МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

РЕКОМЕНДОВАНО
к защите
Заведующий кафедрой
канд. экон. наук, доцент
_____ Е. В. Кийкова

КУРСОВАЯ РАБОТА
Проектирование системы управления метаданными в корпоративной информационной системе ООО «Farpost»
Б-ИС-22-01-84809.2560-а.2.000.КР

Студент
гр. БИС-22-01 _____ Л.А. Богданович

Руководитель
Старший
Преподаватель _____ О.Б. Богданова

Руководитель
Доцент _____ Б.К. Васильев

Владивосток 2026

Содержание

Введение

Современные предприятия в условиях цифровой трансформации сталкиваются с растущими объемами данных, генерируемых различными информационными системами. Одной из ключевых проблем в этой области является управление метаданными – данными о данных, которые обеспечивают контекст, происхождение и смысловую нагрузку основных данных. Отсутствие централизованной системы управления метаданными приводит к снижению прозрачности данных, усложнению процессов их анализа и контроля, а также повышению рисков нарушения требований регуляторов в области защиты информации.

Актуальность темы исследования обусловлена необходимостью создания единой системы управления метаданными в условиях распределенной ИТ-инфраструктуры предприятия, где данные поступают из множества источников в режиме реального времени. Особую сложность представляет интеграция метаданных из потоковых систем обработки данных, таких как Apache Kafka, Debezium и Altinity, а также из систем оркестрации рабочих процессов, как Apache Airflow. Решение данной проблемы позволит повысить качество управления данными, ускорить процесс принятия решений в области бизнес-аналитики и процессах разработки, а также обеспечить соответствие высоким стандартам Data Governance – системы процессов, политик, стандартов и инструментов, обеспечивающих управление данными на всех этапах их жизненного цикла. Централизованная система управления метаданными даст возможность бизнес-аналитикам и руководителям оперативно находить, понимать контекст и оценивать качество данных, сокращая время подготовки отчетов и аналитических материалов. Разработчики смогут быстрее принимать технические решения по оптимизации ETL-процессов и проектированию данных, имея полную информацию о структуре данных, их lineage, тоесть происхождения и взаимосвязях между датасетами.

Объектом внедрения выступает информационная система отдела обработки данных ООО «Фарпост», включающая компоненты для потоковой обработки данных и оркестрации задач. Предметом проектирования является архитектура интеграции системы управления метаданными в существующую ИТ-инфраструктуру.

Целью курсовой работы является проектирование внедрения системы управления метаданными с интеграцией данных из компонентов Debezium, Kafka, Altinity и Airflow в среде Kubernetes. Для достижения поставленной цели необходимо решить следующие задачи:

1) Провести анализ существующей ИТ-инфраструктуры информационной системы отдела обработки данных ООО «Фарпост» и выявить слабые места в текущем процессе управления метаданными;

2) Описать текущую практику работы с метаданными, а также сформулировать требования к новой системе управления метаданными;

3) спроектировать целевую модель процесса управления метаданными с учётом автоматизации и интеграции в существующую архитектуру;

4) выполнить сравнительный анализ современных систем управления метаданными по;

5) разработать логическую архитектуру интеграции OpenMetadata с компонентами Debezium, Kafka, Altinity и Airflow;

6) определить технологии и механизмы автоматического сбора метаданных для потоковых и оркестрируемых источников;

7) спроектировать структуру каталога метаданных;

8) описать интеграцию решения в существующую среду Kubernetes без дублирования уже развёрнутых сервисов;

9) разработать план валидации метаданных и оценить ожидаемые эффекты от внедрения.

При выполнении работы использовались следующие методы проектирования: анализ научной и технической литературы, методы системного проектирования, сравнительный анализ программных решений, а также методы моделирования архитектуры информационных систем.

Структура курсовой работы включает введение, три основные главы и заключение. Первая глава посвящена анализу существующей ИТ-инфраструктуры информационной системы отдела обработки данных ООО «Фарпост».

Вторая глава рассматривает текущие процессы управления метаданными: описывает их типы, источники, методы документирования, выявляет проблемы и формулирует требования к новой системе, а также проектирует обновленный процесс с учётом автоматизации.

Третья глава представляет проект внедрения системы: содержит общую концепцию интеграции, план реализации, сравнительный анализ систем управления метаданными, логическую архитектуру решения, проектирование ingestion-процессов, настройку структуры метаданных, интеграцию в Kubernetes-окружение.

В заключении подводятся итоги работы, подтверждается достижение цели и формулируются выводы о практической ценности решения.

