

Описание эксплуатации
программного обеспечения для электронно-
вычислительных машин
Arenadata Streaming Platform T4 (ADS T4)

Москва
2024

Содержание:

1.	Обзор.....	3
1.1	Компоненты.....	3
1.2	Ключевые особенности	6
1.3	Сценарии использования	7
1.3.1	Обмен сообщениями	7
1.3.2	Отслеживание активности на сайте.....	8
1.3.3	Метрики	9
1.3.4	Агрегация журналов	10
1.3.5	Потоковая обработка.....	10
1.3.6	Event sourcing	11
1.3.7	Commit log.....	12
2.	Управление кластером ADS T4	12
2.1	Действия в кластере.....	12
2.1.1	Управление кластером	14
2.1.2	Обновление кластера	16
2.2	Действия в сервисе	17
3.	Обзор Arenadata Platform Security	20
3.1	Интеграция с кластером ADS T4	21
3.1.1	Интеграция кластера ADPS с кластером ADS T4.....	21
3.1.2	Доступ к веб-интерфейсу Ranger	22

1. Обзор

Arenadata Streaming Platform T4 (ADS T4) – это платформа потоковой передачи данных в реальном времени, разработанная компанией Arenadata. ADS T4 предназначена для того, чтобы предприятия могли обрабатывать, анализировать и реагировать на большие объемы потоков данных в режиме реального времени.

Платформа использует Apache Kafka в качестве основной системы обмена сообщениями, известной своей высокой пропускной способностью и низкой задержкой. ADS T4 обеспечивает распределенную и отказоустойчивую архитектуру, которая может обрабатывать большие объемы данных из различных источников, включая базы данных, устройства IoT, датчики и другие источники потоковой передачи.

1.1 Компоненты

Apache ZooKeeper – это служба распределенной координации, используемая Arenadata Streaming для управления конфигурацией и координацией своих кластеров. Это важнейший компонент системы, поскольку он помогает обеспечить высокую доступность и отказоустойчивость в кластерах потоковой передачи Arenadata.

ZooKeeper предоставляет иерархическое пространство имен, которое позволяет Arenadata Streaming хранить данные конфигурации, управлять распределенными блокировками и координировать распределенные процессы. Он обеспечивает единообразное представление состояния системы на всех узлах кластера, что помогает предотвратить несогласованность данных и обеспечить их целостность.

Например, Arenadata Streaming использует ZooKeeper для управления брокерами Kafka, топиками и партициями. Когда в кластер добавляется новый брокер, ZooKeeper используется для присвоения ему уникального идентификатора и координации распределения данных по кластеру.

Apache Kafka – это распределенная платформа потоковой передачи, используемая Arenadata Streaming для управления приемом, обработкой и анализом потоков данных в реальном времени. Он предоставляет

масштабируемую, отказоустойчивую и высокодоступную инфраструктуру для обработки и хранения данных в реальном времени.

Arenadata Streaming использует возможности Kafka для обработки больших объемов данных и поддержки нескольких источников данных. Он предоставляет платформу обработки данных в режиме реального времени, которая позволяет предприятиям анализировать данные по мере их прохождения через систему, обеспечивая почти мгновенное понимание бизнес-операций.

Schema Registry – это централизованный репозиторий, используемый Arenadata Streaming для хранения и управления схемами данных, создаваемых и потребляемых Apache Kafka. Он позволяет пользователям определять, развивать и совместно использовать схемы в различных приложениях и системах, использующих Kafka.

В Arenadata Streaming сервис Schema Registry позволяет пользователям обеспечивать совместимость данных в различных версиях их приложений и систем. Он обеспечивает способ принудительной проверки данных и гарантирует, что все данные, создаваемые и потребляемые Kafka, соответствуют предопределенной схеме.

KSQL – это потоковый механизм SQL, используемый Arenadata Streaming для обработки потоков данных в реальном времени. Он позволяет пользователям писать SQL-запросы для преобразования, агрегирования и анализа данных в режиме реального времени, что упрощает создание конвейеров обработки данных в реальном времени без необходимости сложного программирования.

В Arenadata Streaming сервис KSQL предоставляет простой, но мощный способ взаимодействия с потоками данных, позволяя пользователям запрашивать, объединять и фильтровать данные по мере их прохождения через систему. Он поддерживает широкий спектр операций SQL, включая работу с окнами, агрегирование и объединение, что позволяет пользователям создавать сложную логику обработки без необходимости в специальном коде.

Kafka Connect – это платформа интеграции данных, используемая Arenadata Streaming для перемещения данных между Apache Kafka и другими системами. Она предоставляет масштабируемую и отказоустойчивую инфраструктуру для приема и экспорта данных в Kafka и из нее, что упрощает интеграцию различных систем и технологий с Kafka.

В Arenadata Streaming сервис Kafka Connect позволяет пользователям интегрировать с Kafka данные из различных источников, таких как базы данных, файловые системы и системы обмена сообщениями. Kafka Connect предоставляет коннекторы, которые можно настроить для извлечения данных из разных систем и записи их в топики Kafka или для чтения данных из топиков Kafka и записи их во внешние системы.

Kafka Connect также используется для MirrorMaker 2. MirrorMaker 2 – это инструмент, используемый Arenadata Streaming для репликации данных между кластерами Apache Kafka. Он заменяет оригинальный инструмент MirrorMaker и предоставляет несколько новых функций и улучшений по сравнению с его предшественником.

