех столбцов, где набор возможных значений ограничен (например, **Status**), в фильтре можно выбрать значение из выпадающего списка. Для некоторых столбцов (например, **Restore point**) требуется ввести искомое значение. Для столбцов, показывающих дату и время (например, **Start time**), временной диапазон выбирать из календаря.

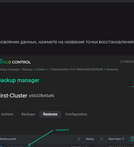
Иконка означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

ВАЖНО

В настоящее время таблица, расположенная на вкладке **Restores**, по умолчанию показывает восстановления данных за текущую дату. Чтобы вывести данные за все время, воспользуйтесь фильтрами.

Детали восстановления БД

Чтобы просмотреть детальную информацию о восстановлении данных, нажмите на название точки восстановления (**Restore point**) в таблице на вкладке **Restores**.



Выбор восстановления БД

Следующая страница показывает информацию о выбранной операции восстановления БД. На ней выведется:

- Таблица с основной информацией.
- Вкладка **Parameters** с подробной информацией о параметрах восстановления.
- Вкладка **Tables** с информацией о восстановленных таблицах.
- Секция **Topology** с информацией о топологии кластера.

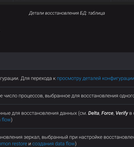
ПРИМЕЧАНИЕ

- Вкладка **Tables** отображается только для действий Partial restore начиная с версии ADBM 2.2.2.
- Секция **Topology** отображается начиная с версии ADBM 2.6.0.

В таблице выводятся следующие данные:

Поля таблицы

Поле	Описание
Status	Статус восстановления данных. См. **Status** выше
Restore type	Тип восстановления данных. См. **Restore type** выше
Start time	Время запуска восстановления данных в формате `DD/MM/YYYY HH:mm:ss`
Update time	Время последнего изменения статуса восстановления в формате `DD/MM/YYYY HH:mm:ss`

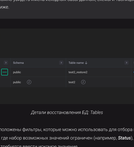


Детали восстановления БД: таблица

На вкладке **Parameters** выводятся следующие данные:

Поля вкладки Parameters

Поле	Описание
Configuration	Версия конфигурации. Для перехода к просмотру деталей конфигурации можно нажать на ее номер
Processes	Максимальное число процессов, выбранное для восстановления одного сегмента
Flags	Флаги, выбранные для восстановления данных (см. *Delta*, *Force*, *Verify* в описании действия Common restore и создание data flow)
Mirrors restore mode	Режим восстановления зеркал, выбранный при настройке восстановления данных (см. **Restore mirrors** в описании действия Common restore и создание data flow)
Action	Название действия, в рамках которого было запущено восстановление данных. Для перехода к просмотру деталей действия можно нажать на его название
Backup	Наименование последнего бэкапа, сформированного перед добавлением выбранной для восстановления restore point. Для перехода к просмотру деталей бэкапа можно нажать на его название
Type	Тип последнего бэкапа, сформированного перед добавлением выбранной для восстановления restore point
Date	Время формирования последнего бэкапа перед добавлением выбранной для восстановления restore point
Table suffix	Суффикс, добавленный к именам таблиц в целевой БД в процессе восстановления данных. Для переноса в ADBM 2.2.2
Target database	Имя целевой БД, в которую восстановлены таблицы. Поле отображается только для действий Partial restore начиная с версии ADBM 2.1.1
Databases	Список исходных БД, которые были выбраны для восстановления данных (из их бэкапов)



Детали восстановления БД: Parameters

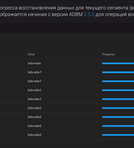
На вкладке **Tables** выводятся следующие данные:

Поля вкладки Tables

Поле	Описание
Database	Имя базы данных, в которую восстановлена таблица
Schema	Имя схемы, в которую восстановлена таблица
Table name	Имя таблицы после восстановления данных
Status	Статус восстановления данных таблицы. См. **Status** выше

РЕКОМЕНДАЦИЯ

После нажатия на иконку в столбце **Database** можно увидеть имена исходных баз данных, схемы и таблицы, из которых восстановлена текущая таблица – в новой строке, которая отображается ниже.



Детали восстановления БД: Tables

В зеленовом столбце таблицы на вкладке **Tables** расположены фильтры, которые можно использовать для отбора необходимых данных. Для открытия фильтра необходимо нажать на иконку . Для тех столбцов, где набор возможных значений ограничен (например, **Status**), в фильтре можно выбрать значение из выпадающего списка. Для некоторых столбцов (например, **Database**) требуется ввести искомое значение.

Иконка означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

В секции **Topology** выводятся следующие данные:

Поля секции Topology

Поле	Описание
Segment	Уникальный идентификатор сегмента данных (content-идентификатор). Соответствует значению `gp_segment_configuration.content` для текущего сегмента
Host	Имя хоста
Progress	Индикатор прогресса восстановления данных для текущего сегмента (в % от рассчитанной продолжительности операции). Отображается начиная с версии ADBM 2.6.0 для операций восстановления с типом `Self-restore`

Детали восстановления БД: Topology

Содержание
Общие шаблоны
Шаблоны для сообщений
Шаблоны для сообщений
Шаблоны для сообщений

Просмотр действий в ADBM

Д. Дорин Бородьков

Содержание

Назад к содержанию

Список действий
Детали действия

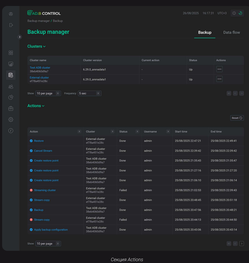
В ADBM **действием (action)** означает API-вызов, SQL-запрос либо внутреннюю процедуру, которую запускается при применении настроек конфигурации либо во время операций, связанных с Kanopsus или DR.

Каждое действие может запустить несколько взаимосвязанных действий. В этом случае порождённые действием называются **subactions**.

Список действий

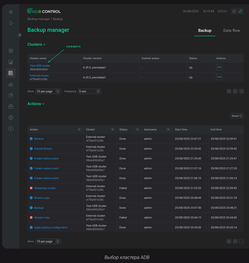
Для просмотра списка действий выполните следующие шаги:

1. Откройте страницу Backup manager в веб-интерфейсе ADB Control. Секция **Actions** открывающейся страницы содержит таблицу с информацией по действиям для всех доступных кластеров ADB.



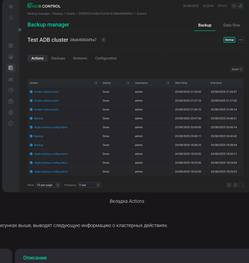
Секция Actions

2. Если необходимо отобразить действия для конкретного кластера, кликните по его названию в таблице кластеров, которая расположена в секции **Clusters** открывающейся страницы.



Выбор кластера ADB

Вкладка **Actions** следующей страницы содержит таблицу с основными данными по действиям внутри выбранного кластера.



Вкладка Actions

Таблицы, изображенные на рисунках выше, выводят следующую информацию о кластерных действиях.

Поля списка действий

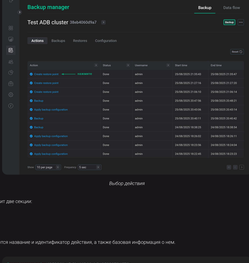
Поле	Описание
Action	Полное наименование действия
Cluster	Имя кластера, в котором выполнялось текущее действие. Отображается, если конкретный кластер не выбран в списке на странице **Backup manager**
Status	Статус действия. Возможные значения: • `Started` — выполнение действия запущено; • `Stopped` — выполнение действия остановлено; • `Done` — действие завершено успешно; • `Failed` — действие завершено с ошибкой; • `Terminating` — выполняется прерывание выполнения действия; • `Awaiting` — действие в режиме ожидания; • `Unknown` — статус действия неизвестен; • `Warning` — в ходе выполнения действия возникли некоторые проблемы, однако они некритичны для верхнеуровневого (родительского) действия.
Username	Имя пользователя, выполнившего действие
Start time	Время начала выполнения действия в формате `DD/MM/YYYY HH:mm:ss`
End time	Время завершения выполнения действия в формате `DD/MM/YYYY HH:mm:ss`

В заголовках столбцов таблиц со списком действий как в секции Actions, так и на вкладке Actions, расположены фильтры, которыми можно воспользоваться для отбора необходимых данных. Для открытия фильтра необходимо навести на иконку ▼. Для тех столбцов, где имеются возможные значения (например, **Action**), в фильтре можно выбрать значение из выпадающего списка. Для некоторых столбцов (например, **Username**) требуется ввести искомое значение.

Иконка ▼ означает, что для столбца определен фильтр. Для сброса всех фильтров нажмите **Reset**.

Детали действия

Чтобы просмотреть детальную информацию о действии, нажмите на название действия (**Action**) в таблице, расположенной в секции Actions или на вкладке Actions.



Выбор действия

Следующая страница содержит две секции:

- **Overview**
- **Subactions**

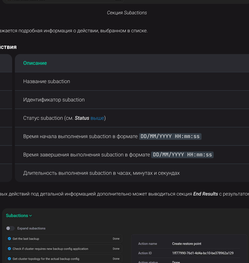
В секции **Overview** отображаются название и идентификатор действия, а также базовая информация о нем.

Секция Overview

Поля секции Overview

Поле	Описание
Cluster	Имя кластера (см. **Cluster** выше)
Submitted by	Имя пользователя, выполнившего выбранное действие
Start time	Время начала выполнения выбранного действия в формате `DD/MM/YYYY HH:mm:ss`
End time	Время завершения выполнения выбранного действия в формате `DD/MM/YYYY HH:mm:ss`
Status	Статус выбранного действия (см. **Status** выше)

Секция **Subactions** отображает список действий, порожденных вернеуровневым действием (subaction). Subactions упорядочены по дате начала выполнения — от самого первого к последнему. Для каждого действия в списке выводится название и статус.



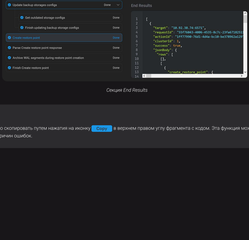
Секция Subactions

В правой части экрана отображается подробная информация о действии, выбранном в списке.

Детали выбранного действия

Поле	Описание
Action name	Название subaction
Action ID	Идентификатор subaction
Action status	Статус subaction (см. **Status** выше)
Start time	Время начала выполнения subaction в формате `DD/MM/YYYY HH:mm:ss`
End time	Время завершения выполнения subaction в формате `DD/MM/YYYY HH:mm:ss`
Duration	Длительность выполнения subaction в часах, минутах и секундах

Для некоторых нижнеуровневых действий под детальной информацией дополнительно может выводиться секция *End Results* с результатом выполнения.



Секция End Results

РЕКОМЕНДАЦИЯ

Результат действия можно скопировать путем нажатия на иконку Copy в верхнем правом углу фрагмента с кодом. Эта функция может быть полезна для выяснения причин ошибок.

Обзор ADB ClickHouse Connector

Дарья Барышева

Назад к содержанию

В отличие от предыдущей версии коннектора Tkhemali 1.X, основанной на механизме внешних (external) таблиц, ADB ClickHouse Connector реализован на базе обертки сторонних данных (foreign data wrapper) и сторонних (foreign) таблиц. Благодаря этому новая версия коннектора к ADQM/ClickHouse обладает следующими функциональными возможностями:

- **Более удобная и безопасная схема управления учетными данными**. Если в предыдущей версии коннектора учетные данные пользователей необходимо было прописывать в параметре `LOCATION` на уровне каждой внешней таблицы либо в файле в явном виде, в ADB ClickHouse Connector логин и пароль пользователя хранятся в отдельном объекте базы данных ADB — сопоставлении пользователя (user mapping). Сопоставление пользователя достаточно описать один раз для одного кластера ClickHouse, с которым планируется соединение. При необходимости данные можно изменять с помощью команды `ALTER USER MAPPING`.
- **Многоуровневые настройки соединения**. Общие настройки, которые будут одинаковы для всех или большинства целевых таблиц ClickHouse (стратегия распределения нагрузки, необходимость использования staging-слоя, ограничения для батчей данных и так далее) можно указать на уровне сервера (server), а частные настройки (имя конкретной таблицы) — на уровне сторонней таблицы. В предыдущей версии коннектора абсолютно все настройки требовалось указывать на уровне внешней (external) таблицы.
- **Обновленный механизм распределения нагрузки по хостам**. В новой версии коннектора список хостов ClickHouse, через которые будет производиться загрузка данных из ADB, указывается в опции `hosts`. Отдельная опция `distribution_type` отвечает за выбор стратегии распределения нагрузки между выбранными хостами. Поддерживаются два типа распределения: случайное и с использованием алгоритма round-robin. Второй способ предпочтительнее в плане равномерности нагрузки. Наряду с этим, при записи в распределенные (distributed) таблицы возможно указать имя кластера ClickHouse в опции `cluster` на стороне foreign-таблицы — в этом случае перечислять явно все хосты не нужно, достаточно указать одну точку входа, а полный список хостов будет получен автоматически из метаданных кластера. В предыдущей версии коннектора было также возможно получать список хостов для distributed-таблиц, однако нагрузка на хосты кластера ClickHouse могла быть неравномерной.
- **Гибкое управление кастомными настройками**. Кастомные настройки подключения и выполнения запросов к ClickHouse теперь можно объявить в опции `clickhouse_properties` в следующем формате: `<custom_option>=<value>;[…]`. Для изменения настроек достаточно выполнить команду `ALTER SERVER` или `ALTER FOREIGN TABLE`. В предыдущей версии коннектора кастомные настройки для ClickHouse хранились в XML-файле, который необходимо было копировать на все сегмент-хосты ADB.
- **Больший контроль над освобождением ресурсов**. Обертка сторонних данных поддерживает транзакции "из коробки", позволяя подписываться на хуки управления транзакциями. Благодаря этому новая версия коннектора обеспечивает больший контроль над ходом выполнения запросов и освобождением ресурсов, что особенно актуально для транзакционной загрузки данных с использованием staging-таблиц. В предыдущей версии коннектора при определенных обстоятельствах staging-слой мог не удаляться и было необходимо удалять его вручную.
- **Упрощение синтаксиса запросов**. Исключена необходимость использования функции `txn` для включения транзакционного режима (использования staging-слоя). В новой версии коннектора для передачи данных из ADB в ClickHouse достаточно выполнить обычный запрос `INSERT` к соответствующей сторонней таблице. За создание staging-таблиц теперь отвечает отдельный флаг `use_staging`. Наряду с этим, в предыдущей версии коннектора было необходимо указывать полное имя таблицы staging при создании external-таблицы. В ADB ClickHouse Connector требуется заполнить только имя целевой таблицы в ClickHouse — в опции `resource` при создании сторонней таблицы. Имена staging-таблиц формируются автоматически на основе шаблона, указанного в опции `staging_table_name_format`.

ВАЖНО

- ADB ClickHouse Connector доступен только в Enterprise-версии ADB:
  - В ADB 7 — начиная с версии 7.4.0.
  - В ADB 6 — начиная с версии 6.23.3.44.
- Начиная с версии 6.24.3.47 в ADB 6 и 7.4.0 в ADB 7 ClickHouse Connector поддерживает как запись, так и чтение данных из ADQM/ClickHouse. Однако, чтобы использовать стороннюю таблицу в ADB для чтения, в ней необходимо определить имена столбцов, точно совпадающие с соответствующими именами в таблице ADQM/ClickHouse.

Верхнеуровневая архитектурная схема новой версии коннектора приведена ниже.

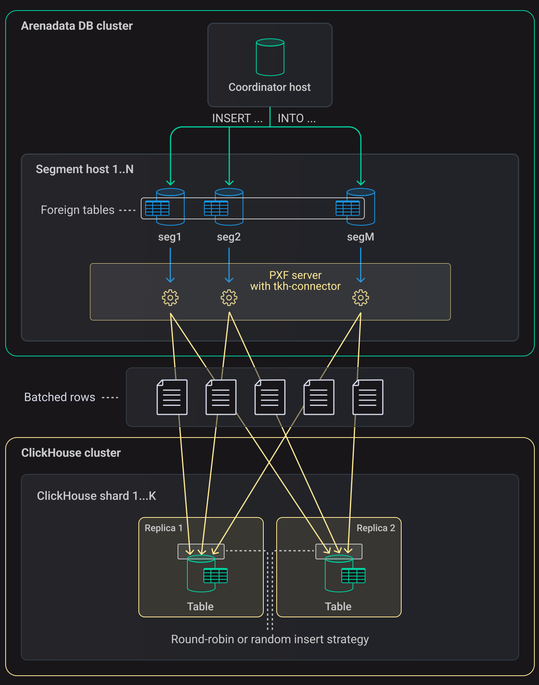


Схема выполнения запроса INSERT

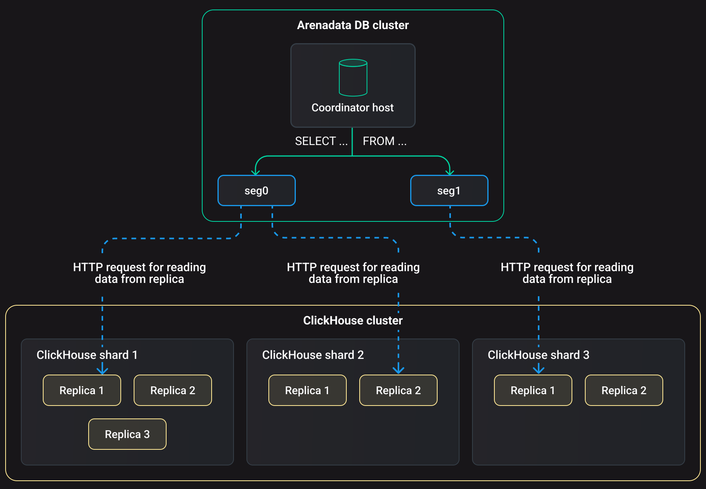


Схема выполнения запроса SELECT

Установка ADB ClickHouse Connector

Дарья Барышева

Назад к содержанию

Для установки ADB ClickHouse Connector выполните следующие шаги:

1. Убедитесь, что сервисы PXF и ADB ClickHouse Connector установлены в кластере ADB. Информацию о добавлении сервисов в кластер можно получить в статье Добавление сервисов.

2. Добавьте новый кастомный профиль с именем `Tkh_clickhouse:text` в конфигурационный файл PXF */var/lib/pxf/conf/pxf-profiles.xml* на **хосте координатора**. Для редактирования файла можно воспользоваться командой `vi` или `vim`:

```
$ sudo vi /var/lib/pxf/conf/pxf-profiles.xml
```

Содержимое отредактированного файла может выглядеть следующим образом. В представленном примере другие профили в файле отсутствуют:

```
<profiles>
    <profile>
        <name>Tkh_clickhouse:text</name>
        <description>Clickhouse</description>
        <plugins>
            <accessor>io.arenadata.tkh.pxf.TkhFdwAccessor</accessor>
            <resolver>io.arenadata.tkh.pxf.TkhResolver</resolver>
        </plugins>
    </profile>
</profiles>
```

ПРИМЕЧАНИЕ

Директория, в которой расположены конфигурационные файлы PXF, определяется переменной окружения `PXF_BASE`. Значение переменной по умолчанию — */var/lib/pxf*. В общем случае путь может быть иным.

3. Подключитесь к хосту координатора ADB под пользователем `gpadmin`, который создается по умолчанию. Все команды, приведенные на шагах ниже, запускаются исключительно на **хосте координатора**:

```
$ sudo su - gpadmin
```

4. Для синхронизации конфигурационных файлов PXF на всех хостах кластера ADB выполните команду:

```
$ pxf cluster sync
```

Результат:

```
Syncing PXF configuration files from coordinator host to standby coordinator host and 2 segment hosts...
PXF configs synced successfully on 3 out of 3 hosts
```

5. Перезапустите сервис PXF:

```
$ pxf cluster restart
```

Результат:

```
Restarting PXF on coordinator host, standby coordinator host, and 2 segment hosts...
PXF restarted successfully on 4 out of 4 hosts
```

6. Подключитесь к базе данных через `psql` (или иную клиентскую программу):

```
$ psql adb
```

7. Убедитесь, что расширения `pxf_fdw` и `tkh_fdw` успешно добавлены. Для этого можно использовать команду `psql` `\dx`:

```
\dx
```

Результат содержит расширения `pxf_fdw` и `tkh_fdw`:

```
                                    List of installed extensions
       Name        | Version |   Schema   |                              Description
-------------------+---------+------------+------------------------------------------------------------------
 adb_fdw           | 1.0     | public     | foreign-data wrapper for remote ADB servers
 arenadata_toolkit | 1.0     | public     | extension is used for manipulation of objects created by adb-bundle
 dblink            | 1.2     | public     | connect to other PostgreSQL databases from within a database
 gp_exttable_fdw   | 1.0     | pg_catalog | External Table Foreign Data Wrapper for Greengage
 gp_toolkit        | 1.8     | gp_toolkit | various GPDB administrative views/functions
 kadb_fdw          | 0.16    | kadb       | Kafka-ADB foreign data wrapper
 plpgsql           | 1.0     | pg_catalog | PL/pgSQL procedural language
 pxf               | 2.1     | public     | Extension which allows to access unmanaged data
 pxf_fdw           | 1.0     | public     | PXF Foreign Data Wrapper for Greengage
 tkh_fdw           | 1.0     | public     | ClickHouse Foreign Data Wrapper for Greenplum
(10 rows)
```

8. Также можно проверить наличие новой обертки сторонних данных (foreign data wrapper) `tkh_fdw`, обратившись к таблице системного каталога `pg_catalog.pg_foreign_data_wrapper`:

```
SELECT * FROM pg_catalog.pg_foreign_data_wrapper;
```

Результат:

```
  oid  |     fdwname      | fdwowner | fdwhandler | fdwvalidator | fdwacl |                                        fdwoptions
-------+------------------+----------+------------+--------------+--------+-----------------------------------------------------
-------------------------------------
 13702 | gp_exttable_fdw |       10 |      13700 |        13701 |        |
 17124 | dblink_fdw       |       10 |          0 |        17123 |        |
 17175 | adb_fdw          |       10 |      17173 |        17174 |        |
 17187 | jdbc_pxf_fdw     |       10 |      17185 |        17186 |        | {protocol=jdbc,"mpp_execute=all segments"}
 17188 | hdfs_pxf_fdw     |       10 |      17185 |        17186 |        | {protocol=hdfs,"mpp_execute=all segments"}
 17189 | hive_pxf_fdw     |       10 |      17185 |        17186 |        | {protocol=hive,"mpp_execute=all segments"}
 17190 | hbase_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=hbase,"mpp_execute=all segments"}
 17191 | s3_pxf_fdw       |       10 |      17185 |        17186 |        | {protocol=s3,"mpp_execute=all segments"}
 17192 | gs_pxf_fdw       |       10 |      17185 |        17186 |        | {protocol=gs,"mpp_execute=all segments"}
 17193 | adl_pxf_fdw      |       10 |      17185 |        17186 |        | {protocol=adl,"mpp_execute=all segments"}
 17194 | wasbs_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=wasbs,"mpp_execute=all segments"}
 17195 | file_pxf_fdw     |       10 |      17185 |        17186 |        | {protocol=file,"mpp_execute=all segments"}
 17201 | tkh_fdw          |       10 |      17200 |        17199 |        | {protocol=tkh_clickhouse,"mpp_execute=all
segments",format=text}
 17210 | kadb_fdw         |       10 |      17209 |        17211 |        | {"mpp_execute=all
segments",k_initial_offset=0,k_automatic_offsets=true,k_latency_ms=2000}
(14 rows)
```

Настройка ADB ClickHouse Connector

Д. Дарья Карповна

Содержание

Назад к содержанию

- Сервер
- Сопоставление пользователя
- Сторонняя таблица

Для отправки данных из ADB в ClickHouse через ADB ClickHouse Connector необходимо предварительно создать следующие объекты на стороне кластера ADB:

1. Server (сервер) — аналогичный информацию о соединении с внешним источником данных;
2. User mapping (сопоставление пользователя) — обеспечивает аутентификацию пользователя во внешнем источнике данных;
3. Foreign table (сторонняя таблица) — таблица ADB, определяющая структуру внешних данных. Сторонняя таблица не хранит данные в ADB, но к ней можно обращаться с помощью запросов как к обычной таблице.