1 Описание существующей ИТ-инфраструктуры информационной системы отдела

ИТ-инфраструктура отдела (рисунок 1) представляет собой сложную распределенную систему, обеспечивающую сбор, обработку, хранение и анализ данных из различных источников. В рамках данного раздела проведен анализ архитектуры информационной системы отдела, ответственного за обработку данных, с акцентом на компоненты, участвующие в потоковой передаче и преобразовании информации.

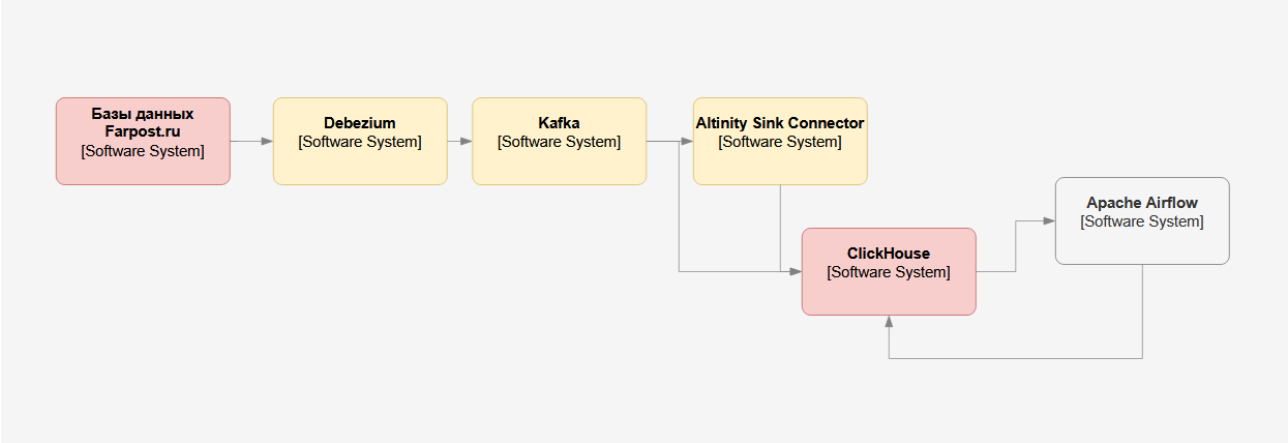

Рисунок 1 – Схема системы обработки данных в отделе

В настоящее время информационная система отдела обработки данных ООО «Фарпост» построена на современной архитектуре потоковой обработки данных. Центральным элементом системы является платформа Apache Kafka [1], выполняющая функцию централизованной шины данных для передачи событий между компонентами. Источниками данных для Kafka выступают реляционные базы данных MySQL предприятия, из которых изменения извлекаются с помощью инструмента Debezium [2], реализующего паттерн Change Data Capture (CDC). Данный подход обеспечивает отслеживание и передачу в реальном времени всех изменений, происходящих в исходных базах данных. События, поступающие в Kafka, имеют структурированный формат и содержат информацию о типе операции, затронутых таблицах и полях, а также временных метках изменений.

Для специализированных задач аналитики часть наиболее критичных таблиц MySQL реплицируется в аналитическую СУБД ClickHouse [3] с использованием Altinity Sink Connector [4]. Данный коннектор также основан на механизме CDC через Debezium и обеспечивает автоматическую передачу изменений в режиме реального времени с минимальной нагрузкой на производственные системы. Особую ценность представляет его способность автоматически обрабатывать изменения схемы данных и обеспечивать отказоустойчивость процесса репликации. Оркестрация всех процессов обработки данных осуществляется с помощью платформы Apache Airflow [5], где процессы преобразования и загрузки данных описываются в виде DAG. Основу

DAG-ов составляют SQL-скрипты, выполняющие агрегацию, фильтрацию и обогащение данных, поступающих из ClickHouse. Airflow обеспечивает планирование выполнения задач, мониторинг их состояния и управление зависимостями между различными этапами обработки данных.

Все компоненты инфраструктуры, включая Kafka, Debezium, Altinity и Airflow, развернуты в кластере Kubernetes. Данное решение обеспечивает отказоустойчивость, масштабируемость и эффективное использование вычислительных ресурсов. Kubernetes управляет развертыванием контейнеров, балансировкой нагрузки, автоматическим восстановлением сервисов при сбоях, а также обеспечивает изоляцию сред выполнения для различных компонентов системы.