Kafka REST Proxy – это инструмент, используемый Arenadata Streaming для предоставления функциональности Apache Kafka в виде RESTful API. Он предоставляет простой и масштабируемый способ интеграции Kafka с другими системами и технологиями, поддерживающими RESTful API.

Apache NiFi – это инструмент интеграции данных с открытым исходным кодом, используемый Arenadata Streaming для автоматизации потока данных между различными системами и технологиями. Он предоставляет визуальный интерфейс drag-and-drop для проектирования и настройки потоков данных, что позволяет пользователям легко создавать сложные конвейеры данных без написания кода.

В Arenadata Streaming сервис Apache NiFi позволяет пользователям создавать и управлять потоками данных в различных системах и технологиях. Он предоставляет широкий спектр процессоров и коннекторов, которые можно

использовать для интеграции с различными источниками данных и местами назначения, включая базы данных, очереди сообщений и облачные платформы.

Apache MiNiFi – это облегченный инструмент сбора данных, используемый Arenadata Streaming для сбора и предварительной обработки данных на сетевой периферии. Он предназначен для работы на устройствах с ограниченными ресурсами, таких как датчики и устройства IoT, и позволяет пользователям собирать и обрабатывать данные в режиме реального времени, не полагаясь на центральный сервер.

В Arenadata Streaming сервис Apache MiNiFi позволяет пользователям собирать и предварительно обрабатывать данные на границе сети перед их отправкой на центральный сервер для дальнейшей обработки и анализа. Он предоставляет широкий спектр процессоров и разъемов, которые можно использовать для сбора данных из различных источников, включая датчики, камеры и другие устройства IoT.

1.2 Ключевые особенности

Быстрое развертывание – Arenadata Streaming упрощает процесс установки и настройки, сокращая время, необходимое для настройки по сравнению с ручными методами.

Удобство для пользователя – Благодаря Arenadata Streaming пользователи могут легко развертывать и настраивать свою инфраструктуру потоковой передачи данных, даже не обладая обширными техническими знаниями.

Согласованная установка – Arenadata Streaming обеспечивает стандартизированное развертывание в нескольких системах, сводя к минимуму ошибки и несоответствия.

Улучшенная производительность – Оптимизируя процесс настройки потоковой передачи данных, Arenadata Streaming повышает производительность системы, сводя к минимуму время простоя и повышая эффективность.

Экспертиза сообщества – Команда Arenadata оценивает исправленные баги и улучшения, полученные от сообщества Data Streaming, гарантируя их включение в продукт для бесперебойной работы.

1.3 Сценарии использования

1.3.1 Обмен сообщениями

Для оптимальной организации обмена сообщениями (messaging) используются брокеры сообщений — приложения, которые выступают посредниками между приложениями источника и приёмника, использующих разные протоколы. Kafka обладает несколькими преимуществами по сравнению с большинством брокеров сообщений:

- Высокая пропускная способность позволяет обработать большой объем данных за незначительный период времени.
- Организованное встроенное партиционирование (partitioning) позволяет контролировать время хранения и объем данных за счет того, что сообщения разделены на сегменты по партициям и не хранятся в одном большом файле.
- Репликация позволяет автоматически переключаться на реплики записей при отказе сервера в кластере, поэтому сообщения остаются доступными даже при сбоях.
- Отказоустойчивость гарантирует, что система обмена сообщениями будет сохранять сообщения при отсутствии активных подписчиков до тех пор, пока новый потребитель не подпишется на очередь сообщений.

Табл. 1. Сервисы, дополняющие и оптимизирующие работу Kafka

Сервис	Сценарий взаимодействия с Kafka	Преимущества использования
NiFi	NiFi как производитель (producer) — принимает данные из источников непосредственно в центральный экземпляр NiFi, который доставляет	<ul style="list-style-type: none"> • Передача данных в Kafka без специального ПО. Возможность визуально отслеживать и контролировать конвейер данных

	данные в соответствующий топик Kafka	
	NiFi как потребитель (consumer) — принимает данные из Kafka и передает их в другую систему	<ul style="list-style-type: none"> Передача данных из Kafka в некоторые файловые и иные внешние системы (HDFS, HBase) без специального ПО. Возможность визуально отслеживать и контролировать конвейер данных
	Динамический самонастраивающийся поток данных — объединение возможностей NiFi, Kafka и аналитических платформ: <ol style="list-style-type: none"> NiFi передает данные в Kafka. Данные становятся доступными для аналитических платформ. Результаты записываются обратно в другой топик Kafka. Данные поступают в NiFi	<ul style="list-style-type: none"> Возможность дальнейшего использования данных, полученных NiFi, без специального ПО. Возможность визуально отслеживать и контролировать конвейер данных
ZooKeeper	Сообщает брокерам Kafka о текущем состоянии кластера	Автоматическое обновление метаданных у клиента Kafka при подключении к любому брокеру
Confluent Schema Registry	Реализует механизм работы с разными схемами данных Kafka	<ul style="list-style-type: none"> Определение полей, необходимых для описания события. Документирование событий и значений каждого его поля в доступном виде. Предотвращение искажения данных. Работа с различными форматами данных от нескольких источников информации

1.3.2 Отслеживание активности на сайте

Kafka — отличный инструмент для отслеживания активности на веб-сайте (website activity tracking). Когда новый пользователь регистрируется на веб-сайте, его активность может отслеживаться следующим образом:

1. Пользователь нажимает кнопку (элемент UI) в интерфейсе веб-страницы.