Примеры использования перечисленных объектов для отправки данных из ADB в ROLAP приведены в статье Примеры использования ADB ClickHouse Connector.

> **ВАЖНО**
>
> - Для каждого кластера ClickHouse, к которой планируется отправлять запросы из ADB, достаточно создать один сервер и одно сопоставление пользователя. Сторонних таблиц может быть несколько, если в ClickHouse хранятся несвязанные таблицы ClickHouse.
> - Все опции, приведенные в таблицах ниже (Опции сервера, Опции сопоставления пользователя, Опции сторонней таблицы), при объявлении объектов расположатся в стороковом виде (в одинарных кавычках '...'). Столбцы **Тип** в таблицах показывают, какой тип данных будет использоваться при валидации введенных значений.

Сервер

Для создания сервера предназначена команда `CREATE SERVER`. Базовый синтаксис которой приведен ниже:

```
CREATE SERVER <server_name> [ TYPE '<server_type>' ] [ VERSION '<server_version>' ]
  FOREIGN DATA WRAPPER <fdw_name>
  [ OPTIONS ( [ <option> '<value>' [, ... ] ] ) ]
```

где:

- `<server_name>` — имя сервера в ADB. Должно быть уникальным в рамках текущей базы данных ADB;
- `<server_type>` — необязательный параметр, определяющий тип сервера;
- `<server_version>` — необязательный параметр, определяющий версию сервера;
- `<fdw_name>` — имя обертки сторонних данных (foreign data wrapper). Необходимо указать `CH_fdw` — foreign data wrapper, который создается автоматически после установки коннектора (см. шаг 8 в статье Установка ADB ClickHouse Connector);
- `<option>` — параметры сервера, определяющие детали подключения к внешнему источнику данных. Описание ADB ClickHouse Connector приведен в таблице Опции сервера ниже. Обратите внимание, что эти опции могут быть определены как на уровне сервера, так и на уровне сторонней таблицы. При этом опции, отмеченные как обязательные, должны быть указаны на одном из уровней;
- `<value>` — значение соответствующее параметром `<option>`.

> **ПРИМЕЧАНИЕ**
>
> - Чтобы создать сервер, пользователю необходимо привилегия `USAGE` для обертки сторонних данных `CH_fdw`. Пользователь, создавший сервер, становится его владельцем.
> - Полную версию синтаксиса команды `CREATE SERVER` можно посмотреть в документации Greenplum.
> - Для редактирования параметров сервера предназначена команда `ALTER SERVER`, для удаления — `DROP SERVER`.

Опции сервера

Имя	Тип	Описание	Default	Обязательность
database	TEXT	Имя базы данных в ClickHouse, в которой расположена целевая таблица. Должна соответствовать шаблону выражению `^[a-zA-Z0-9_]+$`.	—	Да
hosts	TEXT	Список хостов в ClickHouse, на которых расположена целевая таблица. Поскольку кластер ClickHouse не имеет единого узла-координатора, запросы могут быть направлены случайным или круговым способом на любой из хостов кластера, для снижения нагрузки отправка бегчей данных в ClickHouse осуществляется согласно выбранной стратегии `random` или `round-robin` (см. параметр `distribution_type` ниже). Правила валидации параметра `hosts` см. ниже	—	Да
lines_batch_size	INT	Максимальное количество строк в бетче данных, отправляемых в ClickHouse со стороны коннектора. Для ввода допускается целые положительные числа	100000	Нет
bytes_batch_size_mb	INT	Максимальный размер бетча данных, отправляемых в ClickHouse со стороны коннектора (в МБ). При одновременном указании с параметром `lines_batch_size` будет использоваться значение `bytes_batch_size_mb`. Для ввода допускается целые положительные числа	—	Нет
send_compressed	BOOL	Флаг, указывающий на необходимость компрессии данных на стороне коннектора перед отправкой в ClickHouse. Возможные значения: - `true` — использовать сжатие данных. - `false` — не использовать сжатие данных.	false	Нет
send_delay	INT	Задержка между запросами со стороны коннектора к ClickHouse (в миллисекундах). Для ввода допускается целые положительные числа	300	Нет
insert_distributed_sync	BOOL	Включает или отключает режим синхронного добавления данных в распределенные (Distributed) таблицы. Возможные значения: - `true` — данные вставляются синхронно, а запрос `INSERT` считается выполненным успешно, когда все данные записаны на все шарды. Это крайний шаг на пути репликации для каждого шарда, если параметр ClickHouse `internal_replication` = `true`; - `false` — данные вставляются асинхронно.	true	Нет
distribution_type	TEXT	Определяет стратегию распределения нагрузки между хостами ClickHouse (см. `hosts` выше) при выполнении операций вставки данных со стороны коннектора. Возможные значения: - `random` — случайный выбор хоста из списка; - `round-robin` — выбор хоста из списка с последовательным перебором всех. В этом равномерном распределении нагрузки этот вариант предпочтительнее.	random	Нет
use_staging	BOOL	Флаг, указывающий на необходимость создания коннектором стадинг таблицы на стороне ClickHouse перед выполнением вставки данных в целевую таблицу. Используется для эмуляции транзакций, которые отсутствуют в ClickHouse. Staging таблицы поддерживаются только для следующих движков в ClickHouse: MergeTree и ReplicatedMergeTree. Для остальных движков будет возвращаться ошибка при создании стадинг-окружения, так как для всех типов движков можно реализовать подключение партиций в ClickHouse. Возможные значения: - `true` — использовать staging таблицы; - `false` — не использовать staging таблицы. В этом случае вставка данных будет идти напрямую в целевую таблицу, а атомарность добавления данных может быть нарушена. Данный режим подходит для разовых вставок в пустые целевые таблицы.	—	Да, если `use_staging` = `true`
staging_table_name_form at	TEXT	Определяет формат наименования стадинг таблицы (см `use_staging` выше). Поле `%(table)` — первая подстановочная строка (обязательна), `%(pid)` — вторая подстановочная строка (опцио нальна). Для `%(pid)` — рекомендуется использовать `%(table)` — таблица `pid` (целочисленный идентификатор, автоматически генерируемый на хосте кластера ADB	—	Да, если `use_staging` = `true`
clickhouse_properties	TEXT	Перечень кастомных опций ClickHouse в формате `<custom_option>=<value>;[...]` (см `<custom_option>` — имя опции, `<value>` — значение опции	—	Нет
fdw_startup_cost	DOUBLE	Оценка стоимости получения первой строки от источника данных. Для ввода допускается дробные положительные числа. Опция может быть использована только в запросах `SELECT` к ADCM/ClickHouse	1000.0	Нет
fdw_tuple_cost	DOUBLE	Оценка стоимости получения целой строки от источника данных. Для ввода допускается дробные положительные числа. Опция может быть использована только в запросах `SELECT` к ADCM/ClickHouse	0.01	Нет
fdw_tuples_count	DOUBLE	Оценка количества всех строк в источнике данных. Была учета оптимизатором запросов, условие `LIMIT` и т д. Для ввода допускается дробные положительные числа >= `1`. Опция может быть использована только в запросах `SELECT` к ADCM/ClickHouse	100000.0	Нет

> **Правила валидации опции hosts**
>
> - Список хостов указывается в следующем формате `<hostname>:<port> [,...]`, где `<hostname>` — имя каждого хоста, `<port>` — номер HTTP-порта для подключения к соответствующему хосту.
> - Максимальная длина имени каждого хоста `<hostname>` — 255 символов.
> - В `<hostname>` допускаются только латинские буквы `A-Z`, ASCII символы `a-z-.2` дефис `-`, нижнее подчеркивание `_` и точка `.`.
> - Значение для каждого `<hostname>` должно начинаться с буквы `<port>` обязательно. В номере порта допускаются только латинские буквы `0-9`. Максимальная длина номера порта `<port>` — 5 символов.
> - Максимальная длина строки `hosts` (без максимальных длин `<hostname>` + 1 символ `:` + 5 максимальная длина `<port>` + 1 разделитель `,`) = 127 (число хостов) × 262 (6 символов)

Сопоставление пользователя

Для создания сопоставления пользователя предназначена команда `CREATE USER MAPPING`. Базовый синтаксис которой приведен ниже:

```
CREATE USER MAPPING FOR <username> [ USER [ CURRENT_USER | PUBLIC ]
  SERVER <server_name>
  [ OPTIONS ( <option> '<value>' [, ... ] ) ]
```

где:

- `<username>` — имя пользователя в базе данных ADB. Например, с левым указанием пользователя, допускается ввод следующих констант:
  - `USER` или `CURRENT_USER` — текущий пользователь, устанавливающий соединение с ADB;
  - `PUBLIC` — все существующие и будущие пользователи ADB.
- `<server_name>` — имя сервера.
- `<option>` — параметры сопоставления пользователя. Как правило, включают имя пользователя и пароль для подключения к внешнему источнику данных. Список возможных опций для ADB ClickHouse Connector приведен в таблице Опции сопоставления пользователя ниже.
- `<value>` — значение соответствующее параметром `<option>`.

> **ПРИМЕЧАНИЕ**
>
> - Чтобы создать сопоставление пользователя, необходимо быть владельцем соответствующего сервера.
> - Полную версию синтаксиса команды `CREATE USER MAPPING` можно посмотреть в документации Greenplum.
> - Для редактирования параметров сопоставления пользователя предназначена команда `ALTER USER MAPPING`, для удаления — `DROP USER MAPPING`.

Опции сопоставления пользователя

Имя	Тип	Описание	Default	Обязательность
clickhouse_username	TEXT	Имя пользователя в ClickHouse	—	Да
clickhouse_password	TEXT	Пароль пользователя в ClickHouse. В случае отсутствия пароля (например, в тестовом окружении) указывается пустая строка		Нет

Сторонняя таблица

Для создания сторонней таблицы предназначена команда `CREATE FOREIGN TABLE`. Базовый синтаксис которой приведен ниже:

```
CREATE FOREIGN TABLE [ IF NOT EXISTS ] <table_name> ( [
  <column_name> <data_type> [ COLLATE <collation> ] [ <column_constraint> [ ... ] ] [
  [, ... ]
] ) [
  SERVER <server_name>
  [ OPTIONS ( <option> '<value>' [, ... ] ) ]
```

где:

- `<table_name>` — имя сторонней таблицы в ADB;
- `<column_name>` — имя столбца;
- `<data_type>` — тип данных столбца;
- `<collation>` — используемые для столбца сортировка (collation);
- `<column_constraint>` — ограничение (constraint), определенное для данного столбца. Или ограничение `<constraint_name>` указывается опционально. Синтаксис:

```
[ CONSTRAINT <constraint_name> ]
{ NOT NULL |
  NULL |
  DEFAULT <default_expr> }
```

Возможные ограничения:

- `NOT NULL` — указывает, что столбец не может содержать пуй значений.
- `NULL` — указывает, что столбец может содержать пуй значения. Это ограничение используется по умолчанию (если не указано `NOT NULL`).
- `DEFAULT` — определяет для столбца значение по умолчанию `<default_expr>`.

- `<server_name>` — имя сервера.
- `<option>` — параметры сторонней таблицы. Для ADB ClickHouse Connector все опции, определенные на уровне сервера, могут быть переопределены на уровне сторонних таблиц (частично или полностью). При одновременном указании какой-либо опции для сервера и таблицы таблицы имеет больший приоритет. Дополнительно на уровне сторонней таблицы объявляется еще две опции, приведенные в таблице Опции сторонней таблицы ниже.
- `<value>` — значение соответствующее параметром `<option>`.

> **ПРИМЕЧАНИЕ**
>
> - Чтобы создать сторонную таблицу, пользователю необходимо привилегии `USAGE` для обертки сторонних данных `CH_fdw`, а также для всех типов данных, используемых в таблице. Пользователь, создавший сторонную таблицу, становится ее владельцем.
> - Полную версию синтаксиса команды `CREATE FOREIGN TABLE` можно посмотреть в документации Greenplum.
> - Для редактирования параметров сторонней таблицы предназначена команда `ALTER FOREIGN TABLE`, для удаления — `DROP FOREIGN TABLE`.

Опции сторонней таблицы

Имя	Тип	Описание	Default	Обязательность
resource	TEXT	Имя таблицы на стороне ClickHouse. Если вам необходимо реализовать собственную стратегию распределения данных по кластеру ClickHouse, вы можете использовать вставку через распределенные (Distributed) таблицы ClickHouse с настроенным ключом шардирования, указав в имя в опции `resource` При записи данных в distributed таблицу важно помнить о том, что таблицы фактические (локальные) таблицы присутствовали на всех хостах кластера ClickHouse, в противном случае коннектор вернет ошибку. Значение опции должно соответствовать регулярному выражению `^[a-zA-Z0-9_]+$`.	—	Да
cluster	TEXT	Имя кластера в ClickHouse. Для получения дополнительной информации можно обратиться к статье Creating a cluster в документации ADCM. При указании опции `cluster` список хостов ClickHouse `hosts` должен быть отредактирован на уровне сторонней таблицы и состоять из одного элемента `<hostname>:<port>` — даже в случае, если он предварительно объявлен на уровне сервера. Иначе может возникнуть неадекватная ситуация, когда из уровня сервера будет выбран хост, который не принадлежит кластеру, а фактическая будет производиться с хостами, перечисленными на настройках кластера ClickHouse. Опция `cluster` необходима в первую очередь для удобства использования: при ее использовании достаточно указать имя кластера и одну точку входа, полный список хостов будет получен со стороны ClickHouse. При этом нагрузка между хостами будет по-прежнему распределяться согласно выбранной стратегии `distribution_type` (см. таблицу Опции сервера) Максимальная длина имени кластера — 128 символов.	—	Нет

Примеры использования ADB ClickHouse Connector

Дарья Барышева

Содержание

Назад к содержанию

Запись данных
Чтение данных
Пример корректного чтения данных
Пример с ошибкой

В статье показаны примеры передачи данных между ADB и распределенной (distributed) таблицей в ADQM 25.3.2.39 через ADB ClickHouse Connector. При выполнении примеров соблюдаются следующие условия:

- Кластер ADQM установлен согласно руководству Online-установка на следующих хостах: `dev-adqm-01.ru-central1.internal`, `dev-adqm-02.ru-central1.internal`, `dev-adqm-03.ru-central1.internal`, `dev-adqm-04.ru-central1.internal`.
- База данных с именем `default` существует в ADQM.
- Пользователь с именем `default` существует в ADQM.
- Кластер `default_cluster` определен в секции `remote_servers` конфигурационного файла *config.xml* в ADQM. Для получения дополнительной информации можно обратиться к статье Типовой кластер в документации ADQM.
- Все ноды кластера ADB имеют доступ к HTTP-порту `8123` на всех нодах кластера ADQM, через которые планируется производить загрузку данных.

Запись данных

Чтобы проверить запись данных из кластера ADB в ADQM, выполните следующие шаги:

1. Подключитесь к базе данных `default` кластера ADQM под пользователем `default` (например, через clickhouse-client). Создайте последовательно следующие таблицы:
   - Локальная таблица `test_local`:

```
CREATE TABLE test_local ON CLUSTER default_cluster
(
    id Int32,
    value_string String
)
ENGINE = ReplicatedMergeTree('/clickhouse/tables/{shard}/test_local', '{replica}')
ORDER BY id;
```

   - Распределенная таблица `test_distr`:

```
CREATE TABLE test_distr ON CLUSTER default_cluster AS default.test_local
ENGINE = Distributed(default_cluster, default, test_local, rand());
```

2. Подключитесь к базе данных `adb` кластера ADB под пользователем `gpadmin` (например, через `psql`). Создайте последовательно следующие объекты:
   - Сервер `clickhouse_server`:

```
CREATE SERVER clickhouse_server
FOREIGN DATA WRAPPER tsh_fdw
OPTIONS (
        database 'default',
        hosts 'dev-adqm-01.ru-central1.internal:8123, dev-adqm-02.ru-central1.internal:8123, dev-adqm-03.ru-central1.internal:8123, dev-adqm-04:8123',
        distribution_type 'random',
        lines_batch_size '100000',
        send_compressed 'true',
        send_delay '300',
        insert_distributed_sync 'true',
        use_staging 'true',
        staging_table_name_format '$_tmp_$');
```

   - Сопоставление пользователя `gpadmin`:

```
CREATE USER MAPPING FOR gpadmin
SERVER clickhouse_server
OPTIONS (clickhouse_username 'default', clickhouse_password '');
```

   - Сторонняя таблица `tshemali_test`:

```
CREATE FOREIGN TABLE tshemali_test (id INT, value_string TEXT)
SERVER clickhouse_server
OPTIONS (resource 'test_distr',
        hosts 'dev-adqm-01.ru-central1.internal:8123',
        cluster 'default_cluster');
```

> **ВАЖНО**
>
> Поскольку на табличном уровне определена опция `cluster`, вся работа будет проводиться с хостами, полученными из метаданных кластера ADQM. Поэтому требуется переопределить значение опции `hosts`, указанное для `clickhouse_server` на уровне сервера, и оставить только один хост в значении опции на уровне таблицы.

3. Добавьте данные в таблицу ADB `tshemali_test`:

```
INSERT INTO tshemali_test VALUES
(1, 'test1'),
(2, 'test2'),
(3, 'test3'),
(4, 'test4'),
(5, 'test5'),
(6, 'test6'),
(7, 'test7'),
(8,'test8'),
(9,'test9'),
(10,'test10');
```

   Результат:

```
INSERT 0 10
```

4. Убедитесь, что данные, переданные из ADB, успешно добавлены в ADQM:

```
SELECT * FROM test_distr;
```

   Результат:

```
    ┌─id─┬─value_string─┐
 1. │  5 │ test5        │
 2. │  7 │ test7        │
 3. │  9 │ test9        │
 4. │ 10 │ test10       │
 5. │  3 │ test3        │
 6. │  4 │ test4        │
 7. │  8 │ test8        │
 8. │  1 │ test1        │
 9. │  4 │ test4        │
10. │  2 │ test2        │
    └────┴──────────────┘

10 rows in set. Elapsed: 0.008 sec.
```

Чтение данных

> **ПРИМЕЧАНИЕ**
>
> ADB ClickHouse Connector поддерживает возможность чтения данных из ADQM/ClickHouse начиная с версии 6.24.3.47 в ADB 6 и с версии 7.4.0 в ADB 7. Однако, чтобы использовать стороннюю таблицу в ADB для чтения, в ней необходимо определить имена столбцов, точно совпадающие с соответствующими именами в таблице ADQM/ClickHouse.

Пример корректного чтения данных

Следующий пример демонстрирует чтение данных из кластера ADQM в ADB:

1. Выполните шаги 1–2, приведенные выше (если они не были пройдены ранее).
2. Удалите данные из таблицы `test_local` в ADQM:

```
ALTER TABLE default.test_local ON CLUSTER default_cluster DELETE WHERE true;
```

3. Добавьте данные в таблицу ADQM `test_distr`:

```
INSERT INTO test_distr VALUES
(1, 'test1'),
(2, 'test2'),
(3, 'test3'),
(4, 'test4'),
(5, 'test5'),
(6, 'test6'),
(7, 'test7'),
(8,'test8'),
(9,'test9'),
(10,'test10');
```

   Результат:

```
Query id: 9eec272c-e292-46c5-be17-33bee1b0188e

Ok.

10 rows in set. Elapsed: 0.010 sec.
```

4. Убедитесь, что данные, добавленные в ADQM, могут быть успешно прочитаны из ADB:

```
SELECT * FROM tshemali_test ORDER BY id;
```

   Результат:

```
 id | value_string
----+--------------
  1 | test1
  2 | test2
  3 | test3
  4 | test4
  5 | test5
  6 | test6
  7 | test7
  8 | test8
  9 | test9
 10 | test10
(10 rows)
```

Пример с ошибкой

Приведенный ниже пример показывает, что чтение данных из ADQM/ClickHouse в ADB невозможно в случае, если имена столбцов в сторонней таблице ADB и исходной таблице ADQM/ClickHouse не совпадают:

1. Выполните все шаги из примера с корректным чтением данных (если они не выполнены ранее).
2. Убедитесь, что чтение данных в ADB при текущей структуре сторонней таблицы работает успешно:

```
SELECT * FROM tshemali_test ORDER BY id;
```

   Результат:

```
 id | value_string
----+--------------
  1 | test1
  2 | test2
  3 | test3
  4 | test4
  5 | test5
  6 | test6
  7 | test7
  8 | test8
  9 | test9
 10 | test10
(10 rows)
```

3. Переименуйте часть столбцов сторонней таблицы `tshemali_test` в ADB:

```
ALTER FOREIGN TABLE tshemali_test RENAME value_string TO wrong_string;
```

4. Повторно проверьте возможность чтения данных из таблицы `tshemali_test` в ADB:

```
SELECT * FROM tshemali_test ORDER BY id;
```

   В результате возвращается ошибка:

```
ERROR:  CH easy datum. HTTP error: 404; CH message: Code: 47. DB::Exception: Unknown expression identifier `wrong_string` in scope SELECT id, wrong_string FROM default.test_local. (UNKNOWN_IDENTIFIER) (version 25.3.2.39) (ch_easy_datum.c:141)  (seq1 slice1 10.92.38.12:10000 pid=1648668) (ch_easy_datum.c:141)
```

> **ПРИМЕЧАНИЕ**
>
> Если имена совпадают только у части столбцов сторонней таблицы в ADB и исходной таблицы в ADQM/ClickHouse — запрос успешно отрабатывает, пока в результирующем наборе данных используются только указанные столбцы.

Интеграция ADB и Kafka

Дарья Барышева

Назад к содержанию

Для интеграции ADB и Kafka можно использовать один из коннекторов:

- **ADB to Kafka Connector** — основан на протоколе PXF и механизме внешних (external) таблиц. Реализует загрузку данных из ADB в Kafka. Описание коннектора представлено в статьях:
  - Установка ADB to Kafka Connector
  - Настройка ADB to Kafka Connector
  - Примеры использования ADB to Kafka Connector
- **Kafka to ADB Connector** — основан на механизме обертки сторонних данных (foreign data wrapper). Использует библиотеку librdkafka для взаимодействия с Kafka и реализует транзакционную загрузку данных из Kafka в ADB. Описание коннектора представлено в статьях:
  - Установка Kafka to ADB Connector
  - Настройка Kafka to ADB Connector
  - Примеры использования Kafka to ADB Connector

Установка ADB to Kafka Connector

Дарья Барышева

Назад к содержанию

Для установки ADB to Kafka Connector выполните следующие шаги:

1. Убедитесь, что сервисы PXF и ADB to Kafka установлены в кластере ADB. Информацию о добавлении сервисов в кластер можно получить в статье Добавление сервисов.