Несмотря на наличие развитой инфраструктуры для обработки данных, в текущей архитектуре отсутствует централизованная система управления метаданными. В данный момент существует только некоторое количество страниц в Confluence о части агрегаций и репликаций, остальная информация хранится в сыром виде в системах обработки данных. Это создает комплекс проблем, влияющих на эффективность работы с данными. Во-первых, отсутствует единый каталог данных, содержащий полную информацию о структуре, назначении, владельцах и бизнес-контексте различных датасетов. Во-вторых, невозможно отслеживать полный путь данных (lineage) от исходных MySQL-баз через Kafka-топики и Altinity Sink Connector до конечных аналитических таблиц в ClickHouse и витринах данных. В-третьих, метаданные о SQL-скриптах, выполняющих запросы к базам данных, и DAG-ах – скриптах, которые автоматически ежедневно исполняют набор агрегаций данных, в Apache Airflow плохо поддерживаются, а именно редко обновляются и хранят не полные данные о себе, также некоторые агрегации вообще не имеют описания, что затрудняет поиск, понимание зависимостей и повторное использование кода. В-четвертых, отсутствие бизнес-глоссария и системы тегов не позволяет устанавливать связи между техническими метаданными и бизнес-терминами, что усложняет коммуникацию между аналитиками, разработчиками и бизнес-пользователями. В-пятых, отсутствие автоматизированного сбора и актуализации метаданных приводит к их быстрому устареванию и снижению доверия к данным.

Таким образом, существующая ИТ-инфраструктура отдела предоставляет прочную основу для внедрения системы управления метаданными, однако требует решения проблемы централизованного учета и каталогизации данных для повышения эффективности работы с информацией, особенно в условиях использования гибридной архитектуры с транзакционными и аналитическими базами данных.

2 Анализ процесса управления метаданными в отделе

2.1 Описание текущего процесса управления метаданными и характеристика метаданных

В настоящее время в отделе обработки данных ООО «Фарпост» отсутствует формализованный процесс управления метаданными. Управление осуществляется не в организованном порядке силами отдельных разработчиков и аналитиков, что приводит к фрагментации информации, устаревании сведений и отсутствию единого подхода. Процесс документирования метаданных носит преимущественно ручной характер и не интегрирован в основные рабочие процессы отдела (Рисунок 2).

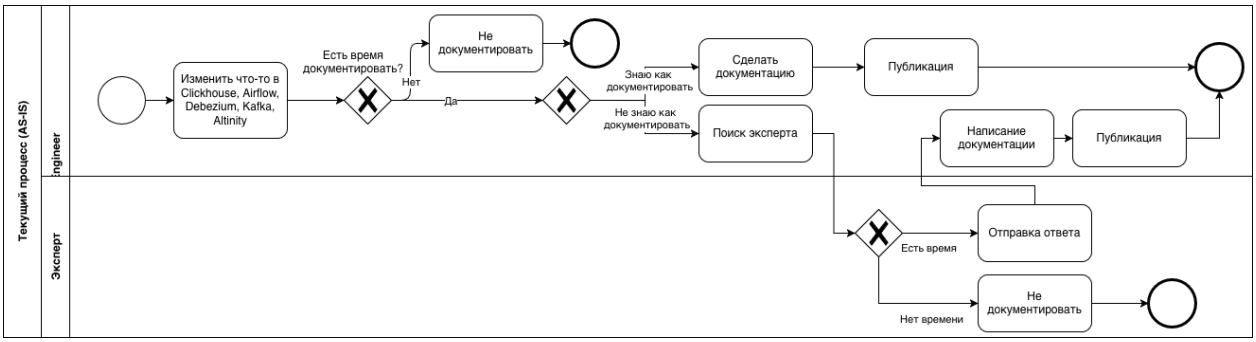

Рисунок 2 – Нынешний процесс управления метаданными

В данный момент процесс выглядит таким образом, вся документация и хранение метаданных ведется исключительно, если присутствует время, а также если есть знание, что и как документировать и использовать. На данный момент не существует регламента документации и использования технических метаданных, которые появляются во время выполнения задач. В основном задокументирован самый нужный минимум, все остальные метаданные в большинстве своем не используются. Кроме того, из-за отсутствия регламента, задокументированные метаданные могут быстро устаревать, дублироваться и быть неполной и неверной.

Метаданные в существующей архитектуре можно классифицировать по следующим категориям. Технические метаданные включают информацию о структуре баз данных таких как: названия таблиц, типы полей, индексы, параметрах Kafka-топиков, конфигурации Altinity Sink и параметрах Airflow DAG-ов. Данные метаданные генерируются автоматически системами, но не централизованы и не имеют единого формата представления.