2. Веб-приложение создает сообщение с метаданными данного элемента UI.
3. Сообщения с метаданными собираются в пакете данных и отправляются в Kafka, создавая при этом логи фиксации (commit log).
4. При следующем действии пользователя с этим элементом UI сообщение о действии добавляется в логи фиксации, а смещение сообщения в очереди увеличивается.

Далее из Kafka эти данные могут быть собраны для аналитики. Эти данные показывают использование веб-сайта в режиме реального времени.

1.3.3 Метрики

Kafka может использоваться для оперативного сбора метрик (metrics) различных приложений и операционных данных:

- технологические процессы;
- статистика аудита и сбора данных;
- активность системы;
- агрегирование статистики работающих приложений и инфраструктуры;
- трекеры потребления потоков данных пользователями в режиме реального времени.

Kafka обладает несколькими преимуществами при использовании в мониторинге:

- Возможность подключать новых производителей для отправки метрик, а затем использовать данные мониторинга для нескольких различных систем.
- Возможность выполнять анализ в реальном времени по большому набору данных, одновременно со сбором метрик.
- Возможность использовать небольшой объем кода для обработки данных в приложении потребителя.
- Возможность выделить в разные модули данные в соответствии с их назначением.

1.3.4 Агрегация журналов

Агрегация журналов (log aggregation) — один из возможных сценариев использования ADS. Для распределенных сред с разными архитектурами Kafka является оптимальным средством для агрегации журналов. Kafka собирает файлы журналов с серверов и помещает их в центральное хранилище (возможно, на файловый сервер или HDFS) для обработки. По сравнению с другими системами Kafka обладает рядом преимуществ:

- Kafka абстрагируется от деталей файлов и представляет абстракцию для данных журнала или событий в виде потока сообщений. Это обеспечивает обработку с меньшей задержкой и упрощает поддержку нескольких источников данных и распределенного потребления данных.
- Kafka предлагает столь же высокую производительность, как и другие системы, но более высокие гарантии отказоустойчивости и гораздо меньшую задержку в обработке данных.

1.3.5 Поточковая обработка

Для улучшения производительности обмена сообщениями используется потоковая обработка данных (stream processing). Преимущества потоковой передачи данных с использованием Kafka перед пакетными конвейерами:

- Потоки сообщений обрабатываются в реальном времени.
- К сообщениям применяются операции преобразования, фильтрации, агрегирования или соединения, чтобы опубликовать обработанные сообщения в другом потоке.
- Поточковые конвейеры снижают нагрузку на источник данных, т.к. не нужно выполнять полные запросы, а можно извлекать данные из лог-файлов СУБД или других систем.
- Сообщения могут храниться столько, сколько необходимо.

Табл. 2. Инструменты и сервисы для потоковой передачи данных

Инструмент или сервис	Роль в сценарии потоковой обработки	Преимущества использования
Kafka Streams	Клиентская библиотека для разработки потоковых приложений Big Data, которые работают с данными, хранящимися в топиках Kafka	Мощный и гибкий API-интерфейс со всеми преимуществами Kafka (масштабируемость, надежность, минимальная задержка, механизмы аналитических запросов) позволяет разработчику писать код в локальном режиме (вне кластера)
NiFi	Использует концепцию потока как последовательность операций: передача, преобразование и обогащение данных над последовательностью отдельных событий (events)	Поток НЕ рассматривается как большая пакетная (batch) операция, требующая выполнения первоначальной загрузки всех данных перед началом обработки. Например, база данных SQL с миллионами строк рассматривается как миллионы отдельных строк, требующих своей обработки
ksqlDB	Платформа для потоковой обработки больших данных в Kafka с помощью структурированных SQL-запросов	Структуры данных в ksqlDB — это программные единицы, которые способны хранить и обрабатывать множество данных, связанных логикой ksqlDB

1.3.6 Event sourcing

Kafka является оптимальным вариантом для поддержки приложений, созданных по шаблону event sourcing для обработки событий. Преимущества использования Kafka в event sourcing:

- Поддержка хранения данных журналов значительного объема.
- Возможность работы с отложенными событиями благодаря механизмам очереди и смещений в Kafka.

1.3.7 Commit log

Kafka может служить внешним commit log для распределенной системы.