2. Добавьте новый кастомный профиль с именем `kafka` в конфигурационный файл PXF */var/lib/pxf/conf/pxf-profiles.xml* на **хосте координатора**. Для редактирования файла можно воспользоваться командой `vi` или `vim`:

```
$ sudo vi /var/lib/pxf/conf/pxf-profiles.xml
```

Содержимое отредактированного файла может выглядеть следующим образом. В представленном примере другие профили в файле отсутствуют:

```
<profiles>
    <profile>
        <name>kafka</name>
        <description>A profile for export data into Apache Kafka</description>
        <plugins>
            <accessor>org.greenplum.pxf.plugins.kafka.KafkaAccessor</accessor>
            <resolver>org.greenplum.pxf.plugins.kafka.KafkaResolver</resolver>
        </plugins>
        <optionMappings>
            <!-- Kafka properties -->
            <mapping option="BOOTSTRAP_SERVERS" property="kafka.bootstrap.servers"/>
            <mapping option="SECURITY_PROTOCOL" property="kafka.property.security.protocol" />
            <mapping option="SASL_MECHANISM" property="kafka.property.sasl.mechanism" />
            <mapping option="TRUSTSTORE_LOCATION" property="kafka.property.ssl.truststore.location" />
            <mapping option="TRUSTSTORE_PASSWORD" property="kafka.property.ssl.truststore.password" />
            <mapping option="KEYSTORE_LOCATION" property="kafka.property.ssl.keystore.location" />
            <mapping option="KEYSTORE_PASSWORD" property="kafka.property.ssl.keystore.password" />
            <mapping option="KERBEROS_SERVICE_NAME" property="kafka.property.sasl.kerberos.service.name" />

            <!-- Kafka JAAS properties -->
            <mapping option="KERBEROS_PRINCIPAL" property="kafka.jaas.property.sasl.kerberos.principal" />
            <mapping option="KERBEROS_KEYTAB" property="kafka.jaas.property.sasl.kerberos.keytab" />
            <mapping option="KERBEROS_USE_KEYTAB" property="kafka.jaas.property.sasl.kerberos.useKeyTab" />
            <mapping option="KERBEROS_STORE_KEY" property="kafka.jaas.property.sasl.kerberos.storeKey" />
            <mapping option="KERBEROS_USE_TICKET_CACHE" property="kafka.jaas.property.sasl.kerberos.useTicketCache" />
            <mapping option="KERBEROS_DEBUG" property="kafka.jaas.property.sasl.kerberos.debug" />
            <mapping option="SASL_USER" property="kafka.jaas.property.sasl.user" />
            <mapping option="SASL_USER_PASSWORD" property="kafka.jaas.property.sasl.user.password" />
            <mapping option="JAAS_CONFIG_FILE" property="kafka.jaas.property.config.file" />

            <!-- General properties -->
            <mapping option="BATCH_SIZE" property="kafka.batch.size"/>
            <mapping option="KAFKA_ADMIN_CONNECTION_TIMEOUT" property="kafka.admin.close.connection.timeout"/>
            <mapping option="TOPIC_AUTO_CREATE_FLAG" property="kafka.topic.auto.create"/>
            <mapping option="AVRO_DEFAULT_DECIMAL_PRECISION" property="avro.decimal.default.precision" />
            <mapping option="AVRO_DEFAULT_DECIMAL_SCALE" property="avro.decimal.default.scale" />
        </optionMappings>
    </profile>
</profiles>
```

ПРИМЕЧАНИЕ

Директория, в которой расположены конфигурационные файлы PXF, определяется переменной окружения `PXF_BASE`. Значение переменной по умолчанию — */var/lib/pxf*. В общем случае путь может быть иным.

3. Подключитесь к хосту координатора ADB под пользователем `gpadmin`, который создается по умолчанию. Все команды, приведенные на шагах ниже, запускаются исключительно на **хосте координатора**:

```
$ sudo su - gpadmin
```

4. Для синхронизации конфигурационных файлов PXF на всех хостах кластера ADB выполните команду:

```
$ pxf cluster sync
```

Результат:

```
Syncing PXF configuration files from coordinator host to standby coordinator host and 2 segment hosts...
PXF configs synced successfully on 3 out of 3 hosts
```

5. Перезапустите сервис PXF:

```
$ pxf cluster restart
```

Результат:

```
Restarting PXF on coordinator host, standby coordinator host, and 2 segment hosts...
PXF restarted successfully on 4 out of 4 hosts
```

Настройка ADB to Kafka Connector

Елена Кашникова

Содержание

- Создание внешней таблицы
- Использование опции SERVER
- Использование кастомных опций
- Кастомная настройка используемого механизма аутентификации SASL
- Сопоставление типов данных Greenplum и AVRO

Назад к содержанию

Создание внешней таблицы

Для создания внешней таблицы при запуске ранее, необходимо использовать новый тип таблицы `CREATE WRITABLE EXTERNAL TABLE`. Аналогичный синтаксис который:

```
CREATE WRITABLE EXTERNAL TABLE kafka_number_1
( i int4,t text,b bytea,j jsonb,q int8 )
LOCATION
( 'pxf://?profile=kafka&SERVER=default'
)
FORMAT 'CUSTOM'(formatter='avroprod_export');
INSERT INTEGRATION 'kafka_number_1' VALUES (1,'ОДИН',E'\\x313a4f4e45',NULL);
```

- `FORMAT 'CUSTOM'` — новый способ записи в Kafka.
- `FORMAT 'AVRO'` — старый способ.
- `FORMAT 'TEXT'` — способ записи в Kafka.
- `FORMAT 'JSON'` — способ записи в Kafka.
- `formatter='avroprod_export'` — новый параметр, указывающий на внутренний профиль по выгрузке данных, который использует новый тип таблицы `CREATE WRITABLE EXTERNAL TABLE`.
- `FORMAT 'CUSTOM'` — новая опция для выгрузки данных в Kafka.
- `INSERT INTEGRATION` — новая команда для выгрузки данных в Kafka.

Для выгрузки данных из таблицы в Kafka через `INSERT` необходимо сначала создать `внешнюю таблицу` для записи данных, а затем уже выгружать данные из `обычной таблицы` в `внешнюю таблицу`

Примеры настройки ADB to Kafka Connector

Содержание

Протокол PLAINTEXT (по умолчанию)
Без использования сертификатов
С использованием механизма PLAIN
Протокол SASL_PLAINTEXT
С использованием механизма PLAIN
Протокол SASL_SSL (с использованием механизмов SCRAM-SHA-512 и GSSAPI (Kerberos))
С использованием механизма SCRAM-SHA-512
С использованием механизма GSSAPI (Kerberos)

Кластер Kafka может использовать несколько слушателей (listeners), каждый из которых может использовать свой собственный протокол безопасности. Для каждого слушателя брокер Kafka использует переменные вида `KAFKA_LISTENER_SECURITY_PROTOCOL_MAP_<имя_слушателя>` для определения протокола, который должен использовать слушатель с заданным именем. Например, переменная `KAFKA_LISTENER_SECURITY_PROTOCOL_MAP_EXTERNAL` определяет, какой протокол должен использовать слушатель с именем `EXTERNAL`.

Для настройки слушателя, использующего протокол PLAINTEXT, необходимо задать переменные вида `KAFKA_LISTENER_SASL_JAAS_CONFIG_<имя_слушателя>` и `KAFKA_LISTENER_PASSWORD_<имя_слушателя>`, которые будут использоваться для аутентификации клиентов, подключающихся к брокеру Kafka.

Для настройки слушателя, использующего протокол SASL_PLAINTEXT, необходимо задать переменные вида `KAFKA_LISTENER_SASL_JAAS_CONFIG_<имя_слушателя>` и `KAFKA_LISTENER_PASSWORD_<имя_слушателя>`, которые будут использоваться для аутентификации клиентов, подключающихся к брокеру Kafka.

Протокол PLAINTEXT (по умолчанию)

Без использования сертификатов

Для использования протокола PLAINTEXT без использования сертификатов необходимо выполнить следующие действия:

1. Подготовить файл docker-compose.yaml, содержащий следующие переменные:

```
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP=PLAINTEXT:PLAINTEXT
KAFKA_LISTENER_SASL_JAAS_CONFIG_PLAINTEXT=org.apache.kafka.common.security.plain.PlainLoginModule required username="admin" password="admin";
KAFKA_LISTENER_PASSWORD_PLAINTEXT=admin
```

2. Подготовить файл конфигурации коннектора (например, kafka-connector.properties), содержащий следующие переменные:

```
security.protocol=PLAINTEXT
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required username="admin" password="admin";
```

3. Запустить сервис docker-compose:

```
docker-compose -f docker-compose.yaml -p kafka up -d
```

4. Добавить в топик данные для аутентификации:

```
/opt/kafka/bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'SCRAM-SHA-512=[iterations=8192,password=admin]' --entity-type users --entity-name admin
```

5. Запустить коннектор:

```
/opt/kafka/bin/connect-standalone.sh /opt/kafka/config/connect-standalone.properties /opt/kafka/config/kafka-connector.properties
```

6. Проверить статус коннектора:

```
curl -s -X GET -u admin:admin http://127.0.0.1:9095/connectors?expand=status
```

Ответ сервиса:

```
{"kafka-connector":{"status":{"name":"kafka-connector","connector":{"state":"RUNNING","worker_id":"127.0.0.1:9095"},"tasks":[{"id":0,"state":"RUNNING","worker_id":"127.0.0.1:9095"}],"type":"source"}}}
```

С использованием опции SERVER

Для использования протокола PLAINTEXT с использованием опции `KAFKA_LISTENER_SASL_JAAS_CONFIG_<имя_слушателя>` необходимо выполнить следующие действия:

1. Подготовить файл docker-compose.yaml, содержащий следующие переменные:

```
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP=EXTERNAL:PLAINTEXT
KAFKA_LISTENER_SASL_JAAS_CONFIG_EXTERNAL=com.sun.security.auth.module.LdapLoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
KAFKA_LISTENER_PASSWORD_EXTERNAL=admin
```

2. Подготовить файл конфигурации коннектора (например, kafka-connector.properties), содержащий следующие переменные:

```
security.protocol=PLAINTEXT
sasl.mechanism=PLAIN
sasl.jaas.config=com.sun.security.auth.module.LdapLoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
```

3. Запустить сервис docker-compose:

```
docker-compose -f docker-compose.yaml -p kafka up -d
```

4. Добавить в топик данные для аутентификации:

```
/opt/kafka/bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'SCRAM-SHA-512=[iterations=8192,password=admin]' --entity-type users --entity-name admin
```

5. Запустить коннектор:

```
/opt/kafka/bin/connect-standalone.sh /opt/kafka/config/connect-standalone.properties /opt/kafka/config/kafka-connector.properties
```

6. Проверить статус коннектора:

```
curl -s -X GET -u admin:admin http://127.0.0.1:9095/connectors?expand=status
```

Ответ сервиса:

```
{"kafka-connector":{"status":{"name":"kafka-connector","connector":{"state":"RUNNING","worker_id":"127.0.0.1:9095"},"tasks":[{"id":0,"state":"RUNNING","worker_id":"127.0.0.1:9095"}],"type":"source"}}}
```

Протокол SASL_PLAINTEXT

С использованием механизма PLAIN

1. Подготовить файл docker-compose.yaml, содержащий следующие переменные:

```
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP=EXTERNAL:SASL_PLAINTEXT
KAFKA_LISTENER_SASL_JAAS_CONFIG_EXTERNAL=com.sun.security.auth.module.LdapLoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
KAFKA_LISTENER_PASSWORD_EXTERNAL=admin
```

2. Подготовить файл конфигурации коннектора (например, kafka-connector.properties), содержащий следующие переменные:

```
security.protocol=SASL_PLAINTEXT
sasl.mechanism=PLAIN
sasl.jaas.config=com.sun.security.auth.module.LdapLoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
```

3. Запустить сервис docker-compose:

```
docker-compose -f docker-compose.yaml -p kafka up -d
```

4. Добавить в топик данные для аутентификации:

```
/opt/kafka/bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'SCRAM-SHA-512=[iterations=8192,password=admin]' --entity-type users --entity-name admin
```

5. Запустить коннектор:

```
/opt/kafka/bin/connect-standalone.sh /opt/kafka/config/connect-standalone.properties /opt/kafka/config/kafka-connector.properties
```

6. Проверить статус коннектора:

```
curl -s -X GET -u admin:admin http://127.0.0.1:9095/connectors?expand=status
```

Ответ сервиса:

```
{"kafka-connector":{"status":{"name":"kafka-connector","connector":{"state":"RUNNING","worker_id":"127.0.0.1:9095"},"tasks":[{"id":0,"state":"RUNNING","worker_id":"127.0.0.1:9095"}],"type":"source"}}}
```

Протокол SASL_SSL (с использованием механизмов SCRAM-SHA-512 и GSSAPI (Kerberos))

С использованием механизма SCRAM-SHA-512

1. Подготовить файл docker-compose.yaml, содержащий следующие переменные:

```
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP=EXTERNAL:SASL_SSL
KAFKA_LISTENER_SASL_JAAS_CONFIG_EXTERNAL=com.sun.security.auth.module.LdapLoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
KAFKA_LISTENER_PASSWORD_EXTERNAL=admin
KAFKA_SSL_KEYSTORE_LOCATION=/opt/kafka/ssl/kafka.keystore.jks
KAFKA_SSL_KEYSTORE_PASSWORD=kafkapass
KAFKA_SSL_KEY_PASSWORD=kafkapass
KAFKA_SSL_TRUSTSTORE_LOCATION=/opt/kafka/ssl/kafka.truststore.jks
KAFKA_SSL_TRUSTSTORE_PASSWORD=kafkapass
KAFKA_SSL_CLIENT_AUTH=required
```

2. Подготовить файл конфигурации коннектора (например, kafka-connector.properties), содержащий следующие переменные:

```
security.protocol=SASL_SSL
sasl.mechanism=SCRAM-SHA-512
sasl.jaas.config=com.sun.security.auth.module.LdapLoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
ssl.truststore.location=/opt/kafka/ssl/kafka.truststore.jks
ssl.truststore.password=kafkapass
ssl.keystore.location=/opt/kafka/ssl/kafka.keystore.jks
ssl.keystore.password=kafkapass
ssl.key.password=kafkapass
```

3. Запустить сервис docker-compose:

```
docker-compose -f docker-compose.yaml -p kafka up -d
```

4. Добавить в топик данные для аутентификации:

```
/opt/kafka/bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'SCRAM-SHA-512=[iterations=8192,password=admin]' --entity-type users --entity-name admin
```

5. Запустить коннектор:

```
/opt/kafka/bin/connect-standalone.sh /opt/kafka/config/connect-standalone.properties /opt/kafka/config/kafka-connector.properties
```

6. Проверить статус коннектора:

```
curl -s -X GET -u admin:admin http://127.0.0.1:9095/connectors?expand=status
```

Ответ сервиса:

```
{"kafka-connector":{"status":{"name":"kafka-connector","connector":{"state":"RUNNING","worker_id":"127.0.0.1:9095"},"tasks":[{"id":0,"state":"RUNNING","worker_id":"127.0.0.1:9095"}],"type":"source"}}}
```

С использованием механизма GSSAPI (Kerberos)

1. Подготовить файл docker-compose.yaml, содержащий следующие переменные:

```
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP=EXTERNAL:SASL_SSL
KAFKA_LISTENER_SASL_JAAS_CONFIG_EXTERNAL=com.sun.security.auth.module.Krb5LoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
KAFKA_LISTENER_PASSWORD_EXTERNAL=admin
KAFKA_SSL_KEYSTORE_LOCATION=/opt/kafka/ssl/kafka.keystore.jks
KAFKA_SSL_KEYSTORE_PASSWORD=kafkapass
KAFKA_SSL_KEY_PASSWORD=kafkapass
KAFKA_SSL_TRUSTSTORE_LOCATION=/opt/kafka/ssl/kafka.truststore.jks
KAFKA_SSL_TRUSTSTORE_PASSWORD=kafkapass
KAFKA_SSL_CLIENT_AUTH=required
KAFKA_SASL_KERBEROS_SERVICE_NAME=kafka
KAFKA_SASL_KERBEROS_KINIT_CMD=/usr/bin/kinit -k -t /opt/kafka/keytab/kafkaadmin.keytab kafkaadmin
KAFKA_SASL_KERBEROS_TICKET_RENEW_WINDOW_FACTOR=0.80
KAFKA_SASL_KERBEROS_TICKET_RENEW_JITTER=0.05
KAFKA_SASL_KERBEROS_MIN_TIME_BEFORE_RELOGIN=60000
KAFKA_SASL_KERBEROS_PRINCIPAL_TO_LOCAL_RULES=RULE:[1:$1@$0](.*@.*)s/@.*//,RULE:[2:$1/$2@$0](.*/.*@.*)s/@.*//,RULE:[2:$1/$2@$0](.*/.*@.*)s/.*/$1/,DEFAULT
```

2. Подготовить файл конфигурации коннектора (например, kafka-connector.properties), содержащий следующие переменные:

```
security.protocol=SASL_SSL
sasl.mechanism=GSSAPI
sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule sufficient debug="true" serviceName="kafka" forceBindingLogin="true" forcePasswordUpdate="false" useFirstPass="true" storePass="true" useTokenCredential="false" clearPass="true" debug="true" serviceUrl="ldap://127.0.0.1:389/dc=netcracker,dc=com" bindDN="cn=admin,dc=netcracker,dc=com" bindCredential="admin" authzIdentity="{cn},ou=users,dc=netcracker,dc=com";
ssl.truststore.location=/opt/kafka/ssl/kafka.truststore.jks
ssl.truststore.password=kafkapass
ssl.keystore.location=/opt/kafka/ssl/kafka.keystore.jks
ssl.keystore.password=kafkapass
ssl.key.password=kafkapass
sasl.kerberos.service.name=kafka
sasl.kerberos.kinit.cmd=/usr/bin/kinit -k -t /opt/kafka/keytab/kafkaadmin.keytab kafkaadmin
sasl.kerberos.ticket.renew.window.factor=0.80
sasl.kerberos.ticket.renew.jitter=0.05
sasl.kerberos.min.time.before.relogin=60000
sasl.kerberos.principal.to.local.rules=RULE:[1:$1@$0](.*@.*)s/@.*//,RULE:[2:$1/$2@$0](.*/.*@.*)s/@.*//,RULE:[2:$1/$2@$0](.*/.*@.*)s/.*/$1/,DEFAULT
```

3. Запустить сервис docker-compose:

```
docker-compose -f docker-compose.yaml -p kafka up -d
```

4. Добавить в топик данные для аутентификации:

```
/opt/kafka/bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'SCRAM-SHA-512=[iterations=8192,password=admin]' --entity-type users --entity-name admin
```

5. Запустить коннектор:

```
/opt/kafka/bin/connect-standalone.sh /opt/kafka/config/connect-standalone.properties /opt/kafka/config/kafka-connector.properties
```

6. Проверить статус коннектора:

```
curl -s -X GET -u admin:admin http://127.0.0.1:9095/connectors?expand=status
```

Ответ сервиса:

```
{"kafka-connector":{"status":{"name":"kafka-connector","connector":{"state":"RUNNING","worker_id":"127.0.0.1:9095"},"tasks":[{"id":0,"state":"RUNNING","worker_id":"127.0.0.1:9095"}],"type":"source"}}}
```

Ошибки аутентификации

В случае ошибки аутентификации клиентам и брокерам при обмене данными выдается ошибка SASL authentication failed. Причины аутентификации могут быть следующими:

- Неверный пароль
- Неверный SASL механизм
- Неверный протокол безопасности

Пример ошибки при попытке подключиться к брокеру при использовании протокола SASL_SSL:

```
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] Connection to node -1 failed authentication due to: SSL handshake failed (org.apache.kafka.clients.NetworkClient)
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] [Producer clientId=producer-1] Connection to node -1 (/127.0.0.1:9095) failed authentication due to: SSL handshake failed
[2022-12-12 12:12:12,123] WARN [Kafka client thread | producer-1] [Producer clientId=producer-1] Bootstrap broker 127.0.0.1:9095 (id: -1 rack: null) disconnected
```

Пример ошибки при попытке подключиться к брокеру при использовании протокола SASL_PLAINTEXT:

```
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] Connection to node -1 failed authentication due to: Unexpected handshake request with client mechanism PLAIN, enabled mechanisms are GSSAPI,SCRAM-SHA-512 (org.apache.kafka.clients.NetworkClient)
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] [Producer clientId=producer-1] Connection to node -1 (/127.0.0.1:9095) failed authentication due to: Unexpected handshake request with client mechanism PLAIN, enabled mechanisms are GSSAPI,SCRAM-SHA-512
[2022-12-12 12:12:12,123] WARN [Kafka client thread | producer-1] [Producer clientId=producer-1] Bootstrap broker 127.0.0.1:9095 (id: -1 rack: null) disconnected
```

Ошибки аутентификации

В случае ошибки аутентификации клиентам выдается ошибка SASL authentication failed. Причины аутентификации могут быть следующими:

- Неверный пароль
- Неверный SASL механизм
- Неверный протокол безопасности

Пример ошибки при попытке подключиться к брокеру при использовании протокола SASL_SSL:

```
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] Connection to node -1 failed authentication due to: SSL handshake failed (org.apache.kafka.clients.NetworkClient)
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] [Producer clientId=producer-1] Connection to node -1 (/127.0.0.1:9095) failed authentication due to: SSL handshake failed
[2022-12-12 12:12:12,123] WARN [Kafka client thread | producer-1] [Producer clientId=producer-1] Bootstrap broker 127.0.0.1:9095 (id: -1 rack: null) disconnected
```

Пример ошибки при попытке подключиться к брокеру при использовании протокола SASL_PLAINTEXT:

```
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] Connection to node -1 failed authentication due to: Unexpected handshake request with client mechanism PLAIN, enabled mechanisms are GSSAPI,SCRAM-SHA-512 (org.apache.kafka.clients.NetworkClient)
[2022-12-12 12:12:12,123] ERROR [Kafka client thread | producer-1] [Producer clientId=producer-1] Connection to node -1 (/127.0.0.1:9095) failed authentication due to: Unexpected handshake request with client mechanism PLAIN, enabled mechanisms are GSSAPI,SCRAM-SHA-512
[2022-12-12 12:12:12,123] WARN [Kafka client thread | producer-1] [Producer clientId=producer-1] Bootstrap broker 127.0.0.1:9095 (id: -1 rack: null) disconnected
```

Установка Kafka to ADB Connector

Дарья Барышева

Назад к содержанию

Для установки Kafka to ADB Connector выполните следующие шаги:

1. Убедитесь, что сервис Kafka to ADB установлен в кластере ADB. Информацию о добавлении сервисов в кластер можно получить в статье Добавление сервисов.
2. Подключитесь к базе данных через `psql` (или иную клиентскую программу):

```
$ psql adb
```

3. Убедитесь, что расширение `kadb_fdw` успешно добавлено. Для этого можно использовать команду `psql` `\dx`:

```
\dx
```

Результат содержит расширение `kadb_fdw`:

```
                                    List of installed extensions
       Name        | Version |   Schema   |                              Description
-------------------+---------+------------+-----------------------------------------------------------------------
 adb_fdw           | 1.0     | public     | foreign-data wrapper for remote ADB servers
 arenadata_toolkit | 1.0     | public     | extension is used for manipulation of objects created by adb-bundle
 dblink            | 1.2     | public     | connect to other PostgreSQL databases from within a database
 gp_exttable_fdw   | 1.0     | pg_catalog | External Table Foreign Data Wrapper for Greengage
 gp_toolkit        | 1.8     | gp_toolkit | various GPDB administrative views/functions
 kadb_fdw          | 0.16    | kadb       | Kafka-ADB foreign data wrapper
 plpgsql           | 1.0     | pg_catalog | PL/pgSQL procedural language
 pxf               | 2.1     | public     | Extension which allows to access unmanaged data
 pxf_fdw           | 1.0     | public     | PXF Foreign Data Wrapper for Greengage
 tkh_fdw           | 1.0     | public     | ClickHouse Foreign Data Wrapper for Greenplum
(10 rows)
```

4. Проверьте наличие новой обертки сторонних данных (foreign data wrapper) `kadb_fdw`, обратившись к таблице системного каталога `pg_catalog.pg_foreign_data_wrapper`:

```
SELECT * FROM pg_catalog.pg_foreign_data_wrapper;
```

Результат:

```
  oid  |     fdwname     | fdwowner | fdwhandler | fdwvalidator | fdwacl |                                        fdwoptions
-------+-----------------+----------+------------+--------------+--------+-----------------------------------------------------
-------------------------------------
 13702 | gp_exttable_fdw |       10 |      13700 |        13701 |        |
 17124 | dblink_fdw      |       10 |          0 |        17123 |        |
 17175 | adb_fdw         |       10 |      17173 |        17174 |        |
 17187 | jdbc_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=jdbc,"mpp_execute=all segments"}
 17188 | hdfs_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=hdfs,"mpp_execute=all segments"}
 17189 | hive_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=hive,"mpp_execute=all segments"}
 17190 | hbase_pxf_fdw   |       10 |      17185 |        17186 |        | {protocol=hbase,"mpp_execute=all segments"}
 17191 | s3_pxf_fdw      |       10 |      17185 |        17186 |        | {protocol=s3,"mpp_execute=all segments"}
 17192 | gs_pxf_fdw      |       10 |      17185 |        17186 |        | {protocol=gs,"mpp_execute=all segments"}
 17193 | adl_pxf_fdw     |       10 |      17185 |        17186 |        | {protocol=adl,"mpp_execute=all segments"}
 17194 | wasbs_pxf_fdw   |       10 |      17185 |        17186 |        | {protocol=wasbs,"mpp_execute=all segments"}
 17195 | file_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=file,"mpp_execute=all segments"}
 17201 | tkh_fdw         |       10 |      17200 |        17199 |        | {protocol=tkh_clickhouse,"mpp_execute=all
segments",format=text}
 17210 | kadb_fdw        |       10 |      17209 |        17211 |        | {"mpp_execute=all
segments",k_initial_offset=0,k_automatic_offsets=true,k_latency_ms=2000}
(14 rows)
```