Операционные метаданные содержат информацию о процессах обработки данных: время выполнения задач в Airflow, объемы обрабатываемых данных, статистику по репликации через Debezium и Altinity Sink Connector, метрики работы и данные топиков Kafka. Эти данные доступны через мониторинговые системы. Данные по Airflow есть только в самой системе.

Бизнес-метаданные представлены в минимальном объеме. Отдельные бизнес-правила и описания полей фиксируются в виде комментариев в SQL-скриптах или в Confluence-страницах,

но отсутствует систематизированный бизнес-глоссарий, сопоставляющий технические термины с бизнес-концепциями.

Особую проблему представляют метаданные происхождения данных. В текущей архитектуре невозможно автоматически отследить полный путь данных: от исходной MySQL-таблицы, через Debezium-коннектор, Kafka-топик и Altinity Sink Connector, до конечной таблицы в ClickHouse и последующих преобразований в Airflow DAG-ах. Эта информация существует лишь в головах разработчиков или в виде устаревших диаграмм в Confluence. Отсутствие актуального lineage, то есть происхождения данных, затрудняет анализ влияния изменений, выявление узких мест в обработке данных и расследование инцидентов, связанных с качеством данных.

Кроме того, в системе отсутствуют метаданные качества данных. Нет централизованного хранения информации о правилах валидации, метриках полноты и корректности данных на различных этапах обработки. Проверки качества реализуются в отдельных DAG-ах Airflow, но результаты не агрегируются и не коррелируются с другими метаданными.

Процесс обновления метаданных не автоматизирован. При изменении схемы базы данных разработчик обязан вручную обновить документацию в Confluence и комментарии в коде, что часто приводит к расхождению между реальным состоянием системы и документацией.

Таким образом, текущий процесс управления метаданными характеризуется фрагментацией, неавтоматизированностью и отсутствием единой системы классификации. Метаданные существуют в изолированных кусках и системах без взаимосвязей между ними, что не позволяет реализовать сквозные сценарии управления данными и обеспечить доверие к информации на всех этапах ее жизненного цикла.

2.2 Требования к новой системе управления метаданными

На основе проведенного анализа существующей инфраструктуры и выявленных проблем сформулированы функциональные и нефункциональные требования к проектируемой системе управления метаданными. Требования определены с учетом специфики архитектуры отдела данных ООО «Фарпост», включающей компоненты для потоковой обработки данных и оркестрации задач.

Ключевым функциональным требованием является автоматический сбор метаданных из всех модулей системы обработки данных. Модулями, из которых требуется извлекать метаданные, являются: Clickhouse, Apache Airflow, Debezium и Kafka, а также Altinity Sink Connector. Метаданные будут собираться с помощью коннекторов, а также в виде отдельных созданных под отдел модулей. В зависимости от источника, метаданные будут представлены в своем формате.

Критически важным требованием является отображение полного происхождения и связей между данными от исходных MySQL-баз через Kafka и Altinity Sink Connector до конечных таблиц в ClickHouse и результатов Airflow-задач. Система должна визуализировать потоки данных, включая преобразования на каждом этапе, с возможностью просмотра уровня отдельных полей и операций в виде графа зависимостей.

Система должна поддерживать создание и управление бизнес-глоссарием с возможностью связывания технических метаданных с бизнес-терминами. Требуется реализовать гибкую систему пользовательских тегов для классификации данных по различным критериям: конфиденциальность, владельцы, домены данных.

Из нефункциональных требований приоритетными являются: интеграция с существующим кластером Kubernetes без необходимости изменения текущей архитектуры и дублирования компонентов, обеспечение отказоустойчивости решения с возможностью восстановления после сбоев, масштабируемость, производительность сбора и обновления метаданных в режиме, близком к реальному времени.

Данные требования полностью соответствуют возможностям современных open-source платформ управления метаданными и позволят создать единую систему, обеспечивающую прозрачность данных, упрощение их поиска и понимания, а также соответствие принципам Data Governance – системы процессов, политик, стандартов и инструментов, обеспечивающих управление данными на всех этапах их жизненного цикла.

На основе проведенного анализа существующей инфраструктуры и выявленных проблем сформулированы следующие требования к проектируемой системе управления метаданными:

Функциональные требования:

1) Автоматический сбор метаданных из Kafka – извлечение информации о схемах таблиц, типах операций и временных метках из событий Debezium.

2) Интеграция с Altinity Sink Connector – сбор данных о реплицируемых MySQL-таблицах, их отображении в ClickHouse и параметрах репликации.

3) Каталогизация данных в ClickHouse – автоматическое обнаружение движков таблиц, схем, партиционирования и индексов с мониторингом изменений.