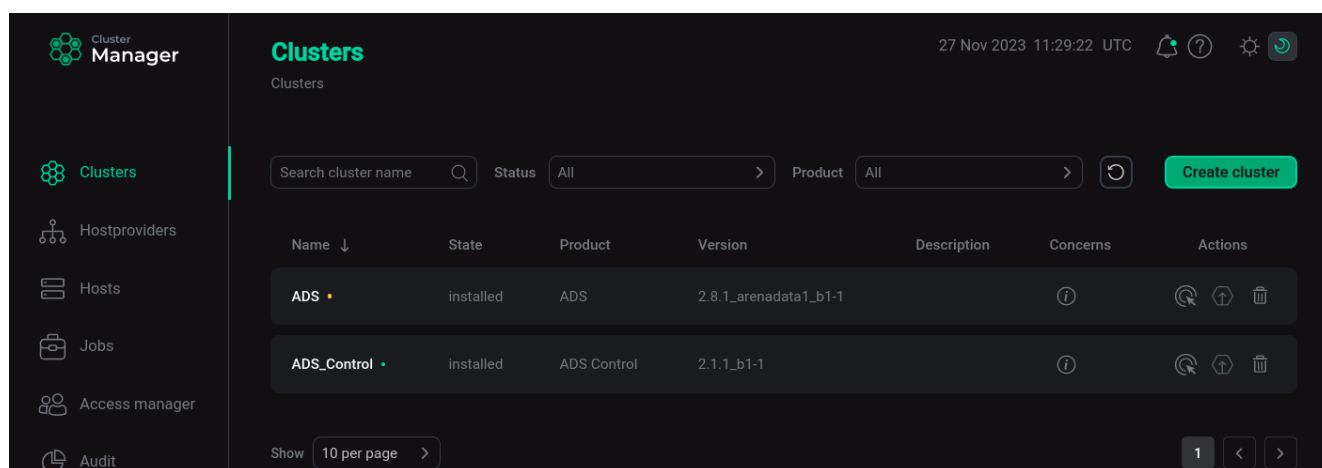
Преимущества использования Kafka:

- Возможность репликации данных между узлами.
- Возможность восстановления данных (механизм повторной синхронизации для отказавших узлов).
- Функция сжатия журналов в Kafka.

2. Управление кластером ADS T4




2.1 Действия в кластере

Действия с кластером выполняются на странице *Clusters*.

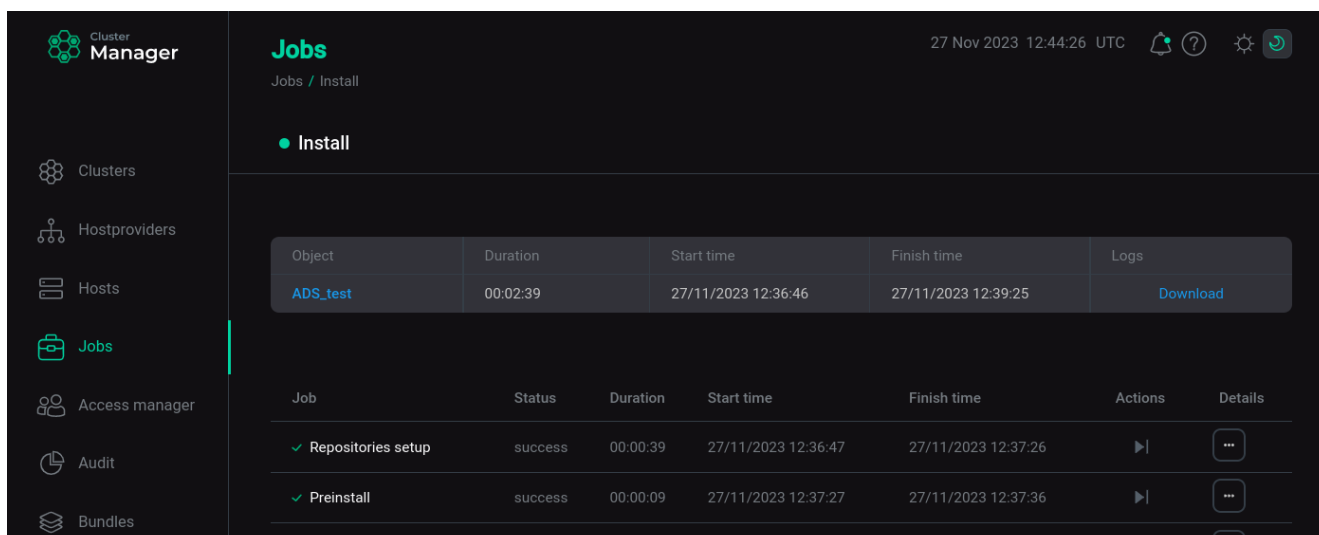


Страница *Clusters* содержит таблицу со следующими столбцами:

- *Name* — имя, указанное при создании кластера.
- *State* — текущее состояние кластера (`created`, `installed` или `upgrade from x.x.x`).
- *Product* — имя продукта.
- *Version* — версия продуктового бандла, который был использован для установки кластера.
- *Description* — описание, указанное при создании кластера.

- **Concerns.** Иконка *i* служит индикатором корректной настройки кластера. Если в конфигурации кластера обнаружены критичные ошибки, в столбце отображается иконка *i* красного цвета — при наведении на нее курсора показывается окно с описанием ошибки и ссылкой, по которой можно перейти, чтобы выполнить требуемое конфигурирование.
- **Actions.** В столбце показываются иконки для управления кластером:
 -  — открывает список действий для работы с кластером.
 -  — указывает, доступна ли новая версия бандла, и позволяет запустить обновление кластера на новую версию.
 -  — удаляет информацию о кластере из ADCM (не удаляет кластер и не производит никаких изменений на хостах, относящихся к кластеру).