5. Убедитесь в создании служебной схемы kadb. Для этого можно использовать команду `psql` `\dnS+`:

```
\dnS+
```

Результат:

```
        Name        |  Owner  | Access privileges  |                        Description
--------------------+---------+--------------------+-----------------------------------------------------------------
 arenadata_toolkit  | gpadmin | gpadmin=UC/gpadmin+|
                    |         | =U/gpadmin         |
 diskquota          | gpadmin |                    |
 gp_toolkit         | gpadmin | gpadmin=UC/gpadmin+|
                    |         | =U/gpadmin         |
 information_schema | gpadmin | gpadmin=UC/gpadmin+|
                    |         | =U/gpadmin         |
 kadb               | gpadmin |                    |
 pg_aoseg           | gpadmin |                    | Reserved schema for Append Only segment list and eof tables
 pg_bitmapindex     | gpadmin |                    | Reserved schema for internal relations of bitmap indexes
 pg_catalog         | gpadmin | gpadmin=UC/gpadmin+| system catalog schema
                    |         | =U/gpadmin         |
 pg_toast           | gpadmin |                    | reserved schema for TOAST tables
 public             | gpadmin | gpadmin=UC/gpadmin+| standard public schema
                    |         | =UC/gpadmin        |
(10 rows)
```

Настройка Kafka for ADB Connector

Содержание

- Объекты для работы с коннектором
- Схема
  - Таблица topic_offset
  - Таблица topic_partitions
- Схема kafk
  - Таблица kafk_topic
  - Таблица kafk_partitions
- Поддерживаемые форматы
  - JSON
  - CSV
  - Text
- Особенности использования коннектора
  - Ограничения
  - Поддержка сжатия данных
  - Поддержка шифрования данных
  - Поддержка аутентификации
  - Поддержка сертификатов SSL/SCT
- Примеры
  - Чтение топика kafk_topic
  - Вставка в топик kafk_topic
  - Чтение топика kafk_topic с фильтрацией по смещениям
  - Чтение топика kafk_topic с фильтрацией по ключу
  - Чтение топика kafk_topic с фильтрацией по времени
  - Чтение топика kafk_topic с фильтрацией по разделам
  - Чтение топика kafk_topic с фильтрацией по разделам и смещениям
  - Чтение топика kafk_topic с фильтрацией по разделам и времени
  - Чтение топика kafk_topic с фильтрацией по разделам и ключу
  - Чтение топика kafk_topic с фильтрацией по разделам, смещениям и ключу
  - Чтение топика kafk_topic с фильтрацией по разделам, смещениям и времени
  - Чтение топика kafk_topic с фильтрацией по разделам, ключу и времени
  - Чтение топика kafk_topic с фильтрацией по разделам, смещениям

Примеры использования Kafka to ADB Connector

© Дарья Кадыкова

Содержание

Назад к содержанию

Протокол PLAINTEXT (по умолчанию)
Протокол SASL_PLAINTEXT
└ С использованием механизма PLAIN
└ С использованием механизма GSSAPI (Kerberos)
Протокол SASL_SSL с использованием механизма GSSAPI (Kerberos)

В статье показаны примеры чтения сообщений коннектором Kafka to ADB, записанных в формате AVRO-OCF в топик Kafka (в составе KC3) при помощи коннектора ADB to Kafka. Для использования обоих коннекторов одновременно необходимы: (для примеров – показана возможность чтения сообщений в формате AVRO OCF (именно в этом формате сообщения записываются коннектором ADB to Kafka). При выполнении примеров обобщены следующие условия:

- Кластер ADB установлен согласно руководству Online установка на следующих хостах: **bds-adb1.ru-central1.internal**, **bds-adb2.ru-central1.internal**, **bds-adb3.ru-central1.internal**
- Кластер ADB установлен согласно руководству Online установка.
- В кластере ADB добавлен и установлен сервис **Kafka to ADB**.
- База данных с именем **bdb** существует в ADB.
- Все ноды кластера ADB имеют доступ к порту **9092** на всех нодах кластера ADB, через которые планируется производить загрузку данных.
- На стороне ADB установлено ядро to Kafka Connector

Протокол PLAINTEXT (по умолчанию)

1. Выполните шаги 1–5 из раздела **Без использования опции SERVER** статьи Примеры использования ADB to Kafka Connector (если они не были выполнены ранее). В результате тестовый топик **topic_adb_to_kafka** будет содержать 5 сообщений.
2. Подключитесь к базе данных **bdb** кластера ADB под пользователем **opadmin** (например, через **psql**). Создайте последовательные следующие объекты:

   - Сервер **kafka_server**:

```
DROP SERVER IF EXISTS kafka_server CASCADE;
CREATE SERVER kafka_server
FOREIGN DATA WRAPPER kadb_fdw
OPTIONS (
    k_brokers 'bds-adb1.ru-central1.internal:9092,bds-adb2.ru-central1.internal:9092,bds-adb3.ru-central1.internal:9092'
);
```

   - Сторонняя таблица **kafka_table** (обратите внимание, что структура таблицы должна соответствовать используемой AVRO схеме):

```
DROP FOREIGN TABLE IF EXISTS kafka_table;
CREATE FOREIGN TABLE kafka_table(k INT, b TEXT)
SERVER kafka_server
OPTIONS (
    format 'avro',
    k_topic 'topic_adb_to_kafka',
    k_consumer_group 'group',
    k_seq_batch '100',
    k_timeout_ms '1000'
);
```

3. Добавьте данные в таблицу ext_adb_to_kafka для их отправки в топик ADB: **topic_adb_to_kafka**

```
INSERT INTO ext_adb_to_kafka VALUES
(6,'test6 ),
(7,'test7'),
(8,'test8 ),
(9,'test9'),
(10,'test10');
```

4. Прочитайте данные из сторонней таблицы **kafka_table**:

```
SELECT * FROM kafka_table;
```

   Результат содержит служебные уведомления, подробно описанные в статье Настройка Kafka to ADB Connector (см. опцию **k_initial_offset** и **k_automatic_offset** в таблице **Опции сервера**), а также список сообщений, прочитанных из Kafka:

```
NOTICE:  kafka-ADB: Offset for partition 0 is not known, and is set to default value 0  (seg0 slice1 10.92.38.148:10000 pid=22274)
# |  b
--+----------
1 | test1
2 | test2
4 | test4
3 | test3
5 | test5
7 | test7
6 | test6
8 | test8
9 | test9
10 | test10
(10 rows)
```

5. Убедитесь в появлении записей в таблице kadb.offsets:

```
SELECT * FROM kadb.offsets;
```

   Результат:

```
rtopic | part | off
-------+------+-----
ASKDA  |  0 0 |  10
(1 row)
```

   где **ASKDA** – это ОID созданной таблицы **kafka_table**, который может быть получен следующим образом

```
SELECT 'public.kafka_table'::regclass::oid;
```

```
  oid
-------
 45426
(1 row)
```

Протокол SASL_PLAINTEXT

С использованием механизма PLAIN

1. Выполните все шаги из подраздела **С использованием механизма PLAIN** статьи Примеры использования ADB to Kafka Connector (если они не были выполнены ранее). В результате тестовый топик **topic_adb_to_kafka_sasl** будет содержать 1 сообщение.
2. Подключитесь к базе данных **bdb** кластера ADB под пользователем **opadmin** (например, через **psql**). Создайте последовательные следующие объекты:

   - Сервер **kafka_server**:

```
DROP SERVER IF EXISTS kafka_server CASCADE;
CREATE SERVER kafka_server
FOREIGN DATA WRAPPER kadb_fdw
OPTIONS (
    k_brokers 'bds-adb1.ru-central1.internal:9092,bds-adb2.ru-central1.internal:9092,bds-adb3.ru-central1.internal:9092'
);
```

   - Сторонняя таблица **kafka_table_sasl** (обратите внимание, что по сравнению с предыдущим примером в определении таблицы присутствуют опции authmeth, специфичные для протокола **SASL_PLAINTEXT** и механизма **PLAIN**):

```
DROP FOREIGN TABLE IF EXISTS kafka_table_sasl;
CREATE FOREIGN TABLE kafka_table_sasl(k INT, b TEXT)
SERVER kafka_server
OPTIONS (
    format 'avro',
    k_topic 'topic_adb_to_kafka_sasl',
    k_consumer_group 'group',
    k_seq_batch '100',
    k_timeout_ms '1000',
    k_security_protocol 'sasl_plaintext',
    k_sasl_mechanism 'PLAIN',
    k_sasl_username 'adb-to-kafka',
    k_sasl_password '123'
);
```

3. Прочитайте данные из сторонней таблицы **kafka_table_sasl**:

```
SELECT * FROM kafka_table_sasl;
```

   Результат содержит служебные уведомления, подробно описанные в статье Настройка Kafka to ADB Connector (см. опцию **k_initial_offset** и **k_automatic_offset** в таблице **Опции сервера**), а также список сообщений, прочитанных из Kafka:

```
NOTICE:  kafka-ADB: Offset for partition 0 is not known, and is set to default value 0  (seg0 slice1 10.92.38.148:10000 pid=8036)
# |  b
--+----------
1 | test1
(1 row)
```

Ошибки аутентификации

В тех случаях, когда логин или пароль пользователя не указаны при объявлении сторонней таблицы ADB либо указаны неверно, при попытке чтения данных из таблицы возвращается ошибка следующего вида:

```
ERROR:  kafka-ADB: [kafka] Failed to retrieve metadata for topic 'topic_adb_to_kafka_sasl': Local: Broker transport failure [-195] (1)
```

С использованием механизма GSSAPI (Kerberos)

1. Выполните все шаги из подраздела **С использованием механизма GSSAPI (Kerberos)** статьи Примеры использования ADB to Kafka Connector (если они не были выполнены ранее). В результате тестовый топик **topic_adb_to_kafka_kerberos** будет содержать 1 сообщение.
2. Подключитесь к базе данных **bdb** кластера ADB под пользователем **opadmin** (например, через **psql**). Создайте последовательные следующие объекты:

   - Сервер **kafka_server**:

```
DROP SERVER IF EXISTS kafka_server CASCADE;
CREATE SERVER kafka_server
FOREIGN DATA WRAPPER kadb_fdw
OPTIONS (
    k_brokers 'bds-adb1.ru-central1.internal:9092,bds-adb2.ru-central1.internal:9092,bds-adb3.ru-central1.internal:9092'
);
```

   - Сторонняя таблица **kafka_table_kerberos** (обратите внимание, что по сравнению с предыдущим примером в определении таблицы присутствуют опции authmeth, специфичные для протокола **SASL_PLAINTEXT** и механизма **GSSAPI**):

```
DROP FOREIGN TABLE IF EXISTS kafka_table_kerberos;
CREATE FOREIGN TABLE kafka_table_kerberos(k INT, b TEXT)
SERVER kafka_server
OPTIONS (
    format 'avro',
    k_topic 'topic_adb_to_kafka_kerberos',
    k_consumer_group 'group',
    k_seq_batch '100',
    k_timeout_ms '1000',
    k_security_protocol 'sasl_plaintext',
    k_sasl_mechanism 'GSSAPI',
    k_sasl_kerberos_keytab '/var/lib/pxf/conf/kerberos/adb-to-kafka.service.keytab',
    k_sasl_kerberos_principal 'adb-to-kafka',
    k_sasl_kerberos_service_name 'kafka'
);
```

3. Прочитайте данные из сторонней таблицы **kafka_table_kerberos**:

```
SELECT * FROM kafka_table_kerberos;
```

   Результат содержит служебные уведомления, подробно описанные в статье Настройка Kafka to ADB Connector (см. опцию **k_initial_offset** и **k_automatic_offset** в таблице **Опции сервера**), а также список сообщений, прочитанных из Kafka:

```
NOTICE:  kafka-ADB: Offset for partition 0 is not known, and is set to default value 0  (seg0 slice1 10.92.38.148:10000 pid=2484)
# |  b
--+----------
1 | kerberos
(1 row)
```

Ошибки аутентификации

В тех случаях, когда keytab, принадлежащий или имя сервиса заполнены неверно при объявлении сторонней таблицы ADB (в том этом в виде нет валидного токена для выбранного сервиса), при попытке чтения данных из таблицы возвращается ошибка следующего вида:

```
ERROR:  kafka-ADB: [kafka] Failed to retrieve metadata for topic 'topic_adb_to_kafka_kerberos': Local: Broker transport failure [-195] (1)
```

Протокол SASL_SSL с использованием механизма GSSAPI (Kerberos)

1. Выполните все шаги из подраздела **Протокол SASL_SSL с использованием механизма GSSAPI (Kerberos)** статьи Примеры использования ADB to Kafka Connector (если они не были выполнены ранее). В результате тестовый топик **topic_ssl** будет содержать 1 сообщение
2. Подключитесь к базе данных **bdb** кластера ADB под пользователем **opadmin** (например, через **psql**). Создайте последовательные следующие объекты:

   - Сервер **kafka_server_ssl** (обратите внимание, что по сравнению с предыдущими примерами в определении сервера необходимо обязательно указать FQDN хостов ADB):

```
DROP SERVER IF EXISTS kafka_server_ssl CASCADE;
CREATE SERVER kafka_server_ssl
FOREIGN DATA WRAPPER kadb_fdw
OPTIONS (
    k_brokers 'bds-adb1.ru-central1.internal:9092,bds-adb2.ru-central1.internal:9092,bds-adb3.ru-central1.internal:9092'
);
```

   - Сторонняя таблица **kafka_table_ssl** (обратите внимание, что по сравнению с предыдущими примерами в определении таблицы присутствуют опции authmeth, специфичные для протокола **SASL_SSL** и механизма **GSSAPI**):

```
DROP FOREIGN TABLE IF EXISTS kafka_table_ssl;
CREATE FOREIGN TABLE kafka_table_ssl(k INT, b TEXT)
SERVER kafka_server_ssl
OPTIONS (
    format 'avro',
    k_topic 'topic_ssl',
    k_consumer_group 'group',
    k_seq_batch '100',
    k_timeout_ms '1000',
    k_security_protocol 'sasl_ssl',
    k_sasl_mechanism 'GSSAPI',
    k_sasl_kerberos_keytab '/var/lib/pxf/conf/kerberos/adb-to-kafka.service.keytab',
    k_sasl_kerberos_principal 'adb-to-kafka',
    k_sasl_kerberos_service_name 'kafka',
    k_ssl_ca_location '/etc/ssl/certs/ca-bundle.crt',
    k_ssl_certificate_location '/etc/ssl/certs/host_cert.cert',
    k_ssl_key_password '1qaz2wx'
);
```

3. Прочитайте данные из сторонней таблицы **kafka_table_ssl**:

```
SELECT * FROM kafka_table_ssl;
```

   Результат содержит служебные уведомления, подробно описанные в статье Настройка Kafka to ADB Connector (см. опцию **k_initial_offset** и **k_automatic_offset** в таблице **Опции сервера**), а также список сообщений, прочитанных из Kafka:

```
NOTICE:  kafka-ADB: Offset for partition 0 is not known, and is set to default value 0  (seg0 slice1 10.92.38.148:10000 pid=28357)
# |    b
--+----------
1 | ssl-test-data
(1 row)
```

Ошибки аутентификации

В тех случаях, когда пути к сертификатам и ключам не указаны при объявлении сторонней таблицы ADB либо указаны неверно, при попытке чтения данных из таблицы возвращается ошибка следующего вида:

```
ERROR:  kafka-ADB: [kafka] Failed to create a Kafka consumer: ssl.key.location failed: error:0A000002:SSL routines::system lib [5]
```

При указании неверных keytab, принципала или имени сервиса сообщение об ошибке аналогично описанному выше для протокола SASL_PLAINTEXT с механизмом GSSAPI.

Обзор ADB to ADB Connector

Д. Дарья Бирюкова

Содержание

Назад к содержанию

- Функции
- Порядок выполнения запросов
  - SELECT
  - INSERT
- Поддержка push-down предикатов
- Возможные схемы работы
  - Автоматическое определение числа обработчиков
  - Использование числа обработчиков, равного числу локальных сегментов
  - Явное указание числа обработчиков

Функции

ADB to ADB Connector — это реализованный на основе обертки сторонних данных (foreign data wrapper) и параллельных курсоров (parallel retrieve cursor) коннектор, обеспечивающий распределенную передачу данных между двумя кластерами ADB. Основные возможности коннектора перечислены ниже:

- Параллельное получение данных сегментами локального кластера напрямую с сегментов удаленного кластера. Для ряда запросов (например, с использованием агрегатов) допускается установка связи вида "coordinator-coordinator".
- Транзакционная вставка данных с локального на удаленный кластер в режиме "coordinator-coordinator". ADB to ADB Connector обеспечивает поддержку транзакционности аналогично postgres_fdw в следующем виде:
  - Старт транзакции при открытии соединения.
  - Реализация обработки вложенных транзакций через SAVEPOINT.
  - Обработка отказа транзакций.
  - Закрытие параметризованных запросов.
- Возможность автоматического определения количества обработчиков (Query Executor, QE) — на Возможные схемы работы ниже. Если пользователь не хочет указывать число сегментов в удаленном кластере, он может определить опцию `use_remote_estimate = true` на уровне сервера или сторонней таблицы — для автоматической оценки этого числа на основе запроса (см. Настройка ADB to ADB Connector). При этом изменение числа сегментов в удаленном кластере не влияет на работу ADB to ADB Connector.

ВАЖНО

- ADB to ADB Connector доступен только в Enterprise-версии ADB (начиная с версии 6.25.1.51 в ADB 6 и версии 7.4 в ADB 7).
- ADB to ADB Connector не поддерживает запросы `UPDATE` и `DELETE`. При попытке их вызова будет возвращена ошибка `ERROR: adb_fdw: UPDATE and DELETE queries are forbidden`.
- Отправка запросов `INSERT` в целевой (remote) cluster не является основным режимом работы ADB to ADB Connector и не подходит для массовой вставки данных. Более эффективной стратегией является использование коннектора в режиме `SELECT` в удаленном кластере с последующим вызовом `INSERT` на его стороне. В этом случае будет задействована параллельная получение данных сегментами удаленного кластера напрямую с сегментов локального кластера, а при вставке удаленный кластер самостоятельно перераспределит данные по своим сегментам.

Порядок выполнения запросов

SELECT

Обработка запросов `SELECT` в ADB to ADB Connector выполняется в следующей последовательности:

1. Пользователь отправляет запрос `SELECT` на хост координатора локального кластера.
2. Хост координатора локального кластера передает запрос на создание параллельного курсора на хост координатора удаленного кластера.
3. Удаленный координатор создает endpoint.
4. В результате вызова функции `gp_get_endpoints` удаленный координатор возвращает локальному координатору служебную информацию по созданным endpoint: количество, IP-адреса и номера портов, а также дополнительные данные, необходимые для аутентификации.
5. На основании полученной информации о числе endpoint создается равное ему количество обработчиков (Query Executor, QE) на сегментах локального кластера, и запускаются служебные сессии — retrieve session.
6. Сегменты локального кластера начинают чтение данных непосредственно с сегментов удаленного кластера.
7. Сегменты локального кластера передают данные локальному координатору для передачи результата пользователю.

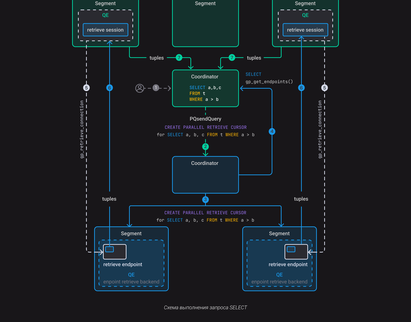


Схема выполнения запроса SELECT

INSERT

Обработка запросов `INSERT` в ADB to ADB Connector осуществляется в режиме "coordinator-coordinator", то есть сегменты локального и удаленного кластеров напрямую друг с другом не взаимодействуют. На текущий момент такая схема является единственным вариантом обеспечения транзакционной вставки данных.

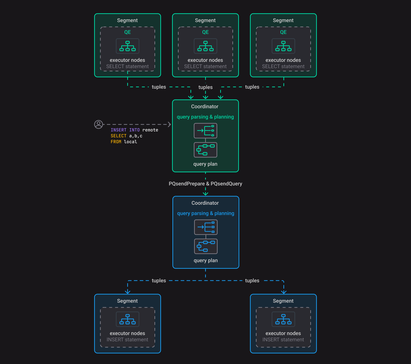


Схема выполнения запроса INSERT

Поддержка push-down предикатов

Ввиду ряда ограничений, присутствующих в Greenplum 6, в текущей версии ADB to ADB Connector поддержка push-down предикатов в запросах `SELECT` (с возможностью обработки на стороне удаленного кластера) реализована не в полной мере. Список доступных предикатов и особенности их обработки отражены в таблице ниже.

Предикат	Поддержка push-down	Примечание
WHERE	Да, с исключениями	На удаленный кластер передаются только предикаты, оперирующие встроенными типами данных, операторами и функциями. Также используемые в `WHERE` операторы и функции должны удовлетворять условию `IMMUTABLE`. Внутренние системные столбцы, за исключением `ctid`, не передаются на удаленный кластер
LIMIT	Нет	Условие применяется на стороне локального кластера
ORDER BY	Нет	Условие применяется на стороне локального кластера
Любая агрегатная функция	Нет	Удаленный кластер возвращает все строки, которые удовлетворяют условиям в `WHERE`, при их наличии. Далее происходит их локальная обработка в рамках агрегатной функции
Любой тип JOIN	Нет	Удаленный кластер возвращает все строки, которые удовлетворяют условиям в `WHERE`, при их наличии. Далее происходит их локальная обработка в рамках JOIN

Возможные схемы работы

ПРИМЕЧАНИЕ

- Дополнительную информацию об упоминаемых ниже опциях можно получить в статье Настройка ADB to ADB Connector.
- Примеры использования ADB to ADB Connector приведены в статье Примеры использования ADB to ADB Connector.

Автоматическое определение числа обработчиков

В этой схеме на уровне сервера или сторонней таблицы указывается опция `use_remote_estimate = true`. В результате необходимое количество обработчиков определяется автоматически — как равное числу endpoints параллельного курсора.

Пример определения сервера приведен ниже:

```
CREATE SERVER test_server
FOREIGN DATA WRAPPER adb_fdw
OPTIONS (
    host 'bds-mdw-ext.ru-central1.internal',
    port '5432',
    dbname 'adb',
    use_remote_estimate 'true');
```

Преимущества	Недостатки
• Автоматическое определение числа обработчиков. Пользователю не нужно знать это число заранее. • Корректная обработка изменения числа сегментов удаленного кластера — без необходимости изменения конфигурации сторонней таблицы.	Создание и открытие параллельного курсора в удаленном кластере на этапе планирования запроса

Использование числа обработчиков, равного числу локальных сегментов

В этой схеме опции `use_remote_estimate` и `num_segments` не заполняются. В результате число обработчиков статическое и совпадает с количеством сегментов локального кластера. Если число endpoints для запроса не совпадает с числом локальных сегментов, то осуществляется маппинг на локальные обработчики.

Пример определения сервера приведен ниже:

```
CREATE SERVER test_server
FOREIGN DATA WRAPPER adb_fdw
OPTIONS (
    host 'bds-mdw-ext.ru-central1.internal',
    port '5432',
    dbname 'adb');
```

Преимущества	Недостатки
Отсутствие запроса на создание и открытие параллельного курсора в удаленном кластере на этапе планирования запроса	При большем числе сегментов на удаленном кластере (по сравнению с локальным кластером) обработчики могут брать на себя обработку данных от нескольких endpoints и создавать тем самым "перекос" в обработке (processing skew). Endpoints, попавшие на один Query Executor будут обрабатываться последовательно, то есть снижается степень параллелизации обработки

Явное указание числа обработчиков

В этой схеме опции `use_remote_estimate` не указывается (по умолчанию равно `false`), однако на уровне сервера заполняется опция `num_segments`, определяющая в явном виде число обработчиков. В отличие от предыдущей схемы, можно определить отличное от числа локальных сегментов число обработчиков — например, равное числу сегментов удаленного кластера. Если число endpoints для запроса не совпадает с числом локальных сегментов, то осуществляется маппинг на локальные обработчики.