4) Извлечение метаданных из Airflow – анализ SQL-скриптов и DAG-файлов с определением зависимостей между задачами и источниками данных.

5) Визуализация data lineage – отображение полного пути данных от MySQL-источников через Kafka и Altinity Sink до ClickHouse и Airflow-результатов.

6) Бизнес-глоссарий и теги – связывание технических метаданных с бизнес-терминами и классификация данных по критериям конфиденциальности, критичности и владельцам.

Нефункциональные требования:

1) Интеграция с Kubernetes – развертывание в существующем кластере без изменения текущей архитектуры и дублирования компонентов.

2) Отказоустойчивость и масштабируемость – обеспечение непрерывной работы и поддержка обработки метаданных от тысяч таблиц.

2.3 Описание процесса управления метаданными с учетом изменений

После внедрения новой системы управления метаданными, процесс документирования любых данных изменился, что можно увидеть на рисунке 3.

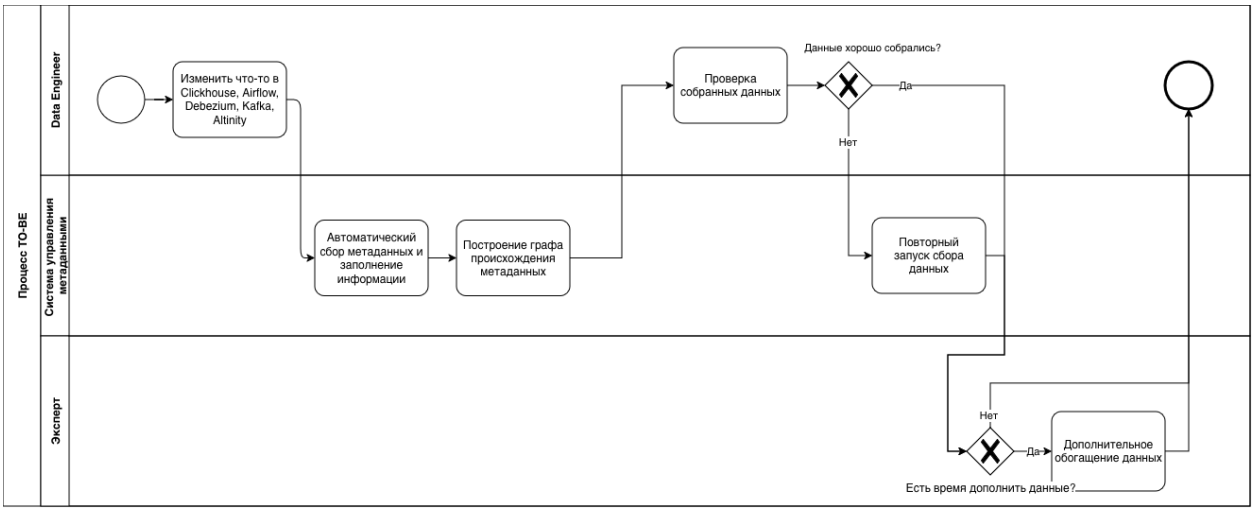

Рисунок 3 – Процесс управления метаданными TO-BE

Теперь у процесса документирования, или, другими словами, управления метаданными теперь такие шаги:

1) Автоматический сбор метаданных, куда входит сбор данных из Kafka-топиков, которые передают события в Debezium, сбор данных из Altinity Sink Connector, Apache Airflow и Clickhouse;

2) Структурирование и обогащение данных. Система управления метаданными автоматически определяет цепочки обработки данных, и строит граф, на котором видно, откуда и куда данные поступают, и какие таблицы от чего зависят. Также происходит привязка технических данных к бизнес-глоссарию, и назначаются владельцы и теги.;

3) Проверка загруженных данных ответственными людьми и заполнение бизнес глоссария.

Главное улучшение процесса документирования, которое дает внедрение системы управления метаданными, состоит в том, что рутинные задачи, по типу переноса технических данных в документацию автоматизируется, остается проверить, что все хорошо перенеслось, и дополнить бизнес-знаниями. Кроме того, система управления метаданными добавляет новые функции, к примеру, предоставление статистики о качестве данных и настройка оповещений об ошибках во время обработки данных.

3 Разработка проекта внедрения системы управления метаданными в корпоративную информационную систему

3.1 Общая концепция интеграции

Основной целью концепции внедрения новой системы управления метаданными является интеграция с основной системой отдела, с минимальным изменением текущих процессов и модулей обработки данных. Внедрение будет основываться на развертывании необходимых для работы системы управления метаданными модулей в Kubernetes кластере отдела. Также необходимо настроить модули так, чтобы они были отказоустойчивы и масштабируемы, для того чтобы предусмотреть возможное увеличение нагрузки и потока дынных.