Вы можете использовать иконки, описанные выше, для выполнения соответствующих действий. После того как действие запущено, ADCM отображает процесс его выполнения и результат на странице *Jobs*. С этой страницы можно перейти на страницу отдельной задачи (кликнув по имени задачи), чтобы увидеть внутренние этапы ее выполнения и проанализировать ошибки в случае их возникновения.

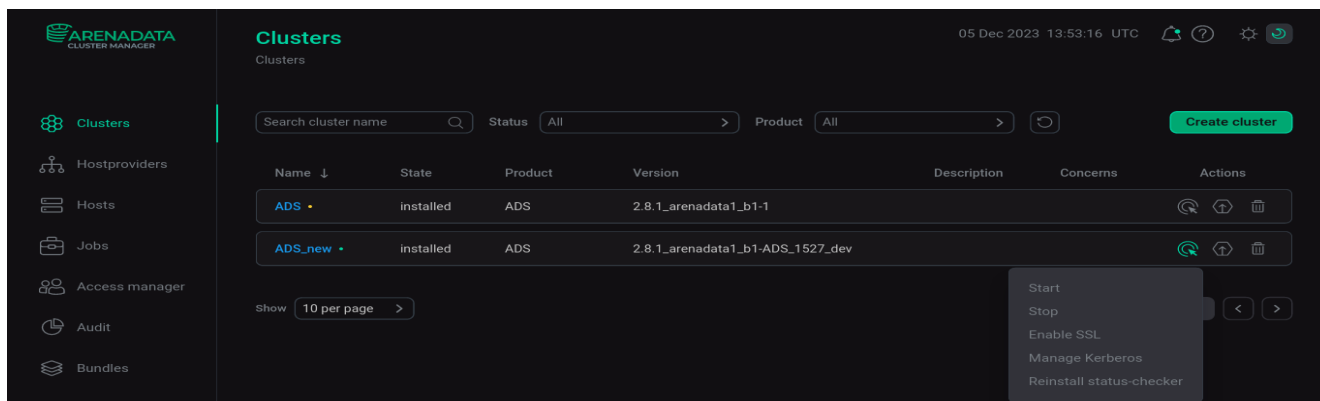


Object	Duration	Start time	Finish time	Logs
ADS_test	00:02:39	27/11/2023 12:36:46	27/11/2023 12:39:25	Download

Job	Status	Duration	Start time	Finish time	Actions	Details
✓ Repositories setup	success	00:00:39	27/11/2023 12:36:47	27/11/2023 12:37:26		
✓ Preinstall	success	00:00:09	27/11/2023 12:37:27	27/11/2023 12:37:36		

2.1.1 Управление кластером

Нажмите иконку , чтобы открыть список доступных действий, и выберите действие для выполнения.



Когда вы выбираете действие, ADCM отображает диалоговое окно для подтверждения выбора. В этом диалоговом окне можно установить флажок *Verbose*, чтобы просмотреть дополнительную информацию о выполнении действия на странице *Jobs*.

Набор доступных действий зависит от состояния кластера:

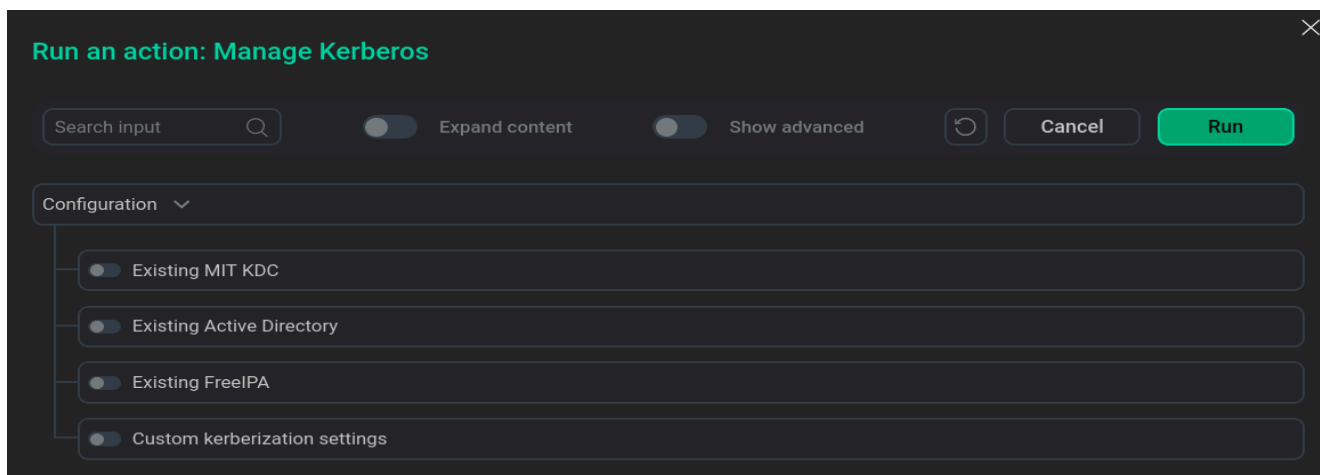
- Для нового кластера (состояние кластера `created`) доступно действие:
 - *Install* — устанавливает все сервисы кластера.
- После установки кластера (состояние кластера `installed`) доступны действия:
 - *Start* — осуществляет запуск сервисов кластера. Запуск сервисов осуществляется в той же последовательности, как при установке — сначала обязательные сервисы, затем дополнительные.
 - *Stop* — осуществляет остановку сервисов кластера. При остановке кластера в первую очередь останавливаются дополнительные сервисы, затем обязательные.
 - *Enable SSL* — осуществляет включение защиты канала по протоколу SSL.
 - *Disable SSL* — появляется после включения *Enable SSL*. Данное действие осуществляет выключение защиты канала по протоколу SSL.