Пример определения сервера приведен ниже:

```
CREATE SERVER test_server
FOREIGN DATA WRAPPER adb_fdw
OPTIONS (
    host 'bds-mdw-ext.ru-central1.internal',
    port '5432',
    dbname 'adb',
    num_segments '16');
```

Преимущества	Недостатки
• Отсутствие запроса на создание и открытие параллельного курсора в удаленном кластере на этапе планирования запроса. • Плюсом по сравнению с предыдущей схемой является возможность устранения "перекосов" в обработке (processing skew) путем явного сопоставления числа обработчиков количеству сегментов на удаленном кластере.	• Расчет количества обработчиков необходимо выполнить вручную. Схема работы с автоопределением лишена этого недостатка. • Возможность "перекосов" в обработке (processing skew) в обработке запросов полностью не исключена (в случае неверного указания `num_segments`).

Установка ADB to ADB Connector

Дарья Барышева

Назад к содержанию

Для установки ADB to ADB Connector выполните следующие шаги:

1. Убедитесь, что сервис ADB to ADB установлен в кластере ADB. Информацию о добавлении сервисов в кластер можно получить в статье Добавление сервисов.

ВАЖНО

Сервис ADB to ADB необходимо установить в обоих кластерах ADB, между которыми планируется передавать данные.

2. Подключитесь к базе данных через `psql` (или иную клиентскую программу):

```
$ psql adb
```

3. Убедитесь, что расширение `adb_fdw` успешно добавлено. Для этого можно использовать команду `psql` `\dx`:

ПРИМЕЧАНИЕ

Если необходимо подключаться к кластеру ADB 6, убедитесь также, что расширение `gp_parallel_retrieve_cursor` установлено в кластере ADB 6. Начиная с ADB версии 7 функциональность этого расширения была интегрирована в ядро, поэтому расширение больше не требуется для взаимодействия между кластерами ADB 7.

```
\dx
```

Результат содержит расширение `adb_fdw`:

```
                                    List of installed extensions
       Name        | Version |   Schema   |                              Description
-------------------+---------+------------+-----------------------------------------------------------------------
 adb_fdw           | 1.0     | public     | foreign-data wrapper for remote ADB servers
 arenadata_toolkit | 1.0     | public     | extension is used for manipulation of objects created by adb-bundle
 dblink            | 1.2     | public     | connect to other PostgreSQL databases from within a database
 gp_exttable_fdw   | 1.0     | pg_catalog | External Table Foreign Data Wrapper for Greengage
 gp_toolkit        | 1.8     | gp_toolkit | various GPDB administrative views/functions
 kadb_fdw          | 0.16    | kadb       | Kafka-ADB foreign data wrapper
 plpgsql           | 1.0     | pg_catalog | PL/pgSQL procedural language
 pxf               | 2.1     | public     | Extension which allows to access unmanaged data
 pxf_fdw           | 1.0     | public     | PXF Foreign Data Wrapper for Greengage
 tkh_fdw           | 1.0     | public     | ClickHouse Foreign Data Wrapper for Greenplum
(10 rows)
```

4. Также можно проверить наличие новой обертки сторонних данных (foreign data wrapper) `adb_fdw`, обратившись к таблице системного каталога `pg_catalog.pg_foreign_data_wrapper`:

```
SELECT * FROM pg_catalog.pg_foreign_data_wrapper;
```

Результат:

```
  oid  |     fdwname     | fdwowner | fdwhandler | fdwvalidator | fdwacl |                                        fdoptions
-------+-----------------+----------+------------+--------------+--------+-----------------------------------------------------
-------------------------------------
 13702 | gp_exttable_fdw |       10 |      13700 |        13701 |        |
 17124 | dblink_fdw      |       10 |          0 |        17123 |        |
 17175 | adb_fdw         |       10 |      17173 |        17174 |        |
 17187 | jdbc_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=jdbc,"mpp_execute=all segments"}
 17188 | hdfs_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=hdfs,"mpp_execute=all segments"}
 17189 | hive_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=hive,"mpp_execute=all segments"}
 17190 | hbase_pxf_fdw   |       10 |      17185 |        17186 |        | {protocol=hbase,"mpp_execute=all segments"}
 17191 | s3_pxf_fdw      |       10 |      17185 |        17186 |        | {protocol=s3,"mpp_execute=all segments"}
 17192 | gs_pxf_fdw      |       10 |      17185 |        17186 |        | {protocol=gs,"mpp_execute=all segments"}
 17193 | adl_pxf_fdw     |       10 |      17185 |        17186 |        | {protocol=adl,"mpp_execute=all segments"}
 17194 | wasbs_pxf_fdw   |       10 |      17185 |        17186 |        | {protocol=wasbs,"mpp_execute=all segments"}
 17195 | file_pxf_fdw    |       10 |      17185 |        17186 |        | {protocol=file,"mpp_execute=all segments"}
 17201 | tkh_fdw         |       10 |      17200 |        17199 |        | {protocol=tkh_clickhouse,"mpp_execute=all
segments",format=text}
 17210 | kadb_fdw        |       10 |      17209 |        17211 |        | {"mpp_execute=all
segments",k_initial_offset=0,k_automatic_offsets=true,k_latency_ms=2000}
(14 rows)
```

Настройка ADB to ADB Connector

Д. Дима Барышева

Содержание

Назад к содержанию

Сервер
Сопоставление пользователя
Сторонняя таблица

Для отправки данных из локального (исходного) кластера ADB в удаленный (целевой) кластер ADB через ADB to ADB Connector необходимо предварительно создать следующие объекты на стороне локального кластера:

1. Server (сервер) — инициирует информацию о соединении с внешним источником данных.
2. User mapping (сопоставление пользователя) — обеспечивает аутентификацию пользователя во внешнем источнике данных.
3. Foreign table (сторонняя таблица) — таблица ADB, определяющая структуру внешних данных. Сторонняя таблица не хранит данные в ADB, но к ней можно обращаться с помощью запросов как к обычной таблице.

Примеры использования перечисленных объектов для отправки данных между кластерами ADB приведены в статье Примеры использования ADB to ADB Connector.

ВАЖНО

- Для каждого целевого кластера ADB, в который планируется отправлять запросы из исходного кластера ADB, достаточно создать один сервер и одно сопоставление пользователя. Сторонних таблиц может быть несколько, если отправка запросов запланирована в различные таблицы ADB.
- Все опции, приведенные в таблицах ниже (Опции сервера, Опции сопоставления пользователя, Опции сторонней таблицы), при объявлении объектов заключаются в строковом виде (в одинарных кавычках `'`). Столбец **Тип** в таблицах показывает, какой тип данных будет использоваться при валидации вводимых значений.

Сервер

Для создания сервера предназначена команда `CREATE SERVER`, базовый синтаксис которой приведен ниже:

```
CREATE SERVER <server_name> [ TYPE <server_type> ] [ VERSION '<server_version>' ]
    FOREIGN DATA WRAPPER <fdw_name>
    [ OPTIONS ( [ <option> '<value>' [, ... ]] ) ]
```

где:

- `<server_name>` — имя сервера в ADB. Должно быть уникальным в рамках текущей базы данных ADB.
- `<server_type>` — необязательный параметр, определяющий тип сервера.
- `<server_version>` — необязательный параметр, определяющий версию сервера.
- `<fdw_name>` — имя обертки сторонних данных (foreign data wrapper). Необходимо указать `adb_fdw` — foreign data wrapper, который создается автоматически после установки кластера (см. шаг 4 в статье Установка ADB to ADB Connector).
- `<option>` — параметры сервера, определяющие детали подключения к внешнему источнику данных. Список возможных опций для ADB to ADB Connector приведен в таблице Опции сервера ниже. Обратите внимание, что эти опции (за исключением `host`, `port`, `num_segments`) могут быть определены как на уровне сервера, так и на уровне пользовательских таблиц. При этом опции, отмеченные как обязательные, должны быть указаны на одном из уровней.
- `<value>` — значение соответствующих параметров `<option>`.

ПРИМЕЧАНИЕ

- Чтобы создать сервер, пользователю необходима привилегия `USAGE` для обертки сторонних данных `adb_fdw`. Пользователь, создавший сервер, становится его владельцем.
- Полную версию синтаксиса команды `CREATE SERVER` можно посмотреть в документации Greenplum.
- Для редактирования параметров сервера предназначена команда ALTER SERVER, для удаления — DROP SERVER.

Опции сервера

Имя	Тип	Описание	Default	Обязательность
host	TEXT	IP-адрес или имя хоста координатора удаленного кластера ADB. Правила валидации параметра `host` см. ниже	—	Да
port	TEXT	Номер порта для подключения к хосту координатора удаленного кластера ADB. Правила валидации параметра `port` см. ниже	—	Да
dbname	TEXT	Имя целевой БД на стороне удаленного кластера ADB	—	Да
mpp_execute	TEXT	Стратегия распределения обработки запроса. Возможно только одно значение — `coordinator`. Изменение опции `mpp_execute` не допускается. Выбор стратегии выполнения (`coordinator` или `segments` или `any`) производится коннектором ADB to ADB Connector автоматически в зависимости от вида запрошенной операции и результата планирования запроса	coordinator	Нет
use_remote_estimate	BOOL	Флаг использования внешнего кластера при планировании запроса. Используется для оценки стоимости операций и расчета числа обработчиков на сегментах. Если `use_remote_estimate = true`, запрашивается внешний кластер (создается параллельный курсор) и число обработчиков определяется согласно набору endpoints параллельного курсора. В противном случае число обработчиков берется из `num_segments`, а если этот параметр не задан — используется текущее значение количества сегментов. Дополнительную информацию можно получить в статье Обход ADB to ADB Connector. Возможные значения: `true`, `false`	false	Нет
num_segments	INT	Число обработчиков на сегментах для обработки данных, поставляемых с endpoints параллельного курсора. Если число сегментов локального кластера меньше числа endpoints, то некоторые обработчики локального кластера достаются больше endpoints (что создает "перекос" в обработке данных — processing skew). Для ввода допускаются целые положительные числа >= `1`	—	Нет
fdw_startup_cost	DOUBLE	Оценка стоимости получения первой строки от источника данных Для ввода допускаются дробные положительные числа	100.0	Нет
fdw_tuple_cost	DOUBLE	Оценка стоимости получения одной строки от источника данных Для ввода допускаются дробные положительные числа	0.01	Нет
fetch_size	INT	Число строк в батче при чтении данных от источника. Для ввода допускаются целые положительные числа >= `1`	100	Нет
updatable	BOOL	Определяет, будут ли разрешены изменения в таблицах на стороне удаленного кластера ADB посредством команды `INSERT`. Возможные значения: `true`, `false`	true	Нет

Правила валидации опций host и port

- Максимальная длина имени хоста в опции `host` — 255 символов.
- В значении опции `host` допускаются только числовые литералы `0-9`, ASCII символы `a-zA-Z`, дефис `-`, нижнее подчеркивание `_` и точка `.`.
- Максимальная длина номера порта в опции `port` — 5 символов.
- В значении опции `port` допускаются только числовые литералы `0-9`.

Сопоставление пользователя

Для создания сопоставления пользователя предназначена команда `CREATE USER MAPPING`, базовый синтаксис которой приведен ниже:

```
CREATE USER MAPPING FOR { <username> | USER | CURRENT_USER | PUBLIC }
    SERVER <server_name>
    [ OPTIONS ( <option> '<value>' [, ... ] ) ]
```

где:

- `<username>` — имя пользователя в базе данных ADB. Наряду с явным указанием пользователя, допускается ввод следующих констант:
  - `USER` или `CURRENT_USER` — текущий пользователь, устанавливающий соединение с ADB.
  - `PUBLIC` - все существующие и будущие пользователи ADB.
- `<server_name>` — имя сервера.
- `<option>` — параметры сопоставления пользователя. Как правило, включают имя пользователя и пароль для подключения к внешнему источнику данных. Список возможных опций для ADB to ADB Connector приведен в таблице Опции сопоставления пользователя ниже.
- `<value>` — значения соответствующих параметров `<option>`.

ПРИМЕЧАНИЕ

- Чтобы создать сопоставление пользователя, необходимо быть владельцем соответствующего сервера.
- Полную версию синтаксиса команды `CREATE USER MAPPING` можно посмотреть в документации Greenplum.
- Для редактирования параметров сопоставления пользователя предназначена команда ALTER USER MAPPING, для удаления — DROP USER MAPPING.

Опции сопоставления пользователя

Имя	Тип	Описание	Default	Обязательность
user	TEXT	Имя пользователя в БД удаленного кластера ADB	—	Да
password	TEXT	Пароль пользователя в БД удаленного кластера ADB	—	Да

Настройка доступов в файле pg_hba.conf

Наряду с созданием user mapping, необходимо обеспечить доступ выбранного пользователя к удаленной базе данных ADB из локального кластера. Для этого добавьте следующую запись в файл pg_hba.conf на стороне удаленного кластера ADB:

```
host    <database_name>  <remote_user>     <local_coordinator_address>     password
```

где:

- `<database_name>` — имя целевой базы данных в удаленном кластере ADB. Значение `all` соответствует всем БД.
- `<remote_user>` — имя пользователя в удаленной базе данных ADB (см. опцию `user` выше).
- `<local_coordinator_address>` — IP-адрес хоста координатора в локальном ADB-кластере (из которого планируется отправка запросов через коннектор). Необходимо использовать номер подсети. Например, `10.92.17.84/32`.

Редактирование файла pg_hba.conf возможно в web-интерфейсе ADCM. Для этого заполните параметр ***Custom pg_hba section*** на вкладке ***Configuration*** сервиса ***ADB*** в удаленном кластере ADB. Чтобы применить изменения, нажмите ***Save*** и запустите сервисное действие ***Reconfigure***.

Сторонняя таблица

Для создания сторонней таблицы предназначена команда `CREATE FOREIGN TABLE`, базовый синтаксис которой приведен ниже:

```
CREATE FOREIGN TABLE [ IF NOT EXISTS ] <table_name> ( [
    <column_name> <data_type> [ COLLATE <collation> ] [ <column_constraint> [ ... ] ]
    [, ... ] ]
) )
    SERVER <server_name>
  [ OPTIONS ( <option> '<value>' [, ... ] ) ]
```

где:

- `<table_name>` — имя сторонней таблицы в ADB.
- `<column_name>` — имя столбца.
- `<data_type>` — тип данных столбца.
- `<collation>` - используемая для столбца сортировка (collation).
- `<column_constraint>` — ограничение (constraint), определяемое на уровне столбца. Имя ограничения `<constraint_name>` указывается опционально. Синтаксис:

```
[ CONSTRAINT <constraint_name> ]
{ NOT NULL |
  NULL |
  DEFAULT <default_expr> }
```

Возможные ограничения:

- `NOT NULL` – указывает, что столбец не может содержать null значений.
- `NULL` – указывает, что столбец может содержать null значения. Это ограничение используется по умолчанию (если не указано `NOT NULL`).
- `DEFAULT` – определяет для столбца значение по умолчанию `<default_expr>`.

- `<server_name>` — имя сервера.
- `<option>` - параметры сторонней таблицы. Для ADB to ADB Connector все опции, определенные на уровне сервера (за исключением `host`, `port`, `num_segments`), могут быть переопределены на уровне сторонних таблиц, (частично или полностью). При одновременном указании какой-либо опции для сервера и таблицы табличный уровень имеет больший приоритет. Дополнительно на уровне сторонней таблицы объявляются еще три опции, приведенные в таблице Опции сторонней таблицы ниже.
- `<value>` - значения соответствующих параметров `<option>`.

ПРИМЕЧАНИЕ

- Чтобы создать стороннюю таблицу, пользователю необходима привилегия `USAGE` для соответствующего сервера, а также для всех типов данных, используемых в таблице. Пользователь, создавший стороннюю таблицу, становится ее владельцем.
- Полную версию синтаксиса команды `CREATE FOREIGN TABLE` можно посмотреть в документации Greenplum.
- Для редактирования параметров сторонней таблицы предназначена команда ALTER FOREIGN TABLE, для удаления — DROP FOREIGN TABLE.

Опции сторонней таблицы

Имя	Тип	Описание	Default	Обязательность
schema_name	TEXT	Имя схемы для обращения к таблице на стороне удаленного кластера ADB. Если параметр не указан, применяется схема таблицы `table_name`	—	Нет
table_name	TEXT	Имя целевой таблицы на стороне удаленного кластера ADB	—	Да
column_name	TEXT	Имя столбца целевой таблицы `table_name`	—	Нет

Примеры использования ADB to ADB Connector

Дарья Барышева

Содержание

Назад к содержанию

Чтение данных

Запись данных

В статье показаны примеры чтения и записи данных между двумя кластерами ADB с использованием ADB to ADB Connector. При выполнении примера соблюдены следующие условия:

- Локальный кластер ADB развернут согласно руководству Online-установка.
- Удаленный кластер ADB развернут согласно руководству Online-установка. Хост координатора кластера — `bds-mdw-ext.ru-central1.internal`.
- База данных с именем `adb` существует в обоих кластерах.
- Пользователь с именем `adb_to_adb` и паролем `123` создан в БД удаленного кластера.
- Для пользователя `adb_to_adb` настроен доступ к базе данных `adb` в удаленном кластере с хоста координатора локального кластера. Для этой цели отредактирован файл *pg_hba.conf* согласно руководству Настройка ADB to ADB connector:

```
host    all  adb_to_adb       10.92.17.84/32     password
```

  где `10.92.17.84/32` — адрес хоста координатора локального кластера ADB (с подсетью).

- ADB to ADB Connector установлен на стороне локального и удаленного кластеров согласно статье Установка ADB to ADB connector.

Чтение данных

Чтобы проверить чтение данных из удаленного кластера в локальный кластер ADB, выполните следующие шаги:

1. Подключитесь к базе данных `adb` удаленного кластера ADB под пользователем `gpadmin` (например, через `psql`) и создайте таблицу `author`:

```
CREATE TABLE author(id INT NOT NULL, name TEXT NOT NULL)
WITH (appendoptimized=true)
DISTRIBUTED BY(id);
```

2. Добавьте данные в таблицу `author` на стороне удаленного кластера:

```
INSERT INTO author(id, name) VALUES
(1,'Virginia Woolf'),
(2,'J.R.R. Tolkien'),
(3,'Harper Lee'),
(4,'J.D. Salinger'),
(5,'George Orwell'),
(6,'John Steinbeck'),
(7,'Margaret Mitchell'),
(8,'Alan Moore'),
(9,'Jack Kerouac'),
(10,'Ernest Hemingway');
```

3. Предоставьте необходимые права пользователю `adb_to_adb` на доступ к новой таблице `author` (на стороне удаленного кластера):

```
GRANT SELECT, INSERT ON public.author TO adb_to_adb;
```

4. Подключитесь к базе данных `adb` локального кластера ADB под пользователем `gpadmin` (например, через `psql`) и последовательно создайте следующие объекты:

   - Сервер `test_server`:

   ```
   CREATE SERVER test_server
   FOREIGN DATA WRAPPER adb_fdw
   OPTIONS (
     host 'bds-mdw-ext.ru-central1.internal',
     port '5432',
     dbname 'adb',
     use_remote_estimate 'true');
   ```

   - Сопоставление пользователя `gpadmin`:

   ```
   CREATE USER MAPPING FOR gpadmin
   SERVER test_server
   OPTIONS (user 'adb_to_adb', password '123');
   ```

   - Сторонняя таблица `test_table`:

   ```
   CREATE FOREIGN TABLE test_table (id INT, name TEXT)
   SERVER test_server
   OPTIONS (table_name 'author');
   ```

5. Выполните выборку данных из таблицы `test_table`. Убедитесь, что данные, добавленные на стороне удаленного кластера, доступны для чтения в локальном кластере:

```
SELECT * FROM test_table ORDER BY id;
```

   Результат:

```
 id |       name
----+-------------------
  1 | Virginia Woolf
  2 | J.R.R. Tolkien
  3 | Harper Lee
  4 | J.D. Salinger
  5 | George Orwell
  6 | John Steinbeck
  7 | Margaret Mitchell
  8 | Alan Moore
  9 | Jack Kerouac
 10 | Ernest Hemingway
(10 rows)
```

Запись данных

Чтобы проверить запись данных из локального в удаленный кластер ADB, выполните следующие шаги:

1. Выполните шаги 1—4, приведенные выше (если они не были пройдены ранее).
2. Подключитесь к базе данных `adb` локального кластера ADB под пользователем `gpadmin` (например, через `psql`) и добавьте данные в стороннюю таблицу `test_table`:

```
INSERT INTO test_table(id, name) VALUES
(11, 'Fyodor Dostoevsky'),
(12, 'Alexander Pushkin'),
(13, 'Mikhail Bulgakov');
```

3. Подключитесь к базе данных `adb` удаленного кластера ADB под пользователем `gpadmin` (например, через `psql`) и проверьте наличие новых кортежей в таблице `author`:

```
SELECT * FROM author ORDER BY id;
```

   Результат содержит новые данные, пришедшие из исходного кластера ADB:

```
 id |       name
----+-------------------
  1 | Virginia Woolf
  2 | J.R.R. Tolkien
  3 | Harper Lee
  4 | J.D. Salinger
  5 | George Orwell
  6 | John Steinbeck
  7 | Margaret Mitchell
  8 | Alan Moore
  9 | Jack Kerouac
 10 | Ernest Hemingway
 11 | Fyodor Dostoevsky
 12 | Alexander Pushkin
 13 | Mikhail Bulgakov
(13 rows)
```

Конфигурационные параметры

Версия 1.0.0

Содержание

ADB
ADB Control
Ethernet
Flash
Hub
USB

Общие параметры

Общие параметры для всех компонентов, которые поддерживают конфигурирование. **Обязательно**

ADB

Общие параметры

Параметр	Описание	Допустимые значения
enabled	Включение компонента. **Обязательно**	true, false
[vendor_id](#)	Идентификатор производителя. **Обязательно**	0x0000 - 0xffff
[product_id](#)	Идентификатор продукта. **Обязательно**	0x0000 - 0xffff
[serial_number](#)	Серийный номер. **Обязательно**	0x00000000 - 0xffffffff
[manufacturer](#)	Строка с именем производителя. **Обязательно**	0 - 255 символов
[product](#)	Строка с названием продукта. **Обязательно**	0 - 255 символов
[usb_version](#)	Версия USB. **Обязательно**	1.0, 2.0, 3.0
[class](#)	Класс устройства. **Обязательно**	0x00 - 0xff
[subclass](#)	Подкласс устройства. **Обязательно**	0x00 - 0xff
[protocol](#)	Протокол устройства. **Обязательно**	0x00 - 0xff
[max_power](#)	Максимальное потребление. **Обязательно**	0 - 255
[self_powered](#)	Собственное питание. **Обязательно**	true, false
[remote_wakeup](#)	Удаленное пробуждение. **Обязательно**	true, false
[interface_count](#)	Количество интерфейсов. **Обязательно**	1 - 255
[interface_class](#)	Класс интерфейса. **Обязательно**	0x00 - 0xff
[interface_subclass](#)	Подкласс интерфейса. **Обязательно**	0x00 - 0xff
[interface_protocol](#)	Протокол интерфейса. **Обязательно**	0x00 - 0xff
[interface_name](#)	Строка с именем интерфейса. **Обязательно**	0 - 255 символов
[endpoint_count](#)	Количество конечных точек. **Обязательно**	1 - 255
[endpoint_address](#)	Адрес конечной точки. **Обязательно**	0x00 - 0xff
[endpoint_attributes](#)	Атрибуты конечной точки. **Обязательно**	0x00 - 0xff
[endpoint_max_packet_size](#)	Максимальный размер пакета конечной точки. **Обязательно**	0 - 65535
[endpoint_interval](#)	Интервал конечной точки. **Обязательно**	0 - 255
[endpoint_data](#)	Данные конечной точки. **Обязательно**	0x00 - 0xff
[bos_capability_count](#)	Количество дескрипторов Capability. **Обязательно**	0 - 255
[bos_capability_data](#)	Данные дескрипторов Capability. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_data](#)	Данные дескриптора Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_data](#)	Данные набора дескрипторов Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_alt_enum_code](#)	Номер альтернативного перечисления набора дескрипторов Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_vendor_code](#)	Код производителя набора дескрипторов Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_windows_version](#)	Версия Windows набора дескрипторов Microsoft OS 2.0. **Обязательно** 0x06030000 - Windows 8.1 0x06040000 - Windows 10 0x06050000 - Windows 11	0x06030000, 0x06040000, 0x06050000
[device_qual_data](#)	Данные квалификации устройства. **Обязательно**	0x00 - 0xff
[device_qual_enabled](#)	Поддержка квалификации устройства. **Обязательно**	true, false
[device_qual_other_speed](#)	Поддержка квалификации устройства для другой скорости. **Обязательно**	true, false
[vendor_data](#)	Данные производителя. **Обязательно**	0x00 - 0xff
[device_to_host_vendor_enabled](#)	Поддержка запросов производителя от устройства к хосту. **Обязательно**	true, false
[host_to_device_vendor_enabled](#)	Поддержка запросов производителя от хоста к устройству. **Обязательно**	true, false