Ключевыми точками интеграции с существующей системой являются:

1) Интеграция с Kafka: сбор метаданных топиков и схем сообщений;

2) Интеграция с Altinity Sink Connector: получение информации о репликации и маппинге таблиц;

3) Интеграция с Apache Airflow: извлечение метаданных из DAG-файлов и SQL-скриптов;

4) Интеграция с ClickHouse: автоматическое обнаружение схем и структуры таблиц.

Система будет построена на принципе «наблюдателя», где модули сбора метаданных пассивно потребляют информацию из существующих потоков данных без вмешательства в основные процессы обработки. Это обеспечит:

1) Отсутствие дополнительной нагрузки на источники данных за счет использования read-only доступов;

2) Изоляцию процессов сбора метаданных от критически важных ETL-процессов;

3) Возможность отключения модулей сбора без влияния на основную бизнес-логику.

Также одним из важных моментов концепции является централизованная обработка всех собранных метаданных непосредственно внутри модулей системы управления данными. Вся информация о структурах таблиц, схемах сообщений Kafka, параметрах репликации Altinity Sink Connector, DAG-файлах Airflow и конфигурациях ClickHouse будет автоматически агрегироваться, нормализоваться и обогащаться в рамках единого программного комплекса. Это обеспечит единообразие представления метаданных и исключит необходимость ручной синхронизации информации между различными источниками.

Для обеспечения максимальной доступности и удобства работы с данными будет использован веб-интерфейс (рисунок 4), предоставляемый системой управления метаданными, обеспечивающий пользователям интуитивно понятный доступ ко всей собранной статистике и метаданным.