- *Manage Kerberos* — позволяет включать, реконфигурировать и отключать аутентификацию по протоколу Kerberos.

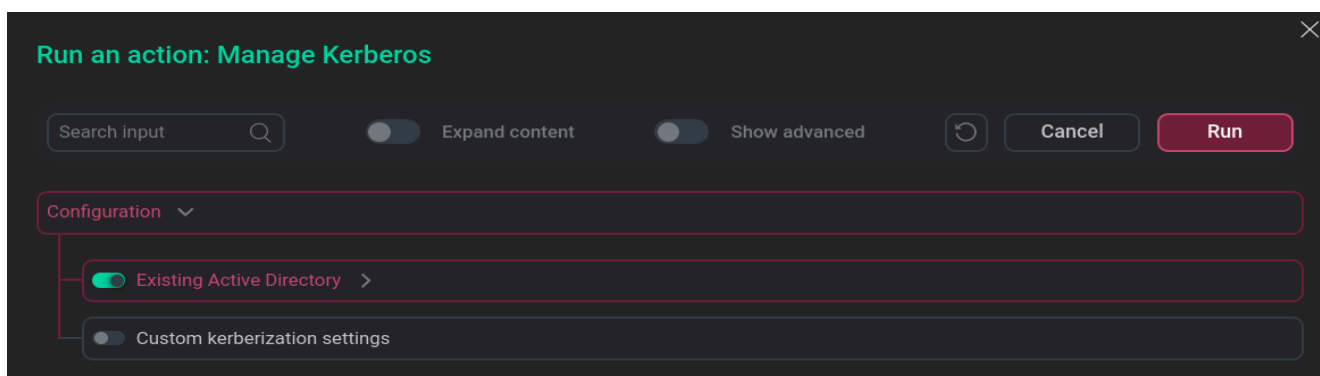
После выбора действия в появившемся окне будет предложено несколько опций для запуска действия:

- Existing MIT KDC
- Existing Active Directory
- Existing FreeIPA

Каждую из опций действия *Manage Kerberos* можно совместить с опцией *Custom kerberization settings*.



После включенной аутентификации по одному из протоколов после выбора действия *Manage Kerberos* в открывшемся окне будет предложено редактирование опции, по которой уже выполнено включение, и опция *Custom kerberization settings*.



- *Reinstall status-checker* — инициирует перенастройку и перезапуск statuschecker для всех сервисов кластера. Используется в случае миграции кластера под управление нового сервера ADCM.
- Для подготовленного к обновлению версии кластера (состояние кластера `upgrade from X.X.X`) доступно действие:
 - *Upgrade* — выполняет обновление версии кластера.



2.1.2 Обновление кластера

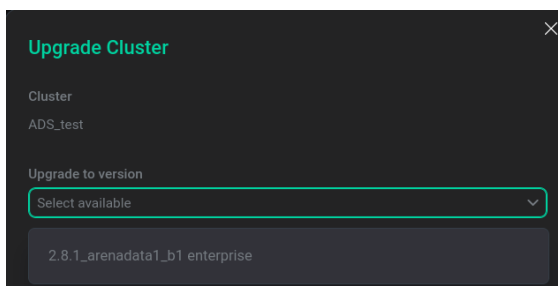
Перед обновлением важно убедиться, что конфигурации на хостах соответствуют конфигурациям ADCM. Если на хостах были ручные исправления, приведите их в согласованное состояние самостоятельно.

При обновлении кластера ADS порядок обновления продуктов должен быть следующим:

1. ADCM
2. ADPS
3. ADS

Для обновления версии кластера:

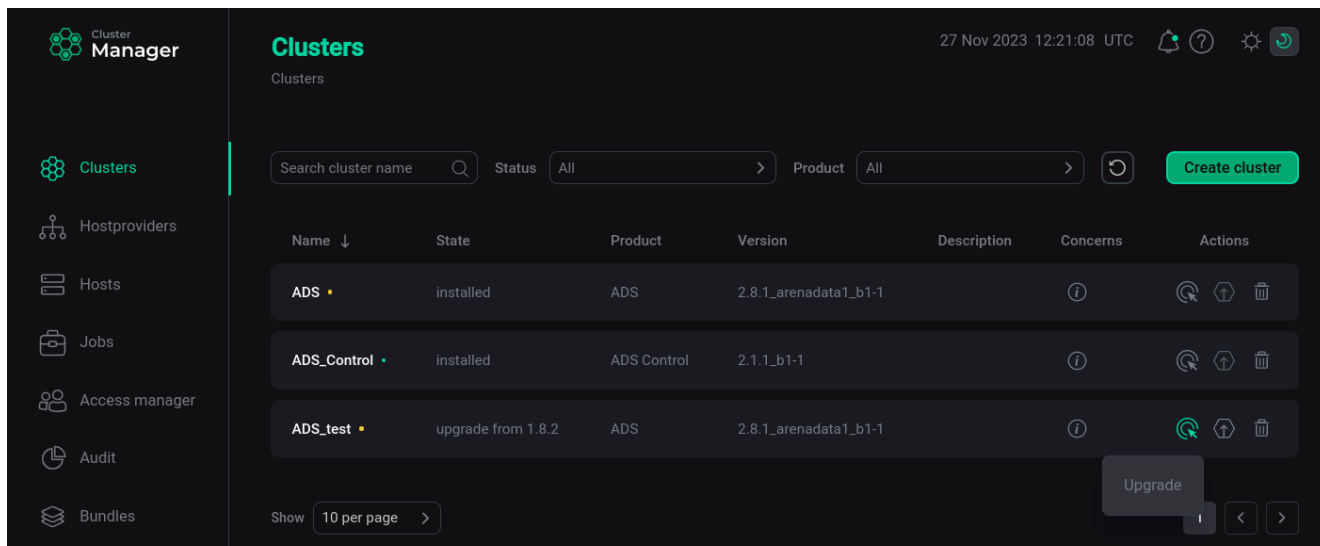
1. Загрузите бандл ADS новой версии.
2. Откройте страницу *Clusters*. После загрузки бандла иконка  становится активной.
3. Кликните на иконку , в открывшемся окне *Upgrade Cluster* выберите версию, доступную для обновления, и нажмите *Upgrade*.



4. В открывшемся диалоговом окне кликните *Run*.

В результате статус в столбце *State* изменяется на `upgrade from XXX` (в зависимости от версии кластера) — кластер готов к обновлению.

5. Кликните иконку  в столбце *Actions* и запустите действие *Upgrade*.



В открывшемся диалоговом окне нажмите *Run*, чтобы подтвердить выполнение обновления. Как только процесс обновления будет завершен, ADCM изменит состояние кластера на `installed`.

2.2 Действия в сервисе


Пользовательский интерфейс ADCM позволяет выполнять действия для управления сервисами кластера ADS. Для перехода к действиям нажмите на имя кластера на странице *Clusters* и откройте вкладку *Services*.

The screenshot displays the 'Services' tab within the 'Clusters / ADS / Services' path. The table lists the following services:

Name	Version	State	Concerns	Actions
Kafka	2.8.1	installed	<i>i</i>	[Refresh] [Add] [Delete]
Kafka Connect	2.8.1	installed	<i>i</i>	[Refresh] [Add] [Delete]
Kafka REST Proxy	6.2.1	installed	<i>i</i>	[Refresh] [Add] [Delete]
Kafka-Manager	3.0.0.5_arenadata5	installed	<i>i</i>	[Refresh] [Add] [Delete]
MINIFI	1.20.0_arenadata1	installed	<i>i</i>	[Refresh] [Add] [Delete]
NIFI	1.20.0_arenadata1	installed	<i>i</i>	[Refresh] [Add] [Delete]
Schema-Registry	6.2.1	installed	<i>i</i>	[Refresh] [Add] [Delete]
Zookeeper	3.5.10	installed	<i>i</i>	[Refresh] [Add] [Delete]
ksqldb	6.2.1	installed	<i>i</i>	[Refresh] [Add] [Delete]

Вкладка *Services* содержит таблицу со следующими столбцами:

- *Name* — имя сервиса.
- *Version* — версия сервиса.
- *State* — текущее состояние сервиса (например, `created` или `installed`).
- *Concerns*. Иконка *i* служит индикатором корректной настройки компонентов сервиса. Если в конфигурации сервиса обнаружены критичные ошибки (например, обязательные компоненты сервиса не установлены на хостах кластера), в столбце отображается иконка *i* красного цвета — при наведении на нее курсора показывается окно с описанием ошибки и ссылкой, по которой можно перейти, чтобы выполнить требуемое конфигурирование.

- *Actions*. При нажатии на иконку  открывается список действий для работы с сервисом.



После того как действие запущено, ADCM отображает процесс и результат его выполнения на странице *Jobs*. Также на странице *Jobs* можно проанализировать ошибки в случае их возникновения.

Набор действий зависит от типа сервиса.

Доступные действия для каждого сервиса описаны в статьях документации https://docs.arenadata.io/ru/ADStreaming/current/tutorials/cluster_management/service_actions.html:

- Действия в Kafka
- Действия в Kafka-Manager
- Действия в Kafka Connect
- Действия в Kafka REST Proxy
- Действия в ksqlDB
- Действия в NiFi
- Действия в MiNiFi
- Действия в Schema-Registry
- Действия в ZooKeeper
- Действия в Monitoring Clients

В столбце *Actions* также отдельно показываются иконки для следующих действий:

-  — перевод сервиса в режим обслуживания (maintenance mode). В данный момент эта функциональность не реализована в данном кластере.
-  — удаление сервиса из кластера.

При неудачном результате удаления сервиса на странице *Jobs* можно проанализировать причины ошибки. Некоторые сервисы взаимозависимы. Например, перед удалением некоторых сервисов необходимо удалить сервис Kafka.

3. Обзор Arenadata Platform Security

Arenadata Platform Security (ADPS) — это специализированный фреймворк для управления безопасностью, построенный на базе сервисов Apache Ranger и Apache Knox. ADPS может использоваться в ADS для обеспечения авторизации и управления доступом к сервисам на основе преднастроенных политик.