Дополнительные параметры

endpoint0

Конечная точка 0. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[max_packet_size](#)	Максимальный размер пакета. **Обязательно** 8 - Low Speed 64 - Full Speed 512 - High Speed 1024 - Super Speed	8, 64, 512, 1024
[stall_enabled](#)	Поддержка остановки. **Обязательно**	true, false
[stall_data](#)	Данные остановки. **Обязательно**	0x00 - 0xff
[buffer_size](#)	Размер буфера. **Обязательно**	1 - 65535
[buffer_data](#)	Данные буфера. **Обязательно**	0x00 - 0xff
[buffer_data_offset](#)	Смещение данных буфера. **Обязательно**	0 - 65535
[buffer_data_length](#)	Длина данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern](#)	Шаблон данных буфера. **Обязательно**	0x00 - 0xff
[buffer_data_pattern_offset](#)	Смещение шаблона данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern_length](#)	Длина шаблона данных буфера. **Обязательно**	0 - 65535

endpoint1

Конечная точка 1. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[max_packet_size](#)	Максимальный размер пакета. **Обязательно** 8 - Low Speed 64 - Full Speed 512 - High Speed 1024 - Super Speed	8, 64, 512, 1024
[stall_enabled](#)	Поддержка остановки. **Обязательно**	true, false
[stall_data](#)	Данные остановки. **Обязательно**	0x00 - 0xff
[buffer_size](#)	Размер буфера. **Обязательно**	1 - 65535
[buffer_data](#)	Данные буфера. **Обязательно**	0x00 - 0xff
[buffer_data_offset](#)	Смещение данных буфера. **Обязательно**	0 - 65535
[buffer_data_length](#)	Длина данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern](#)	Шаблон данных буфера. **Обязательно**	0x00 - 0xff
[buffer_data_pattern_offset](#)	Смещение шаблона данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern_length](#)	Длина шаблона данных буфера. **Обязательно**	0 - 65535

endpoint2

Конечная точка 2. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[max_packet_size](#)	Максимальный размер пакета. **Обязательно** 8 - Low Speed 64 - Full Speed 512 - High Speed 1024 - Super Speed	8, 64, 512, 1024
[stall_enabled](#)	Поддержка остановки. **Обязательно**	true, false
[stall_data](#)	Данные остановки. **Обязательно**	0x00 - 0xff
[buffer_size](#)	Размер буфера. **Обязательно**	1 - 65535
[buffer_data](#)	Данные буфера. **Обязательно**	0x00 - 0xff
[buffer_data_offset](#)	Смещение данных буфера. **Обязательно**	0 - 65535
[buffer_data_length](#)	Длина данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern](#)	Шаблон данных буфера. **Обязательно**	0x00 - 0xff
[buffer_data_pattern_offset](#)	Смещение шаблона данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern_length](#)	Длина шаблона данных буфера. **Обязательно**	0 - 65535

ADB Control

Общие параметры

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[connection_timeout](#)	Таймаут соединения. **Обязательно**	0 - 65535
[connection_data](#)	Данные соединения. **Обязательно**	0x00 - 0xff
[connection_data_offset](#)	Смещение данных соединения. **Обязательно**	0 - 65535
[connection_data_length](#)	Длина данных соединения. **Обязательно**	0 - 65535
[connection_data_pattern](#)	Шаблон данных соединения. **Обязательно**	0x00 - 0xff
[connection_data_pattern_offset](#)	Смещение шаблона данных соединения. **Обязательно**	0 - 65535
[connection_data_pattern_length](#)	Длина шаблона данных соединения. **Обязательно**	0 - 65535

Дополнительные параметры

[ethernet_control](#)

Управление Ethernet. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[type](#)	Тип. **Обязательно** **ethernet** - Ethernet **wifi** - Wi-Fi **bluetooth** - Bluetooth **usb** - USB **serial** - Serial port **other** - Другое	ethernet, wifi, bluetooth, usb, serial, other
[mac_enabled](#)	Поддержка MAC. **Обязательно**	true, false
[mac_data](#)	Данные MAC. **Обязательно**	0x00 - 0xff
[mac_data_offset](#)	Смещение данных MAC. **Обязательно**	0 - 65535
[mac_data_length](#)	Длина данных MAC. **Обязательно**	0 - 65535
[mac_data_pattern](#)	Шаблон данных MAC. **Обязательно**	0x00 - 0xff
[mac_data_pattern_offset](#)	Смещение шаблона данных MAC. **Обязательно**	0 - 65535
[mac_data_pattern_length](#)	Длина шаблона данных MAC. **Обязательно**	0 - 65535
[mtu](#)	MTU. **Обязательно**	0 - 65535
[mtu_data](#)	Данные MTU. **Обязательно**	0x00 - 0xff
[mtu_data_offset](#)	Смещение данных MTU. **Обязательно**	0 - 65535
[mtu_data_length](#)	Длина данных MTU. **Обязательно**	0 - 65535
[mtu_data_pattern](#)	Шаблон данных MTU. **Обязательно**	0x00 - 0xff
[mtu_data_pattern_offset](#)	Смещение шаблона данных MTU. **Обязательно**	0 - 65535
[mtu_data_pattern_length](#)	Длина шаблона данных MTU. **Обязательно**	0 - 65535
[speed](#)	Скорость. **Обязательно**	0 - 65535
[speed_data](#)	Данные скорости. **Обязательно**	0x00 - 0xff
[speed_data_offset](#)	Смещение данных скорости. **Обязательно**	0 - 65535
[speed_data_length](#)	Длина данных скорости. **Обязательно**	0 - 65535
[speed_data_pattern](#)	Шаблон данных скорости. **Обязательно**	0x00 - 0xff
[speed_data_pattern_offset](#)	Смещение шаблона данных скорости. **Обязательно**	0 - 65535
[speed_data_pattern_length](#)	Длина шаблона данных скорости. **Обязательно**	0 - 65535

[wifi_control](#)

Управление Wi-Fi. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[type](#)	Тип. **Обязательно** **ap** - Точка доступа **sta** - Станция	ap, sta
[ssid](#)	SSID. **Обязательно** Для точки доступа: имя сети Для станции: имя сети для подключения	0 - 255 символов
[ssid_data](#)	Данные SSID. **Обязательно**	0x00 - 0xff
[ssid_data_offset](#)	Смещение данных SSID. **Обязательно**	0 - 65535
[ssid_data_length](#)	Длина данных SSID. **Обязательно**	0 - 65535
[ssid_data_pattern](#)	Шаблон данных SSID. **Обязательно**	0x00 - 0xff
[ssid_data_pattern_offset](#)	Смещение шаблона данных SSID. **Обязательно**	0 - 65535
[ssid_data_pattern_length](#)	Длина шаблона данных SSID. **Обязательно**	0 - 65535
[bssid](#)	BSSID. **Обязательно** Для точки доступа: MAC-адрес точки доступа Для станции: MAC-адрес подключенной точки доступа	0x00 - 0xff
[bssid_data](#)	Данные BSSID. **Обязательно**	0x00 - 0xff
[bssid_data_offset](#)	Смещение данных BSSID. **Обязательно**	0 - 65535
[bssid_data_length](#)	Длина данных BSSID. **Обязательно**	0 - 65535
[bssid_data_pattern](#)	Шаблон данных BSSID. **Обязательно**	0x00 - 0xff
[bssid_data_pattern_offset](#)	Смещение шаблона данных BSSID. **Обязательно**	0 - 65535
[bssid_data_pattern_length](#)	Длина шаблона данных BSSID. **Обязательно**	0 - 65535
[channel](#)	Канал. **Обязательно** Для точки доступа: канал, на котором работает точка доступа Для станции: канал, на котором ожидается подключение к точке доступа	1 - 255
[channel_data](#)	Данные канала. **Обязательно**	0x00 - 0xff
[channel_data_offset](#)	Смещение данных канала. **Обязательно**	0 - 65535
[channel_data_length](#)	Длина данных канала. **Обязательно**	0 - 65535
[channel_data_pattern](#)	Шаблон данных канала. **Обязательно**	0x00 - 0xff
[channel_data_pattern_offset](#)	Смещение шаблона данных канала. **Обязательно**	0 - 65535
[channel_data_pattern_length](#)	Длина шаблона данных канала. **Обязательно**	0 - 65535
[key](#)	Ключ. **Обязательно** Для точки доступа: ключ для подключения к точке доступа Для станции: ключ для подключения к точке доступа	0 - 255 символов
[key_data](#)	Данные ключа. **Обязательно**	0x00 - 0xff
[key_data_offset](#)	Смещение данных ключа. **Обязательно**	0 - 65535
[key_data_length](#)	Длина данных ключа. **Обязательно**	0 - 65535
[key_data_pattern](#)	Шаблон данных ключа. **Обязательно**	0x00 - 0xff
[key_data_pattern_offset](#)	Смещение шаблона данных ключа. **Обязательно**	0 - 65535
[key_data_pattern_length](#)	Длина шаблона данных ключа. **Обязательно**	0 - 65535

Ethernet

Общие параметры

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[speed](#)	Скорость. **Обязательно**	10, 100, 1000, 2500, 5000, 10000
[duplex](#)	Дуплекс. **Обязательно**	half, full
[mac_data](#)	Данные MAC. **Обязательно**	0x00 - 0xff
[mac_data_offset](#)	Смещение данных MAC. **Обязательно**	0 - 65535
[mac_data_length](#)	Длина данных MAC. **Обязательно**	0 - 65535
[mac_data_pattern](#)	Шаблон данных MAC. **Обязательно**	0x00 - 0xff
[mac_data_pattern_offset](#)	Смещение шаблона данных MAC. **Обязательно**	0 - 65535
[mac_data_pattern_length](#)	Длина шаблона данных MAC. **Обязательно**	0 - 65535
[mtu](#)	MTU. **Обязательно**	0 - 65535
[mtu_data](#)	Данные MTU. **Обязательно**	0x00 - 0xff
[mtu_data_offset](#)	Смещение данных MTU. **Обязательно**	0 - 65535
[mtu_data_length](#)	Длина данных MTU. **Обязательно**	0 - 65535
[mtu_data_pattern](#)	Шаблон данных MTU. **Обязательно**	0x00 - 0xff
[mtu_data_pattern_offset](#)	Смещение шаблона данных MTU. **Обязательно**	0 - 65535
[mtu_data_pattern_length](#)	Длина шаблона данных MTU. **Обязательно**	0 - 65535
[loopback](#)	Поддержка loopback. **Обязательно**	true, false
[loopback_data](#)	Данные loopback. **Обязательно**	0x00 - 0xff
[loopback_data_offset](#)	Смещение данных loopback. **Обязательно**	0 - 65535
[loopback_data_length](#)	Длина данных loopback. **Обязательно**	0 - 65535
[loopback_data_pattern](#)	Шаблон данных loopback. **Обязательно**	0x00 - 0xff
[loopback_data_pattern_offset](#)	Смещение шаблона данных loopback. **Обязательно**	0 - 65535
[loopback_data_pattern_length](#)	Длина шаблона данных loopback. **Обязательно**	0 - 65535

Flash

Общие параметры

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[type](#)	Тип. **Обязательно** **nor** - NOR **nand** - NAND **emmc** - eMMC **sd** - SD **other** - Другое	nor, nand, emmc, sd, other
[size](#)	Размер. **Обязательно** Для NOR: размер NOR-памяти Для NAND: размер NAND-памяти Для eMMC: размер eMMC-памяти Для SD: размер SD-карты Для other: размер памяти	0 - 65535
[block_size](#)	Размер блока. **Обязательно** Для NOR: размер блока NOR-памяти Для NAND: размер блока NAND-памяти Для eMMC: размер блока eMMC-памяти Для SD: размер блока SD-карты Для other: размер блока памяти	0 - 65535
[page_size](#)	Размер страницы. **Обязательно** Для NOR: размер страницы NOR-памяти Для NAND: размер страницы NAND-памяти Для eMMC: размер страницы eMMC-памяти Для SD: размер страницы SD-карты Для other: размер страницы памяти	0 - 65535
[data](#)	Данные. **Обязательно**	0x00 - 0xff
[data_offset](#)	Смещение данных. **Обязательно**	0 - 65535
[data_length](#)	Длина данных. **Обязательно**	0 - 65535
[data_pattern](#)	Шаблон данных. **Обязательно**	0x00 - 0xff
[data_pattern_offset](#)	Смещение шаблона данных. **Обязательно**	0 - 65535
[data_pattern_length](#)	Длина шаблона данных. **Обязательно**	0 - 65535

Hub

Общие параметры

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[port_count](#)	Количество портов. **Обязательно**	1 - 255
[port_data](#)	Данные портов. **Обязательно**	0x00 - 0xff
[port_data_offset](#)	Смещение данных портов. **Обязательно**	0 - 65535
[port_data_length](#)	Длина данных портов. **Обязательно**	0 - 65535
[port_data_pattern](#)	Шаблон данных портов. **Обязательно**	0x00 - 0xff
[port_data_pattern_offset](#)	Смещение шаблона данных портов. **Обязательно**	0 - 65535
[port_data_pattern_length](#)	Длина шаблона данных портов. **Обязательно**	0 - 65535

Дополнительные параметры

[usb_hub_control](#)

Управление USB-хабом. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[port](#)	Порт. **Обязательно**	1 - 255
[port_enabled](#)	Включение порта. **Обязательно**	true, false
[port_data](#)	Данные порта. **Обязательно**	0x00 - 0xff
[port_data_offset](#)	Смещение данных порта. **Обязательно**	0 - 65535
[port_data_length](#)	Длина данных порта. **Обязательно**	0 - 65535
[port_data_pattern](#)	Шаблон данных порта. **Обязательно**	0x00 - 0xff
[port_data_pattern_offset](#)	Смещение шаблона данных порта. **Обязательно**	0 - 65535
[port_data_pattern_length](#)	Длина шаблона данных порта. **Обязательно**	0 - 65535

USB

Общие параметры

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[usb_version](#)	Версия USB. **Обязательно**	1.0, 2.0, 3.0
[speed](#)	Скорость. **Обязательно** **low** - Low Speed **full** - Full Speed **high** - High Speed **super** - Super Speed	low, full, high, super
[speed_data](#)	Данные скорости. **Обязательно**	0x00 - 0xff
[speed_data_offset](#)	Смещение данных скорости. **Обязательно**	0 - 65535
[speed_data_length](#)	Длина данных скорости. **Обязательно**	0 - 65535
[speed_data_pattern](#)	Шаблон данных скорости. **Обязательно**	0x00 - 0xff
[speed_data_pattern_offset](#)	Смещение шаблона данных скорости. **Обязательно**	0 - 65535
[speed_data_pattern_length](#)	Длина шаблона данных скорости. **Обязательно**	0 - 65535
[hub_enabled](#)	Поддержка хаба. **Обязательно**	true, false
[hub_port_count](#)	Количество портов хаба. **Обязательно**	1 - 255
[hub_port_data](#)	Данные портов хаба. **Обязательно**	0x00 - 0xff
[hub_port_data_offset](#)	Смещение данных портов хаба. **Обязательно**	0 - 65535
[hub_port_data_length](#)	Длина данных портов хаба. **Обязательно**	0 - 65535
[hub_port_data_pattern](#)	Шаблон данных портов хаба. **Обязательно**	0x00 - 0xff
[hub_port_data_pattern_offset](#)	Смещение шаблона данных портов хаба. **Обязательно**	0 - 65535
[hub_port_data_pattern_length](#)	Длина шаблона данных портов хаба. **Обязательно**	0 - 65535
[hub_port_speed](#)	Скорость порта хаба. **Обязательно**	low, full, high, super
[hub_port_speed_data](#)	Данные скорости порта хаба. **Обязательно**	0x00 - 0xff
[hub_port_speed_data_offset](#)	Смещение данных скорости порта хаба. **Обязательно**	0 - 65535
[hub_port_speed_data_length](#)	Длина данных скорости порта хаба. **Обязательно**	0 - 65535
[hub_port_speed_data_pattern](#)	Шаблон данных скорости порта хаба. **Обязательно**	0x00 - 0xff
[hub_port_speed_data_pattern_offset](#)	Смещение шаблона данных скорости порта хаба. **Обязательно**	0 - 65535
[hub_port_speed_data_pattern_length](#)	Длина шаблона данных скорости порта хаба. **Обязательно**	0 - 65535

Дополнительные параметры

[usb_control](#)

Управление USB. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[type](#)	Тип. **Обязательно** **host** - Хост **device** - Устройство	host, device
[speed](#)	Скорость. **Обязательно**	low, full, high, super
[speed_data](#)	Данные скорости. **Обязательно**	0x00 - 0xff
[speed_data_offset](#)	Смещение данных скорости. **Обязательно**	0 - 65535
[speed_data_length](#)	Длина данных скорости. **Обязательно**	0 - 65535
[speed_data_pattern](#)	Шаблон данных скорости. **Обязательно**	0x00 - 0xff
[speed_data_pattern_offset](#)	Смещение шаблона данных скорости. **Обязательно**	0 - 65535
[speed_data_pattern_length](#)	Длина шаблона данных скорости. **Обязательно**	0 - 65535
[hub_enabled](#)	Поддержка хаба. **Обязательно**	true, false
[hub_port_count](#)	Количество портов хаба. **Обязательно**	1 - 255
[hub_port_data](#)	Данные портов хаба. **Обязательно**	0x00 - 0xff
[hub_port_data_offset](#)	Смещение данных портов хаба. **Обязательно**	0 - 65535
[hub_port_data_length](#)	Длина данных портов хаба. **Обязательно**	0 - 65535
[hub_port_data_pattern](#)	Шаблон данных портов хаба. **Обязательно**	0x00 - 0xff
[hub_port_data_pattern_offset](#)	Смещение шаблона данных портов хаба. **Обязательно**	0 - 65535
[hub_port_data_pattern_length](#)	Длина шаблона данных портов хаба. **Обязательно**	0 - 65535
[hub_port_speed](#)	Скорость порта хаба. **Обязательно**	low, full, high, super
[hub_port_speed_data](#)	Данные скорости порта хаба. **Обязательно**	0x00 - 0xff
[hub_port_speed_data_offset](#)	Смещение данных скорости порта хаба. **Обязательно**	0 - 65535
[hub_port_speed_data_length](#)	Длина данных скорости порта хаба. **Обязательно**	0 - 65535
[hub_port_speed_data_pattern](#)	Шаблон данных скорости порта хаба. **Обязательно**	0x00 - 0xff
[hub_port_speed_data_pattern_offset](#)	Смещение шаблона данных скорости порта хаба. **Обязательно**	0 - 65535
[hub_port_speed_data_pattern_length](#)	Длина шаблона данных скорости порта хаба. **Обязательно**	0 - 65535

Общие

Общие параметры

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[name](#)	Имя. **Обязательно**	0 - 255 символов
[name_data](#)	Данные имени. **Обязательно**	0x00 - 0xff
[name_data_offset](#)	Смещение данных имени. **Обязательно**	0 - 65535
[name_data_length](#)	Длина данных имени. **Обязательно**	0 - 65535
[name_data_pattern](#)	Шаблон данных имени. **Обязательно**	0x00 - 0xff
[name_data_pattern_offset](#)	Смещение шаблона данных имени. **Обязательно**	0 - 65535
[name_data_pattern_length](#)	Длина шаблона данных имени. **Обязательно**	0 - 65535
[description](#)	Описание. **Обязательно**	0 - 255 символов
[description_data](#)	Данные описания. **Обязательно**	0x00 - 0xff
[description_data_offset](#)	Смещение данных описания. **Обязательно**	0 - 65535
[description_data_length](#)	Длина данных описания. **Обязательно**	0 - 65535
[description_data_pattern](#)	Шаблон данных описания. **Обязательно**	0x00 - 0xff
[description_data_pattern_offset](#)	Смещение шаблона данных описания. **Обязательно**	0 - 65535
[description_data_pattern_length](#)	Длина шаблона данных описания. **Обязательно**	0 - 65535

Дополнительные параметры

[usb_device](#)

Устройство. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[usb_version](#)	Версия USB. **Обязательно**	1.0, 2.0, 3.0
[vendor_id](#)	Идентификатор производителя. **Обязательно**	0x0000 - 0xffff
[product_id](#)	Идентификатор продукта. **Обязательно**	0x0000 - 0xffff
[serial_number](#)	Серийный номер. **Обязательно**	0x00000000 - 0xffffffff
[manufacturer](#)	Строка с именем производителя. **Обязательно**	0 - 255 символов
[product](#)	Строка с названием продукта. **Обязательно**	0 - 255 символов
[class](#)	Класс устройства. **Обязательно**	0x00 - 0xff
[subclass](#)	Подкласс устройства. **Обязательно**	0x00 - 0xff
[protocol](#)	Протокол устройства. **Обязательно**	0x00 - 0xff
[max_power](#)	Максимальное потребление. **Обязательно**	0 - 255
[self_powered](#)	Собственное питание. **Обязательно**	true, false
[remote_wakeup](#)	Удаленное пробуждение. **Обязательно**	true, false
[interface_count](#)	Количество интерфейсов. **Обязательно**	1 - 255
[interface_class](#)	Класс интерфейса. **Обязательно**	0x00 - 0xff
[interface_subclass](#)	Подкласс интерфейса. **Обязательно**	0x00 - 0xff
[interface_protocol](#)	Протокол интерфейса. **Обязательно**	0x00 - 0xff
[interface_name](#)	Строка с именем интерфейса. **Обязательно**	0 - 255 символов
[endpoint_count](#)	Количество конечных точек. **Обязательно**	1 - 255
[endpoint_address](#)	Адрес конечной точки. **Обязательно**	0x00 - 0xff
[endpoint_attributes](#)	Атрибуты конечной точки. **Обязательно**	0x00 - 0xff
[endpoint_max_packet_size](#)	Максимальный размер пакета конечной точки. **Обязательно**	0 - 65535
[endpoint_interval](#)	Интервал конечной точки. **Обязательно**	0 - 255
[endpoint_data](#)	Данные конечной точки. **Обязательно**	0x00 - 0xff
[bos_capability_count](#)	Количество дескрипторов Capability. **Обязательно**	0 - 255
[bos_capability_data](#)	Данные дескрипторов Capability. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_data](#)	Данные дескриптора Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_data](#)	Данные набора дескрипторов Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_alt_enum_code](#)	Номер альтернативного перечисления набора дескрипторов Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_vendor_code](#)	Код производителя набора дескрипторов Microsoft OS 2.0. **Обязательно**	0x00 - 0xff
[ms_os_20_descriptor_set_windows_version](#)	Версия Windows набора дескрипторов Microsoft OS 2.0. **Обязательно** 0x06030000 - Windows 8.1 0x06040000 - Windows 10 0x06050000 - Windows 11	0x06030000, 0x06040000, 0x06050000
[device_qual_data](#)	Данные квалификации устройства. **Обязательно**	0x00 - 0xff
[device_qual_enabled](#)	Поддержка квалификации устройства. **Обязательно**	true, false
[device_qual_other_speed](#)	Поддержка квалификации устройства для другой скорости. **Обязательно**	true, false
[vendor_data](#)	Данные производителя. **Обязательно**	0x00 - 0xff
[device_to_host_vendor_enabled](#)	Поддержка запросов производителя от устройства к хосту. **Обязательно**	true, false
[host_to_device_vendor_enabled](#)	Поддержка запросов производителя от хоста к устройству. **Обязательно**	true, false