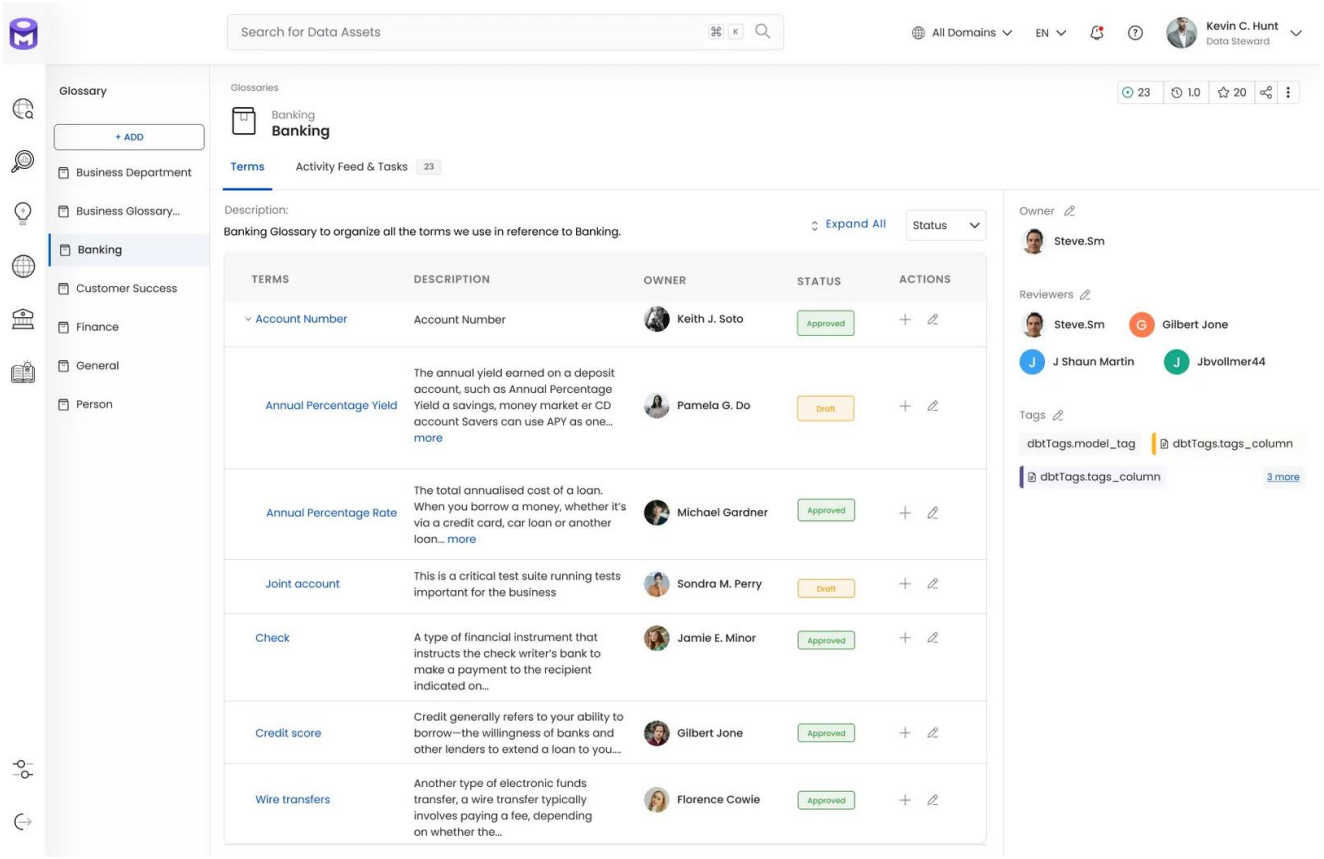

Рисунок 4 – пример интерфейса системы управления метаданными

Через этот интерфейс сотрудники отдела смогут визуализировать полное происхождение данных, просматривать актуальные схемы таблиц и сообщений, отслеживать зависимости между задачами Airflow, а также взаимодействовать с бизнес-глоссарием для связывания технических объектов с бизнес-терминами.

3.2 Оценка готовности инфраструктуры к внедрению системы управления метаданными

Оценка готовности инфраструктуры к внедрению системы управления метаданными была выполнена путем анализа существующих источников технических и бизнес-метаданных, а также степени их формализации, актуальности и связности между собой.

Наиболее значимые пробелы выявлены на уровне управления данными и метаданными, который на таблице 1, отмечен как «по большей части отсутствующий». В текущей архитектуре отсутствует единый каталог данных, обеспечивающий централизованный доступ к информации о наборах данных, их владельцах и назначении. Глоссарий бизнес-терминов присутствует лишь частично и не связан напрямую с техническими объектами системы. Граф происхождения данных (data lineage) отсутствует, что не позволяет наглядно проследить путь данных от источников до аналитических витрин и отчетов.

Таблица 1 – Анализ уровня управления данными

Объект	Статус
Единый каталог данных	Отсутствует
Глоссарий Бизнес-терминов	Частично присутствует
График происхождения данных	Отсутствует
Владельцы и теги	Частично присутствует
Итог:	**По большей части отсутствуют данные, не готово**

В то же время анализ уровня данных и оркестрации, также отраженный в таблице 2, показал наличие частичной готовности. Для Kafka, Debezium и Altinity Sink Connector в системе уже присутствуют технические метаданные, включая конфигурации коннекторов, схемы сообщений и параметры репликации. Однако данные метаданные распределены между различными компонентами инфраструктуры и не агрегированы в едином каталоге. Отсутствие централизованного представления потоков данных затрудняет анализ их взаимосвязей и оценку влияния изменений в источниках на потребителей.

Таблица 2 – Анализ уровня данных и орекстрации

Объект	Статус
Debezium	Есть метаданные
Altinity Sink Connector	Есть метаданные
Kafka	Есть метаданные
Clickhouse	Есть документация, которую можно доработать
Apache Airflow	Есть документация, которую можно доработать
Итог:	**Частично готов**

Как показано в таблице 3, на инфраструктурном уровне существенных пробелов выявлено не было. Использование Kubernetes в качестве базовой платформы обеспечивает стандартизированный и наблюдаемый контур для развертывания сервисов, а наличие настроенных механизмов мониторинга создает технические предпосылки для внедрения дополнительных сервисов. Данный уровень характеризуется статусом «готов» и не является ограничивающим фактором для дальнейшего развития архитектуры.

Таблица 3 – Инфраструктурный уровень

Объект	Статус
Kubernetes	Готов
Storage, Network, RBAC	Готов
Мониторинг	Готов
Итог:	**Готов**

Для ClickHouse и Apache Airflow, как видно из схемы оценки готовности, существует базовая документация, однако она требует дальнейшей доработки и стандартизации.

Механизмы назначения владельцев данных и использования тегов реализованы фрагментарно и не поддерживаются на системном уровне, что также отражено в оценочных таблицах. В результате ответственность за данные зачастую определяется неявно, а поиск ответственных лиц при возникновении инцидентов или изменении требований требует дополнительных временных затрат.

Таким образом, анализ, обобщенный в таблицах 1-3, показывает, что ключевые пробелы связаны не с техническими ограничениями инфраструктуры, а с отсутствием централизованного слоя управления метаданными. Существующие источники метаданных и документации формируют прочную основу для внедрения специализированной системы управления метаданными, однако без такого решения данные остаются слабо связаны между собой и недостаточно прозрачны для аналитиков и бизнес-пользователей.