Перед установкой ADPS необходимо убедиться, что выполнены следующие требования:

- Кластер Arenadata Streaming версии 1.6.2 или выше установлен и настроен согласно руководству Установка Arenadata Streaming.
- Соблюдены все требования к установке ADPS:
 - Требования к оборудованию
 - Требования к сети
 - Программные требования
- Сервисы ADPS не должны использовать хосты кластера ADS. Рекомендуется установка ADPS на отдельной виртуальной машине или выделенном сервере.
- При использовании внешней СУБД для хранения метаданных Ranger (PostgreSQL/MySQL) она должна быть запущена и доступна. При установке Ranger создаются два новых пользователя (имена по умолчанию: rangeradmin и rangerlogger) и две новые базы данных (имена по умолчанию: ranger и ranger_audit).
- Необходимо настроить один из следующих типов аутентификации:
 - SASL PLAINTEXT
 - MIT Kerberos
 - MS Active Directory

После установки ADPS и его интеграции с кластером ADS можно перейти к работе с плагинами Ranger, доступными для сервисов ADS.

3.1 Интеграция с кластером ADS T4

После проверки аутентификации пользователя необходимо определить его права доступа. Права доступа пользователя к ресурсам определяются авторизацией.

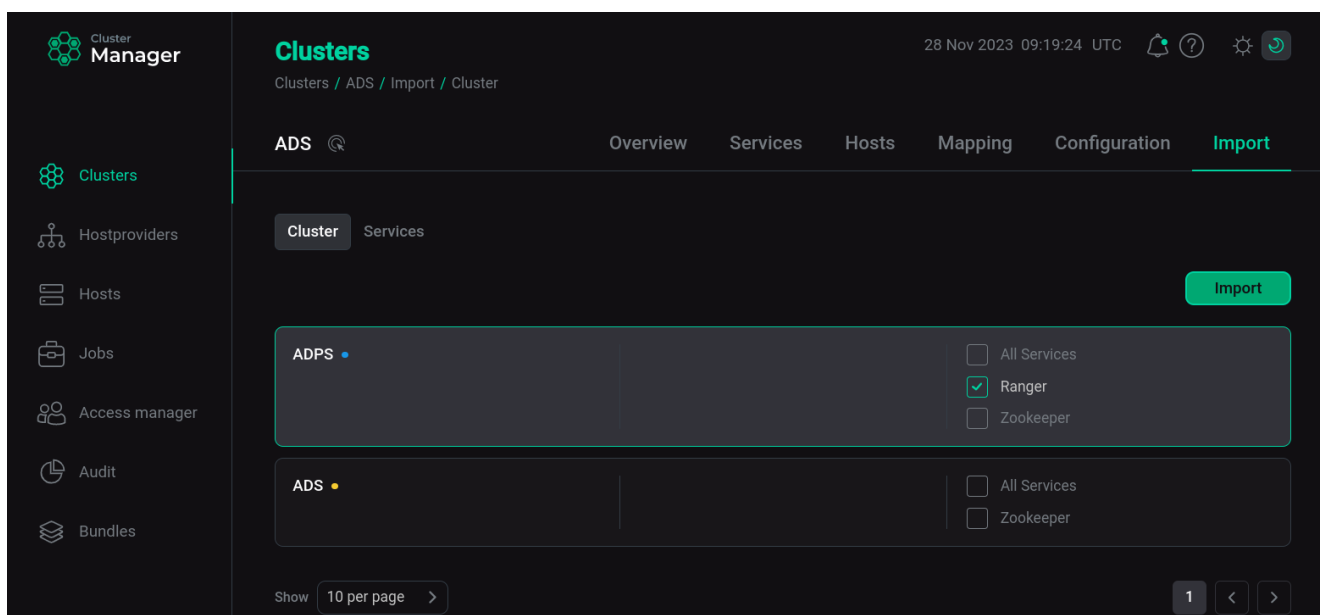
Возможности Ranger:

- Создание служб для сервиса Kafka ADS и добавление политики доступа пользователей к службам.
- Создание служб на основе тегов — позволяет управлять доступом к ресурсам нескольких компонентов без создания отдельных служб и политик в каждом компоненте.
- Использование Ranger Tab Sync для синхронизации хранилища тегов Ranger с внешней службой метаданных, например Apache Atlas.

3.1.1 Интеграция кластера ADPS с кластером ADS T4

Для включения интеграции кластера ADPS с кластером ADS T4:

1. Перейдите во вкладку *Import* кластера ADS.
2. Выберите сервис *Ranger* ранее добавленного кластера ADPS.
3. Кликните *Import*.



3.1.2 Доступ к веб-интерфейсу Ranger

Для доступа к веб-интерфейсу Ranger:

1. Введите в браузере обозначение хоста, на котором установлен ADPS, в качестве номера порта используйте значение, указанное в параметре *ranger.service.http.port* (по умолчанию 6080) сервиса Ranger.
2. Введите учетные данные администратора и кликните **Sign in** для авторизации в качестве администратора Ranger.

Пароли для доступа к веб-интерфейсу Ranger задаются перед установкой сервисов кластера ADPS в разделе *Credentials* конфигурации сервиса Ranger.

В результате открывается окно интерфейса *Service manager*.