Дополнительные параметры

endpoint0

Конечная точка 0. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[max_packet_size](#)	Максимальный размер пакета. **Обязательно** 8 - Low Speed 64 - Full Speed 512 - High Speed 1024 - Super Speed	8, 64, 512, 1024
[stall_enabled](#)	Поддержка остановки. **Обязательно**	true, false
[stall_data](#)	Данные остановки. **Обязательно**	0x00 - 0xff
[buffer_size](#)	Размер буфера. **Обязательно**	1 - 65535
[buffer_data](#)	Данные буфера. **Обязательно**	0x00 - 0xff
[buffer_data_offset](#)	Смещение данных буфера. **Обязательно**	0 - 65535
[buffer_data_length](#)	Длина данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern](#)	Шаблон данных буфера. **Обязательно**	0x00 - 0xff
[buffer_data_pattern_offset](#)	Смещение шаблона данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern_length](#)	Длина шаблона данных буфера. **Обязательно**	0 - 65535

endpoint1

Конечная точка 1. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[max_packet_size](#)	Максимальный размер пакета. **Обязательно** 8 - Low Speed 64 - Full Speed 512 - High Speed 1024 - Super Speed	8, 64, 512, 1024
[stall_enabled](#)	Поддержка остановки. **Обязательно**	true, false
[stall_data](#)	Данные остановки. **Обязательно**	0x00 - 0xff
[buffer_size](#)	Размер буфера. **Обязательно**	1 - 65535
[buffer_data](#)	Данные буфера. **Обязательно**	0x00 - 0xff
[buffer_data_offset](#)	Смещение данных буфера. **Обязательно**	0 - 65535
[buffer_data_length](#)	Длина данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern](#)	Шаблон данных буфера. **Обязательно**	0x00 - 0xff
[buffer_data_pattern_offset](#)	Смещение шаблона данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern_length](#)	Длина шаблона данных буфера. **Обязательно**	0 - 65535

endpoint2

Конечная точка 2. **Обязательно**

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[max_packet_size](#)	Максимальный размер пакета. **Обязательно** 8 - Low Speed 64 - Full Speed 512 - High Speed 1024 - Super Speed	8, 64, 512, 1024
[stall_enabled](#)	Поддержка остановки. **Обязательно**	true, false
[stall_data](#)	Данные остановки. **Обязательно**	0x00 - 0xff
[buffer_size](#)	Размер буфера. **Обязательно**	1 - 65535
[buffer_data](#)	Данные буфера. **Обязательно**	0x00 - 0xff
[buffer_data_offset](#)	Смещение данных буфера. **Обязательно**	0 - 65535
[buffer_data_length](#)	Длина данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern](#)	Шаблон данных буфера. **Обязательно**	0x00 - 0xff
[buffer_data_pattern_offset](#)	Смещение шаблона данных буфера. **Обязательно**	0 - 65535
[buffer_data_pattern_length](#)	Длина шаблона данных буфера. **Обязательно**	0 - 65535

Ошибка

Общие параметры

Параметр	Описание	Допустимые значения
[enabled](#)	Включение компонента. **Обязательно**	true, false
[error_code](#)	Код ошибки. **Обязательно** 0x00000000 - Нет ошибки 0x00000001 - Неизвестная ошибка 0x00000002 - Ошибка выделения памяти 0x00000003 - Ошибка доступа к памяти 0x00000004 - Ошибка доступа к файлу 0x00000005 - Ошибка доступа к устройству 0x00000006 - Ошибка доступа к сети 0x00000007 - Ошибка доступа к процессу 0x00000008 - Ошибка доступа к потоку 0x00000009 - Ошибка доступа к синхронизации 0x0000000a - Ошибка доступа к таймеру 0x0000000b - Ошибка доступа к прерыванию 0x0000000c - Ошибка доступа к порту 0x0000000d - Ошибка доступа к шине 0x0000000e - Ошибка доступа к устройству 0x0000000f - Ошибка доступа к драйверу 0x00000010 - Ошибка доступа к ресурсу 0x00000011 - Ошибка доступа к объекту 0x00000012 - Ошибка доступа к сервису 0x00000013 - Ошибка доступа к протоколу 0x00000014 - Ошибка доступа к интерфейсу 0x00000015 - Ошибка доступа к данным 0x00000016 - Ошибка доступа к конфигурации 0x00000017 - Ошибка доступа к параметрам 0x00000018 - Ошибка доступа к настройкам 0x00000019 - Ошибка доступа к системе 0x0000001a - Ошибка доступа к ядру 0x0000001b - Ошибка доступа к модулю 0x0000001c - Ошибка доступа к библиотеке 0x0000001d - Ошиб

Поддерживаемые расширения

Дарья Борисова

AD.B

Назад к содержанию

Расширения (extension) — это специальные модули, обеспечивающие дополнительную функциональность базы данных: функции, операторы, типы данных и т.д. Каждое расширение представляет собой несколько объектов SQL, сгруппированных в рамках единого пакета для совместного использования.

Ниже представлен полный список расширений, доступных в Arenadata DB.

Имя расширения	Версия	Описание	Только Enterprise	Способ установки
adb_fdw	1.0	Обертка сторонних данных (foreign data wrapper), используемая для обмена данными между двумя кластерами ADB	+	ADB to ADB Connector
address_standardizer	3.3.2	Основанный на правилах (rule) стандартизатор, который используется для парсинга адреса на составляющие элементы	-	ADB service → Install PostGIS action
address_standardizer_data_us	3.3.2	Предоставляет примеры таблиц правил для адресных данных, используемых в США	-	ADB service → Install PostGIS action
arenadata_toolkit	1.0	Используется для работы со схемой arenadata_toolkit и ее объектами	-	ADB service
btree_gin	1.3	Обеспечивает GIN-индексацию основных типов данных	-	`CREATE EXTENSION btree_gin;`
citext	1.6	Реализует функции для нечувствительных к регистру строк	-	`CREATE EXTENSION citext;`
dblink	1.2	Обеспечивает подключение к другим базам данных PostgreSQL в рамках сессии к текущей БД	-	ADB service
diskquota	2.3	Позволяет ограничить количество дискового пространства, используемого схемами, ролями или табличными пространствами в БД	-	ADB service → Install diskquota action
file_fdw	1.0	Реализует обертку сторонних данных (foreign data wrapper), которая позволяет работать с файлами в файловой системе сервера	-	`CREATE EXTENSION file_fdw;`
fuzzystrmatch	1.1	Предоставляет набор функций для определения сходств и расстояний (например, Левенштейна) между строками на базе различных алгоритмов	-	`CREATE EXTENSION fuzzystrmatch;`
gp_distribution_policy	1.0	Обеспечивает проверку политик распределения данных в кластере	-	`CREATE EXTENSION gp_distribution_policy;`
gp_exttable_fdw	1.0	Преобразует внешнюю таблицу в стороннюю таблицу	-	`CREATE EXTENSION gp_exttable_fdw;`
gp_internal_tools	1.0.0	Обеспечивает разнообразные внутренние инструменты для Greenpage DB	-	`CREATE EXTENSION gp_internal_tools;`
gp_legacy_string_agg	1.0.0	Реализует legacy-функцию `string_agg` с одним аргументом для Greenpage DB	-	`CREATE EXTENSION gp_legacy_string_agg;`
gp_sparse_vector	1.0.1	Обеспечивает реализацию SParse-векторов для Greenpage DB	-	`CREATE EXTENSION gp_sparse_vector;`
gp_toolkit	1.8	Предоставляет информацию из системных каталогов и логов кластера, а также состояние окружения	-	`CREATE EXTENSION gp_toolkit;`
gpadcc	0.8	Используется для сервиса ADB Control	+	ADB Control service
hstore	1.6	Реализует тип данных для хранения пары ключ/значение в рамках одного табличного столбца	-	`CREATE EXTENSION hstore;`
isn	1.2	Обеспечивает поддержку международных стандартов нумерации продуктов	-	`CREATE EXTENSION isn;`
kadb_fdw	0.16	Обертка сторонних данных (foreign data wrapper), используемая для взаимодействия ADB и Kafka	+	Kafka to ADB Connector
ltree	1.1	Реализует тип данных для хранения иерархических древовидных структур	-	`CREATE EXTENSION ltree;`
madlib	2.1.0	Библиотека для масштабируемой аналитики в базе данных	-	ADB service → Install MADlib action
orafce	4.9	Предоставляет функции и операторы SQL, эмулирующие набор пакетов и функций из СУБД Oracle	-	`CREATE EXTENSION orafce;`
pageinspect	1.9	Используется для ненизкоуровневой инспекции страниц баз данных	-	`CREATE EXTENSION pageinspect;`
pg_buffercache	1.4.1	Позволяет анализировать содержимое общего буферного кеша PostgreSQL в реальном времени	-	`CREATE EXTENSION pg_buffercache;`
pg_hint_plan	1.3.9	Позволяет влиять на план выполнения запросов, формируемый планировщиком, с использованием подсказок (hint), встраиваемых непосредственно в SQL-комментарии	-	`CREATE EXTENSION pg_hint_plan;`

pg_stat_statements	1.7	Позволяет отслеживать статистику выполнения сервером всех операторов SQL	-	В базе данных `adb` выполните команду:
				`CREATE EXTENSION pg_stat_statements;`
				На хосте координатора добавьте `pg_stat_statements` в значение серверного конфигурационного параметра `shared_preload_libraries`, и перезапустите СУБД:
				`$ gpstop -r`

pg_trgm	1.4	Обеспечивает определение сходств текстов и индексный поиск на основе сопоставления триграмм (trigram)	-	`CREATE EXTENSION pg_trgm;`
pgcrypto	1.3	Реализует криптографические функции (шифрование/дешифрование)	-	`CREATE EXTENSION pgcrypto;`
plperl	1.0	Обеспечивает поддержку процедурного языка PL/Perl	-	`CREATE EXTENSION plperl;`
plperlu	1.0	Обеспечивает поддержку процедурного языка PL/PerlU (untrusted)	-	`CREATE EXTENSION plperlu;`
plpgsql	1.0	Обеспечивает поддержку процедурного языка PL/pgSQL	-	ADB service
plpython3u	1.0	Обеспечивает поддержку процедурного языка PL/Python3U (untrusted)	-	`CREATE EXTENSION plpython3u;`
plpythonu	1.0	Обеспечивает поддержку процедурного языка PL/PythonU (untrusted)	-	`CREATE EXTENSION plpythonu;`
postgis	3.3.2	Обеспечивает поддержку геометрии, географии и растровых пространственных типов и функций PostGIS	-	ADB service → Install PostGIS action
postgis_raster	3.3.2	Обеспечивает поддержку растровых (картографических) данных	-	ADB service → Install PostGIS action
postgis_sfcgal	3.3.2	Предоставляет SFCGAL-функции PostGIS	-	ADB service → Install PostGIS action
postgis_tiger_geocoder	3.3.2	Предоставляет TIGER-геокодер и обратный геокодер в PostGIS	-	ADB service → Install PostGIS action
postgres_fdw	1.0	Обертка сторонних данных (foreign data wrapper), используемая для подключения к удаленной БД PostgreSQL	-	`CREATE EXTENSION postgres_fdw;`
pxf	2.1	Позволяет получить доступ к неупрашляемым данным	-	PXF service → ***install*** action
pxf_fdw	1.0	Реализует обертку сторонних данных (foreign data wrapper) для PXF	-	PXF service → ***install*** action
sslinfo	1.2	Используется для получения информации об SSL-сертификатах	-	`CREATE EXTENSION sslinfo;`
tablefunc	1.1	Предоставляет функции, возвращающие таблицы (например, `gpnetstat`)	+	`CREATE EXTENSION tablefunc;`
tkh_fdw	1.0	Реализует обертку сторонних данных (foreign data wrapper), используемую для взаимодействия ADB и ClickHouse	+	ADB ClickHouse Connector
uuid-ossp	1.1	Предоставляет функции для генерации универсальных уникальных идентификаторов (Universally Unique Identifier, UUID)	-	`CREATE EXTENSION "uuid-ossp";`

Список команд SQL, не поддерживаемых в ADB

Дарья Барышева

Назад к содержанию

Несмотря на то, что в основе Arenadata DB используется PostgreSQL, часть команд и операторов языка SQL не поддерживается в ADB. Это обусловлено особенностями распределенной базы данных. Ниже приведен список SQL-команд, отсутствующих в ADB.

ПРИМЕЧАНИЕ

С полным перечнем SQL-команд (включая поддерживаемые) можно ознакомиться в документации Greenplum.

SQL-команда
ALTER SYSTEM
ALTER TRIGGER
ALTER PUBLICATION
ALTER SUBSCRIPTION
COMMIT PREPARED
CREATE CONSTRAINT TRIGGER
CREATE PUBLICATION
CREATE SUBSCRIPTION
CREATE TRIGGER
DROP OWNED
DROP PUBLICATION
DROP SUBSCRIPTION
DROP TRIGGER
JOIN-тип `LATERAL`
PREPARE TRANSACTION
ROLLBACK PREPARED
SET CONSTRAINTS

Релизы ADB 7

Дарья Барышева

Содержание

Назад к содержанию

7.4.1
7.4.0
7.2.0

7.4.1

7.4.1.1

Дата: 23.12.2025

Новые функции

Выпуск ADBM 2.8.1

Выпуск ADB Control 4.15.1

Исправленные баги

Кластерные действия ***Move Host Segment***, ***Expand*** и ***Init Standby Master*** теперь перезагружают только новые хосты

Исправлена ошибка, возникавшая при переключении на резервный координатор в режиме ***force activation***

Исправлена ошибка закрытия параллельного курсора, приводившая к зависанию коннектора ADB to ADB

Исправлена ошибка чтения из сторонней таблицы внутри транзакции

Исправлена ошибка с отправкой команд `DROP` на сегменты, когда объект отсутствует на координаторе

Исправлена ошибка, из-за которой временные файлы могли оставаться в системе, если запрос с материализованным CTE прерывался ошибкой

7.4.0

7.4.0.1

Дата: 28.11.2025

Новые функции

Релиз включает следующие компоненты: ADBM, ADB Control и коннекторы (ADB ClickHouse, ADB to Kafka и Kafka to ADB, ADB to ADB). Поддерживается offline-установка

ADB теперь использует Greengage DB в качестве основы

DBeaver адаптирован для работы с Greengage DB. Используйте версию 25.2.3 или новее

Добавлена поддержка Linux cgroup v2, позволяющая ограничивать пропускную способность диска

Реализована функциональность по очистке "брошенных" (orphaned) файлов

Добавлена возможность задавать ключ распределения для `Redistribute Motion`, при использовании внешних таблиц

Добавлена поддержка LDAP-аутентификации

Добавлена поддержка LLVM 18

Реализованы параметры TLS для внешних таблиц PXF и оберток сторонних данных (FDW)

Выпуск ADBM 2.8.0

Выпуск ADB Control 4.15.0

Улучшения

Проведен рефакторинг `arenadata_toolkit`. Расширение приведено в соответствие с актуальной архитектурой Greengage DB, несколько компонентов переработана, часть функциональности перенесена в расширение `gp_toolkit`

Реализована предварительная проверка, предотвращающая добавление резервного координатора при отключенном зеркалировании

Переработана обработка параметров ОС и добавлена предварительная проверка необходимых системных настроек

Добавлена поддержка выбора типа зеркалирования `group` или `spread`

Добавлена поддержка SSL для соединений между `pxf_fdw` и PXF-сервисом

Включена поддержка макроса `DEXTRA_DYNAMIC_MEMORY_DEBUG`

CTE с DML больше нельзя выполнять в виде встроенного (inline) запроса

Добавлена ошибка планировщика для запросов `SELECT INTO`, содержащих модифицирующие CTE

Добавлен вывод статистики в формате JSON для `EXPLAIN ANALYZE`

Предотвращено удаление WAL-файлов при восстановлении, если `archive_mode=always`

Добавлен явный вывод ошибки при указании `ENCODING` для столбцов внешних таблиц

Добавлена статистика `EXPLAIN ANALYZE` для запросов с DML внутри CTE

Добавлен флаг трансформации по умолчанию для `gpfdist`, который применяется автоматически, если для таблицы не задан уровень трансформации

Обеспечена совместимость `gpcloud` с OpenSSL 1.1.1

Добавлена проверка, что предыдущая операция расширения кластера успешно завершена перед запуском следующей

Исправленные баги

Исправлена проблема, из-за которой временные схемы не очищались после сбоев сегментов

Исправлена проблема, из-за которой `pg_dump` генерировал некорректный оператор `DROP EXTERNAL TABLE`, препятствующий корректному восстановлению

Исправлена работа `gplogfilter` — фильтры по диапазону времени теперь корректно пропускают ненужные файлы

Исправлена проблема, из-за которой дисковое пространство не освобождалось своевременно после удаления временных таблиц

Исправлена проблема, из-за которой GUC-параметры на сегментах не обновлялись при выполнении `gpstop -u`

Исправлен сбой при запуске `EXPLAIN (ANALYZE, FORMAT JSON)` для `gp_toolkit.gp_table_size_skew_coefficients()`

Исправлена проблема, из-за которой вставки завершались ошибкой, если `default_tablespace` указывал на несуществующее табличное пространство

Исправлен некорректный расчет места, занимаемого spill-файлами при запуске нескольких процессов в одной сессии

Исправлена проблема, из-за которой `VACUUM` мог повредить bitmap-индекс AO-таблицы на зеркальных сегментах

Исправлена ошибка, при которой LDAP-подключения игнорировали `hostssl`, и не использовали SSL для указанных пользователей

Исправлена ошибка, препятствовавшая использованию подстановочных (wildcard) символов в адресах интерконнекта

Исправлена ошибка в расширении `tablefunc`, не позволявшая использовать DML-запросы

Исправлено некорректное выделение памяти при генерации статистики для SQL-запросов с хеш-соединениями (hash join)

Прочее

MADlib теперь поставляется как расширение

7.2.0

7.2.0.1

Дата: 19.09.2024

Улучшения

Исправлен вывод команды `EXPLAIN` в формате JSON для предоставления более точных деталей выполнения запроса

В целях улучшения качества сборки компиляция GPDB теперь завершается ошибкой при получении предупреждений от компилятора

Реализовано уведомление об ошибках в запросах, включающих CTE в рамках выражений `SELECT INTO`

Прерывание LDAP-запросов сигналами более не рассматривается как ошибка аутентификации, что повышает стабильность

Добавлена официальная поддержка Ubuntu 22.04 для платформ x86-64

Логика анализа запросов перемещена в ядро PostgreSQL с целью улучшения производительности и интеграционных возможностей

Упрощена логика определения `dbid` для повышения надежности команды `pg_rewind`

Выполнена оптимизация расчета числа строк для реплицированных таблиц в ходе выполнения DML-операций

Логика распределения данных обновлена для улучшения обработки спецификаций combined hash

Добавлена поддержка Python 3.11 с целью улучшения операций с контейнерами

Механизм обработки CTE обновлен так, чтобы всегда материализовывать данные, используемые вне `SELECT`

Включены последние обновления данных с типом `timezone` для более точного управления временными зонами

Обновлена поддержка JIT для улучшения производительности запросов в GPDB

Добавлена документация для запуска JIT-тестов, и добавлена поддержка LLVM для последних версий

Внесены исправления в Dockerfile для корректного функционирования GPDB 7, а также добавлена поддержка Java 11

Исправленные баги

Исправлена утечка ресурсов во время ошибок при выполнении команд `COPY TO/FROM PROGRAM`

Исправлены предупреждения о компиляции для улучшения совместимости с современными компиляторами

Исправлена проблема с изменением метода доступа к таблице на колоночный

Исправлена проблема, вызывавшая ошибку при расчете коэффициента сжатия таблиц AO

Исправлена ошибка, связанная с очисткой временных схем после падения сегментов

Исправлены ошибки, связанные с некорректным доступом к строкам в коде анализа таблиц AOCO

Логика команды `gpstop -u` доработана с целью корректного применения обновленных GUC на сегментах

Известные проблемы ADB 7

Антон Васильев

Назад к содержанию

Проблема	С версии	Fixed	Обходное решение

Задача Ansible зависает на определенном хосте во время установки пакетов или обновления APT-кеша	7.4.0	—	Решение:
			1. На хосте, на котором зависла установка, найдите все запущенные процессы, связанные с APT:
			`$ ps aux | grep apt`
			2. Завершите найденные процессы:
			`$ kill <PID>`
			После завершения зависших процессов задача автоматически перезапустится. Убедитесь, что завершили зависшие процессы на всех затронутых хостах — в этом случае задача в ADCM возобновится, и вы сможете запустить ее заново в веб-интерфейсе ADCM.
			В качестве профилактической меры отключите и остановите `apt-daily-upgrade.timer`, `apt-daily.timer` и `apt-daily.service`, чтобы предотвратить блокировку APT фоновыми автоматическими обновлениями:
			`$ sudo systemctl disable apt-daily-upgrade.timer apt-daily.timer apt-daily.service`
			`$ sudo systemctl stop apt-daily-upgrade.timer apt-daily.timer apt-daily.service`

Известные проблемы ADB 6

Последнее обновление: 2025-05-05

Назад к содержимому

Проблема	Из версии	Статус	Обходное решение
В результате `VACUUM` могут пропадать данные в таблицах с `PRIMARY KEY` `INTEGER` и `AUTOINCREMENT`, если при последовательном выполнении `ALTER TABLE` и `DELETE`	—	—	Не устанавливайте и не обновляйте пакет `pbkdf2/0.0.1`. Вместо него используйте `pbkdf2/0.0.2` — в нем исправлены ошибки в обработке `Buffer` и устранены уязвимости `CVE-2025-47906`, `CVE-2025-47907`.
Для запросов со сложным, вложенным `WHERE` оптимизатор выбирает неоптимальный план, если `sqo_hoist=0` или `sqo_write_pushdown=0`	—	—	Используйте `SELECT DISTINCT ON (begin, end)`, `ORDER BY`, `GROUP BY` и вложенные `JOIN` вместо `WHERE` — такие запросы выполняются быстрее.
`sqo_hoist=0` `sqo_write_pushdown=0` `sqo_hoist=1` `sqo_write_pushdown=1`	—	—	Случаи неоптимальных планов для `sqo_write_pushdown=1`: `SELECT DISTINCT FR (begin, seos_id, command_id, prefix, ind, seos_id, seqn, command_id, prefix, ind) FROM sqo_hoist=1 sqo_write_pushdown=1 SEQUENCE SCAN seqn, seos_id, command_id, prefix, ind;` Вложенные подзапросы с пустым `ORDER BY` или `GROUP BY` — `SELECT seos_id, seqn, MAX(cnt) AS total_size, SUM(cnt) AS total_fsize FROM (SELECT DISTINCT ON (seos_id, command_id, prefix) seqn, size, cnt, nsfiles FROM sqo_hoist=1 sqo_write_pushdown=1 ORDER BY seqn) GROUP BY seos_id, seqn;`
Описание пункта `STRUCT` `union` `ALTER REVOKED GROUP` в разделе «Изменения в версиях»	6.21.1/37	—	Используйте отсортированные списки, предустановленные в первом столбце
PHP не всегда устанавливает cURL даже при наличии `--with-libcurl` или `--with-curl`	6.21.1/37	—	Чтобы обеспечить надёжную обработку типа `Timestamp with time zone` / `Date` / `Time` / `Interval` рекомендуется передавать параметр `data_size_response=true`.
Некоторые параметры `SELECT` `DISTINCT` `ORDER BY` `GROUP BY` `LIMIT` не учитываются в кешах, а при повторе запроса выполняются заново. Кроме `DISTINCT`, оптимизация работает для `LIMIT`	6.21.2/52	—	При перечислении не в `SELECT` `DISTINCT` `ON` `ORDER BY` `GROUP BY` `LIMIT` `OFFSET` `FROM` `WHERE` `HAVING` `WINDOW` `QUALIFY` `ORDER

Релизы ADB 5

Д. Дарья Барышева

ADB

Содержание

Назад к содержанию

5.29.11
5.29.8
5.29.6
5.29.5
5.29.4
5.29.3
5.29.2
5.29.1
5.29.0
5.28.14
5.28.13
5.28.11
5.28.9
5.28.7
5.28.6

5.29.11

5.29.11.19

Дата: 16.06.2023

Новые функции

Синхронизация с Greenplum Database 5.29.11

Улучшения

Обновление `tkh-connector` и `gptkh` до последних версий PXF 5

Исправленные баги

Исправлена ошибка `FATAL: could not open relation`, возникавшая после переключения сегментов

Исправлено создание расширения `gptkh` в случае, когда скрипты расширения удалили что-либо

Исправлена ошибка `ERROR: gp_relation_node has more entries than pg_aocsseg for relations`

5.29.8

5.29.8.18

Дата: 13.10.2022

Новые функции

Синхронизация с Greenplum Database 5.29.8

Улучшения

Удаление relcache в табличных пространствах (tablespace), размещенных не по дефолтному пути

5.29.6

5.29.6.18

Дата: 15.06.2022

Новые функции

Синхронизация с Greenplum Database 5.29.6

5.29.5

5.29.5.18

Дата: 18.04.2022

Новые функции

Синхронизация с Greenplum Database 5.29.5

5.29.4

5.29.4.18

Дата: 03.03.2022

Новые функции

Синхронизация с Greenplum Database 5.29.4

5.29.3

5.29.3.18

Дата: 24.01.2022

Новые функции

Синхронизация с Greenplum Database 5.29.3

Исправленные баги

Ошибка с тестом `starve_case isolation2` для ORCA

5.29.2

5.29.2.17

Дата: 22.12.2021

Новые функции

Синхронизация с Greenplum Database 5.29.2

5.29.1

5.29.1.17

Дата: 26.11.2021

Новые функции

Синхронизация с Greenplum Database 5.29.1

Обновление `gpbackup` до версии 1.22.0_arenadata5

5.29.0

5.29.0.16

Дата: 03.11.2021

Новые функции

Синхронизация с Greenplum Database 5.29.0

5.28.14

5.28.14.16

Дата: 30.08.2021

Новые функции

Синхронизация с Greenplum Database 5.28.14

Обновление `pgbouncer` до версии 1.16.0

Обновление PXF до версии 5.16.3_arenadata2

Обновление `gpbackup` до версии 1.21.0_arenadata3

Обновление `gpbackup_s3_plugin` до версии 1.7.0_arenadata1

Исправленные баги

Ограничено выполнение многоуровневых коррелированных запросов

5.28.13

5.28.13.16

Новые функции

Синхронизация с Greenplum Database 5.28.13

5.28.11

5.28.11.15

Новые функции

Синхронизация с Greenplum Database 5.28.11

Исправленные баги

Сканирование индексов в коррелированных подзапросах приводило к некорректным результатам

5.28.9

5.28.9.14

Новые функции

Синхронизация с Greenplum Database 5.28.9

Исправленные баги

[5X backport] Функция etsg игнорировала кодировку multi-byte при обрезке длинных сообщений

[5X backport] Механизм PartitionSelector не возвращал кортежи в случае пропуска типа данных

5.28.7

5.28.7.13

Новые функции

Синхронизация с Greenplum Database 5.28.7

Исправленные баги

Исправлен некорректный URL в описании сервиса ADB

[5X backport] ORCA строила некорректный план для выражения `NOT IN`

5.28.6

5.28.6.11

Новые функции

Синхронизация с Greenplum Database 5.28.6

Исправленные баги

Исправлена ошибка, возникавшая при выполнении `gpexpand` для таблиц с точкой в имени

Исправлен сбой в очереди ожидания для ресурсных групп

Исправлена ошибка `SEGFAULT`, возникавшая на этапе перераспределения таблиц в ходе выполнения `gpexpand`

Previous NOM Control

Sample Number

Sample Number

Sample Number

Известные проблемы ADB Control

Дарья Барышева

AD.B CONTROL

Назад к содержанию

Проблема	С версии	Fixed	Обходное решение
Кластеры, подключенные к внешнему ADB Control, должны быть обновлены вместе с основным кластером до ADB 6.29.0.3 для корректной работы ADBM и ADB Control	4.14.2	—	Обновите кластеры до ADB 6.29.0.3

В AltLinux 8 обновление Docker engine может повредить пакет *docker-compose*, что приводит к ошибкам в запуске сервиса ADB Control	—	На стороне AltLinux	Исправьте */usr/lib/python3/site-packages/compose/service.py* путем замены:
			`container.image_config['ContainerConfig'].get('Volumes') or {}`
			на:
			`container.image_config['Config'].get('Volumes') or {}`

Запуск действия ***Reconfigure*** может завершиться ошибкой для сервиса ADB Control на хосте с операционной системой CentOS в случае использования внешней базы Query DB, поскольку в репозитории CentOS отсутствует необходимая версия PostgreSQL	4.13.0	—	Выполните следующие шаги: 1. Удалите полностью PostgreSQL 9 с хоста с ADB Control. 2. Установите клиент PostgreSQL версии 14 или выше. При этом не следует размещать ADB Control на мастер-хосте ADB
При настройке LDAP-аутентификации для ADB Control с типом `FreeIPA` список групп не загружается при создании нового сопоставления с группами LDAP-сервера на вкладке ***Users → LDAP group mapping***	4.0.0	—	Используйте базовую аутентификацию или LDAP с типом `MSAD` (Microsoft Active Directory)
При обновлении кластера с версии 6.27.1.56 до 6.27.1.58 возникает ошибка при использовании внешнего ADB Control	4.10.3	4.10.3	Обновите кластер с сервисом ADB Control до обновления кластеров, которые используют этот ADB Control в качестве внешнего (в поле ***External ADB Control server parameters***)
После отмены транзакции узлы плана в секции Plan & progress будут по-прежнему иметь статус `executing` и отображать увеличивающееся время выполнения	4.8.8	4.9.1	Необходимо иметь в виду, что у отмененных запросов нет выполняющихся процессов
В ADB Control 4.8.8 реализованы изменения в API, затрагивающие взаимодействие между основным и внешним сервисами ADB Control разных версий. Когда кластер ADB с основным ADB Control обновляется до версии 6.27.1.56.b1, в кластерах более ранних версий, использующих этот ADB Control как внешний, будет невозможно выполнять действия ***Reconfigure***/***Start*** с сервисом ADB Control	4.8.8	4.8.8	Чтобы обеспечить работоспособность всех действий сервиса ADB Control, обновите кластеры с внешним ADB Control и версией 6.27.1.56.b1 и ниже до версии 6.27.1.56.b1
При обновлении ADB Control до версии 4.3.2 настройки задачи `Export job` (заполненные на странице ***Configuration → Job policy → Metrics offload***) возвращаются к значениям по умолчанию	4.3.2	4.3.2	Заполните и сохраните необходимые настройки. Проблема не затрагивает последующие обновления
Активация резервного мастера (standby) после переключения с мастера завершается с ошибкой в кластерах с установленным ADB Control/ADBM	4.3.1	4.3.2	Используйте внешний ADB Control/ADBM
В ADB Control 4.2.1 невозможно зарегистрировать внешние кластеры ADB с версиями ниже 6.23.3	4.2.1	4.2.1	Обновите кластер ADB до версии 6.23.3

Отключение отправки метрик вложенных запросов в ADB Control может вызвать "зависание" ADB-запросов	4.1.0	4.10.3	Включите GUC, отвечающий за отправку метрик вложенных запросов, в ADB Control или в ADB (как показано ниже):
			`$ gpconfig -c adcc.monitor_inner_queries -v on`
			`$ gpstop -u`

Команды появляются в статусе `Running` на странице ***Monitoring → Commands → Online***, но спустя 14-16 секунд переходят в статус `Unknown` (без обновления метрик). После завершения команды все метрики становятся доступны на странице ***Monitoring → Commands → History***, но статус `Unknown` сохраняется	4.1.0	4.2.0	1. Используя `psql` или иной SQL IDE, установите пароль для пользователя `adcc` с помощью следующей команды: `ALTER USER adcc WITH ENCRYPTED PASSWORD '<password>';` 2. Откройте страницу ***Configuration → Clusters*** в UI ADCC и перейдите к редактированию соединения с кластером. 3. Установите пароль `<password>`, выбранный на шаге 1, в поле ***Password*** и нажмите ***Save***.

Существует потенциальный риск утечки памяти для специфических запросов, затрагиваемых `gpperfmon`. При выполнении запросов к большим таблицам с использованием функций UDF, генерирующих межкортежные запросы, можно столкнуться с повышенным потреблением RAM и даже ошибками OOM. Например, представление `information_schema.columns` может вызвать значительную утечку памяти при обращении к большому системному каталогу	4.0.0	4.1.0	Возможное решение заключается в отключении метрик мониторинга (операция требует перезапуска кластера):
			1. Удалите `gpadcc` из *shared_preload_libraries*, например:
			`$ gpconfig -c shared_preload_libraries -v ""`
			2. Отключите следующие параметры:
			`$ gpconfig -c gp_enable_gpperfmon -v off`
			`$ gpconfig -c adcc.explain_log_verbose -v off`
			`$ gpconfig -c adcc.explain_log_analyze -v off`
			3. Перезапустите кластер:
			`$ gpstop -a -M fast`
			`$ gpstart a`
			Имейте в виду, что это обходное решение не отключает ADCC UI и Arenadata DB Backup Manager

Сервер Backend ADCC не может подключиться к серверу ADCC UI. Периодически может не работать обновление данных на странице с детальной информацией	4.0.0	4.1.0	1. Подключитесь к серверу ADCC через SSH.
			2. Перейдите в директорию */opt/adcc*.
			3. Отредактируйте файл *docker-compose.yaml*. Для сервисов *adcc_ui_server* и *adcc_backend_server* измените значение переменной окружения `ADCC_EUREKA_CLIENT_PREFER_IP_ADDRESS` на `true`.
			4. Выполните следующую команду и дождитесь остановки всех Docker-контейнеров:
			`$ docker-compose down`
			5. Выполните команду:
			`$ docker-compose up -d`

Невозможно настроить задачу `Data cleanup` на странице ***Configuration***. ADCC способен работать без очистки данных, однако необходимо следить за размером базы данных с метриками	4.0.0	4.1.0	1. Подключитесь к серверу ADCC через SSH.
			2. Перейдите в директорию */opt/adcc*.
			3. Отредактируйте файл *docker-compose.yaml*. для сервиса *adcc_scheduler* добавьте дополнительную переменную окружения `ADCC_SCHEDULER_GRPC_PORT: 6572`.
			4. Выполните следующую команду и дождитесь остановки всех Docker-контейнеров:
			`$ docker-compose down`
			5. Выполните команду:
			`$ docker-compose up -d`

Релизы АДЛИН

Содержание

2.8.1
2.8.0
2.7.6
2.7.5
2.7.4
2.7.3
2.7.2
2.7.1
2.7.0
2.6.4
2.6.3
2.6.2
2.6.1
2.6.0
2.5.5
2.5.4
2.5.3
2.5.2
2.5.1
2.5.0
2.4.0
2.3.2
2.3.1
2.3.0
2.2.3
2.2.2
2.2.1
2.2.0
2.1.2
2.1.1
2.1.0
2.0.4
2.0.3
2.0.2
2.0.1
2.0.0
1.6.3
1.6.2
1.6.1
1.6.0
1.5.2
1.5.1
1.5.0
1.4.0
1.3.5
1.3.4
1.3.3
1.3.2
1.3.1
1.3.0
1.2.1
1.2.0
1.1.0
1.0.0

2.8.1

Дата выхода: 2026-03-20

Исправления багов

- Исправлена проблема с отображением таблицы «Дополнительные поля» при пустой

2.8.0

Дата выхода: 2026-03-20

Новая функция

- АПИ: метод «Создание» (POST /

Улучшения

- Расширение таблицы «Поля» в карточке клиента.

Исправления багов

- Исправлена ошибка при сохранении в карточке клиента.
- Исправлено отображение поля «Фамилия» в карточке клиента.
- Исправлено: «Сумма», «Скидка» и «Итого» в карточке заказа.

2.7.6

Дата выхода: 2026-03-19

Исправления багов

- Исправлено: «Сумма», «Скидка» и «Итого» в карточке заказа.

2.7.5

Дата выхода: 2026-03-19

Новая функция

- Переносимые шаблоны в Excel (.xlsx) и Word (.docx) в обе версии: экспорт

- Добавлена вкладка «Поля». В ней можно: 1. Отмечать обязательные поля и задавать

- Добавлена проверка на клиенте: при попытке сохранить карточку с пустым полем

Улучшения

- Переносимость шаблонов между версиями 1С:Фреш и 1С:Предприятие.

Исправления багов

- Исправлено: 500 «Ошибка при выполнении запроса» при сохранении карточки. Причина: в

- Исправлено: «Сумма», «Скидка» и «Итого» в карточке заказа.

2.7.4

Дата выхода: 2026-03-19

Исправления багов

- Исправлено: «Сумма», «Скидка» и «Итого» в карточке заказа.

2.7.3

Дата выхода: 2026-03-19

Исправления

- Стабильная: в карточке заказа при сумме, большей или равной

Исправления багов

- Исправлено: «Сумма», «Скидка» и «Итого» в карточке заказа.

Известные проблемы ADBM

Дарья Барышева

ADB.M

Назад к содержанию

Проблема	С версии	Fixed	Обходное решение
Агенты ADBM с версией ниже 2.8.0 (совместимые с ADB 6) не могут подключиться к серверу ADBM 2.8.0	2.8.0	—	Используйте отдельные установки ADBM для кластеров ADB 6 и ADB 7

Если хранилище NFS или CIFS, используемое для резервного копирования, становится недоступным, процесс `pgbackrest` может зависнуть и привести к блокировке операций ADBM. Например, выполнение бэкапа может оставаться в зависшем состоянии даже после истечения периода, заданного параметром **Protocol timeout** в конфигурации бэкапа	—	—	1. Определите идентификатор процесса `pgbackrest` и проверьте его состояние:
			`$ pgrep -a pgbackrest`
			`$ ps -o pid,state,command -p <PID>`
			Если параметр `state` имеет значение `D`, значит, процесс перешел в состояние `Uninterruptible Sleep` и не может быть завершен сигналом SIGKILL или по тайм-ауту `pgbackrest`, пока подключение к хранилищу не будет восстановлено.
			2. Смонтируйте хранилище NFS или CIFS с опцией `soft`, которая позволяет системным вызовам возвращать ошибки вместо бесконечного ожидания. Например, добавьте следующие содержимое в файл /etc/fstab:
			• Для NFS:
			`server:/share /mnt/share nfs soft,timeo=300,retrans=3 0 0`
			• Для CIFS:
			`//server/share /mnt/share cifs soft,timeo=300,retrans=3,credentials=/path/to/credentials 0 0`

Страница **Audit** не показывает операции	2.6.5	—	В настройках ADB Control → Up component parameters задайте значение `8890` для параметра **Listening port** для основного и внешних кластеров. Запустите действия **Reconfigure & Restart** для применения изменений
Кластеры, подключенные к внешнему ADB Control, должны быть обновлены вместе с основным кластером до ADB 6.29.0.3 для корректной работы ADBM и ADB Control	2.6.5	—	Обновите кластеры до ADB 6.29.0.3
Директория c WAL standby не утилизируется должным образом, если значение `wal_sender_archiving_status_interval` больше `0` в параметру `archive_mode` присвоено значение `always`, что в некоторых случаях приводит к ошибкам восстановления в ADBM	2.0.4	—	Присвойте параметру `wal_sender_archiving_status_interval` значение `0`
Создание бэкапной конфигурации в ADBM CLI завершается ошибкой, если SSl-логин использовался с дефолтными настройками `uploadChunkSize` и `uploadConcurrency`	2.0.4	—	Перед созданием конфигурации настройте параметры `uploadChunkSize` и `uploadConcurrency` в соответствии с ресурсами вашей системы. Плагин S3 пытается аллоцировать объем памяти, равный `segment_per_host` * `uploadChunkSize` * `uploadConcurrency`
Страница **Information** → **Backup manager** не отображает информацию для сегмент-хостов в статусе `DOWN`	1.0.0	—	Ожидаемое поведение, которое, возможно, будет изменено в следующих релизах
Прерывание потока (stream) завершается ошибкой, если при создании stream для опции **Restore mirrors** в кэшбэк значение `IO parallel`	2.2.0	2.4.0	При создании потоков с типом `Streaming` следует выбирать значение `After primary` для опции **Restore mirrors**
После обновления ADB до версии 6.26.2 запуск бэкапов с конфигурациями, созданными до обновления, завершается ошибкой	1.7.3	2.3.2	После обновления примените действие **Verify** к кластерам, имеющим бэкапные конфигурации

Восстановление данных завершается ошибкой, если один из сегмент-хостов кластера в статусе `down` — для точек восстановления, созданных после остановки хоста	1.7.0	2.0.4	1. Отредактируйте файл ssh_config под пользователем `root` на мастер-хосте:
			`$ vi /etc/ssh/ssh_config`
			2. Добавьте строку в конец файла:
			`ConnectTimeout 20`
			3. Перезапустите SSH daemon (`sshd`):
			`$ systemctl restart sshd`

Модальное окно **Create Stream** имеет следующее ограничение: тип восстановления `Streaming` является прототипом — его не следует использовать	1.3.1	1.7.3	Для потока (stream) допускается использовать только опцию **Copy**
Зеркала не восстанавливаются автоматически на целевом кластере. Если в целевом кластере уже есть зеркала — операция восстановления данных завершится ошибкой, если зеркал нет — восстановление выполнится без ошибок, однако целевой кластер не будет иметь зеркал даже если в исходном кластере они присутствовали	1.3.1	1.3.3	DR-кластер не должен иметь сегментов-зеркал (mirror)
Модальные окна **Create Stream** и **Restore** имеют следующее ограничение: опция **Restore mirrors** неприменима	1.3.1	1.3.3	Сбросьте флаг **Restore mirrors** при настройке потока (stream) или восстановления
Копирование данных вне времени Disaster recovery завершается с ошибкой, если: • В исходном кластере нет standby, а в целевом кластере есть • В исходном кластере есть standby, а в целевом кластере нет.	1.3.1	1.3.3	Standby должен присутствовать либо отсутствовать в обоих кластерах (исходном и целевом) одновременно
Список восстановлений БД пуст для кластеров, в именах которых присутствует пробел	1.1.0	1.3.3	Названия кластеров не должны содержать пробел
Восстановление standby недоступно для целевого кластера	1.3.1	1.3.2	DR-кластер не должен иметь резервного мастера (standby)
Невозможно восстановить данные из бэкапа в DR-кластере, если число узлов исходного кластера отлично от DR-кластера	1.3.1	1.3.2	DR-кластер и исходный кластер должны иметь одинаковую топологию
Кластер отображается со статусом `Down` в ADBM, если master размещается на том же хосте, что и сегмент	1.3.1	1.3.2	Отредактируйте /opt/adbm-agent/service-configuration и оставьте только одну запись, описывающую роли агента ADBM для текущего хоста: `ADBM_AGENT_ROLES=EXECUTOR,DISPATCHER`, затем перезапустите агент командой `systemctl restart adbm-agent`
DR-кластер не может быть использован для записи	1.3.1	1.3.2	Не применяйте команды DDL или DML в DR-кластере

DR-кластер не запускается после восстановления	1.3.1	1.3.2	1. Сделайте бэкап файла pg_hba.conf для сегментов `-1` и `0` в DR-кластере перед запуском stream. Сохраните его в долговременное хранилище (не используйте /tmp). Примените следующие команды (под пользователем `gpadmin`) на мастере:
			`$ cp /data1/master/gpseg-1/pg_hba.conf /home/gpadmin`
			`$ gpssh -v -f /home/gpadmin/arenadata_configs/arenadata_segment_hosts_hosts -m 'ls /data1/primary/gpseg*/pg_hba.conf /home/gpadmin'`
			2. Восстановите pg_hba.conf после завершения действия **Disaster recovery copy stream** с помощью следующих команд (под пользователем `gpadmin`) на мастере:
			`$ gpssh -v -f /home/gpadmin/pg_hba.conf /data1/master/gpseg-1/`
			`$ gpssh -v -f /home/gpadmin/arenadata_configs/arenadata_segment_hosts_hosts -m 'find /data1/primary/gpseg* -maxdepth 0 -exec cp /home/gpadmin/pg_hba.conf {} \;'`
			3. Измените параметр `synchronous_standby_names` на всех primary-сегментах с `'*'` на `''` в файле /<data_directory>/primary/gpseg*/postgresql.auto.conf.

Модальные окна **Create Stream** и **Restore** имеют следующее ограничение: выбор конкретных баз данных в поле **Database** приводит к ошибке	1.3.1	1.3.2	Выберите все базы данных при настройке потока (stream) или восстановления
Нет возможности добавить внешние ADB-кластеры в ADBM через ADBM UI	1.0.0	1.3.1	1. Убедитесь, что внешний ADB-кластер с установленным сервисом ADBM существует в ADCM. 2. Подключитесь к консоли Linux. Сгенерируйте кеш Basic Authorization base64 для пользователя с доступом к Arenadata DB Backup Manager с помощью следующей команды: `echo -n <adbm_user>:<adbm_user_password>	base64`, где `<adbm_user>` — логин пользователя, `<adbm_user_password>` — пароль. Например: `$ echo -n adbm_user:Password	base64` Результат: `YWRtbV91c2VyOlBhc3Rk3Q3k` 3. Отправьте следующий POST-запрос на хост ADBM общего назначения (general-purpose): • Заголовки: • `Accept` • `Accept-Charset` • `Authorization`. Параметр должен начинаться с `Basic`, а затем включать значение base64 с шага 2. • `Content-Type` • `X-ADBM_USER`. Параметр должен содержать значение `adbm_user` с шага 2. • Тело запроса: `{ "clusterName": "<указание внешнего кластера ADB в ADCM>", "clusterType": "<наименование кластера ADB>", "clusterVersion": "<версия кластера ADB>" }` Ниже приведен пример запроса POST с использованием команды Linux `curl`: `$ curl --request POST \ --url http://adbe-main:8976/backend/adbm/api/v1/clusters \ --header 'Accept: application/json' \ --header 'Accept-Charset: ISO-8859-1' \ --header 'Authorization: Basic YWRtbV91c2VyOlBhc3Rk3Q3k' \ --header 'Content-Type: application/json' \ --header 'X-ADBM-USER: adbm_user' \ --data '{ "clusterName": "adb-demo", "clusterType": "ADB demo stand", "clusterVersion": "6.22"}'` В случае успешного выполнения запрос будет содержать JSON со сгенерированным идентификатором внешнего кластера в ADBM. Кластер становится видимым на странице **Arenadata DB Backup Manager** → **Clusters** со статусом `Down` 4. В ADCM перейдите к внешнему кластеру ADB и выберите **Services** → **ADBM** → **Configuration**. 5. Установите флаг **Show advanced** и в нижней области формы измените значение сервисного конфигурационного параметра агента ADBM `ADBM_EUREKA_CLIENT_SERV_URL_DEF_ZONE`. Замените его значением URL сервисного реестра для хоста ADBM общего назначения, например **http://adbe-main:8761/eureka**. 6. Примените действия **Reconfigure & Restart** к ADBM. В случае успеха статус кластера сменится на **Arenadata DB Backup Manager** → **Clusters** изменится на `Up`.
Не поддерживаются FQDN в столбцах `hostname` и `address` таблицы `gp_segment_configuration`. Фактическое это блокирует применение ADBM в кластерах, использующих FQDN	1.0.0	1.1.0	Измените имена в таблице `gp_segment_configuration` на сокращенный вариант. Имейте в виду, что это чрезвычайно опасная операция, которая потенциально может нанести вред кластеру
Сегмент-серверы имеют некорректную дополнительную роль. ADBM будет работать, однако с аллоцацией запросами на сегмент-серверах. Пользователь может обнаружить это только по логу	1.0.0	1.1.0	Измените конфигурационный файл на странице **ADCM** → **ADBM** → **Configuration**: 1. Установите флаг **Show advanced**. 2. Найдите поле **ADBM agent service config parameters**. 3. Замените `master` на `adb.master` и `segment` на `adb.segment`. 4. Нажмите **Save** и запустите действия **Reconfigure & Restart**.