3.3 Сравнение систем управления метаданными

Выявленный контраст между высокой готовностью ИТ-инфраструктуры и критическим отсутствием инструментов для управления данными подтверждает необходимость внедрения специализированной программной платформы. Чтобы устранить разрыв между техническим слоем и бизнес-аналитикой, требуется выбрать решение, способное консолидировать разрозненные метаданные и автоматизировать построение цепочек их происхождения. С целью поиска оптимального инструмента, соответствующего технологическому стеку ООО «ФарПост», был проведен сравнительный анализ наиболее востребованных систем с открытым исходным кодом.

Для сравнения были выбраны четыре наиболее популярных open-source решения для управления метаданными: OpenMetadata, Apache Atlas, DataHub и Amundsen.

OpenMetadata [6] – это современная платформа управления данными, построенная на принципах открытых стандартов и API-first подхода. Ключевой особенностью системы является использование единой стандартизированной схемы метаданных, что обеспечивает централизованное хранение информации и упрощает интеграцию с внешними инструментами. Платформа предоставляет широкий набор встроенных коннекторов для современных источников данных (включая ClickHouse, Kafka, Airflow) и обладает развитым функционалом для совместной работы, позволяя пользователям оставлять комментарии, оценивать качество данных и управлять тегами через интуитивно понятный веб-интерфейс.

Apache Atlas [7] – это зрелое решение корпоративного уровня, исторически разработанное для управления метаданными и Data Governance в экосистеме Hadoop. Платформа обеспечивает

глубокие возможности по классификации данных, управлению политиками доступа и аудиту изменений. Однако архитектура Atlas жестко привязана к компонентам стека Big Data, что делает её внедрение и поддержку в микросервисных средах Kubernetes, не использующих Hadoop, избыточно сложной и ресурсоемкой задачей.

DataHub [8] – масштабируемая платформа третьего поколения, первоначально разработанная компанией LinkedIn для решения задач поиска и обнаружения данных. Архитектурно DataHub базируется на потоковой модели передачи метаданных, что позволяет обрабатывать изменения в инфраструктуре в режиме, близком к реальному времени. Решение отличается мощными механизмами полнотекстового поиска, детальной визуализацией происхождения данных и модульной структурой, однако требует значительных ресурсов для развертывания и поддержки полного стека компонентов.

Amundsen [9] – проект от компании Lyft, основной фокус которого смещен на улучшение пользовательского опыта аналитиков и упрощение поиска данных. Система функционирует как поисковый движок, индексирующий метаданные для быстрого нахождения таблиц, дашбордов и владельцев данных.

Сравнение проводилось по следующим критериям, сформированным на основе требований ООО «Фарпост»:

Функциональные критерии:

1) Поддержка автоматического сбора метаданных из Kafka и Debezium;

2) Наличие интеграции с Altinity Sink Connector/ClickHouse;

3) Возможности автоматической каталогизации данных в ClickHouse (движки таблиц, схемы, партиционирование);

4) Поддержка извлечения метаданных из Apache Airflow;

5) Качество визуализации data lineage;

6) Функционал бизнес-глоссария и гибкой системы тегов.

Нефункциональные критерии:

1) Простота интеграции с существующим Kubernetes-кластером;

2) Архитектурная отказоустойчивость и горизонтальная масштабируемость;

3) Активность сообщества и качество документации;

4) Готовность к production-использованию.

Результаты сравнения представлены в таблице 4.

Таблица 4 – сравнение систем управления метаданными.

Критерии	**OpenMetadata**	**Apache Atlas**	**DataHub**	**Amundsen**
Сбор метаданных из Kafka/Debezium	Есть коннекторы	Требует собственной разработки	Частичная поддержка	Отсутствует
Интеграция с ClickHouse	Есть коннектор	Экспериментальная поддержка	Отсутствует	Требует доработки
Интеграция с Airflow	Глубокая интеграция	Требует разработки	Частичная поддержка	Нативная интеграция
Data Lineage визуализация	Отличная визуализация	Базовая визуализация	Отличная визуализация	Хорошая визуализация
Бизнес-глоссарий и теги	Гибкая система	Основные функции	Продвинутые возможности	Ограниченная функциональность
Масштабируемость (тысячи таблиц)	Распределенная архитектура	Хорошая масштабируемость	Отличная масштабируемость	Ограниченная
Активность сообщества	Высокая	Средняя	Высокая	Средняя
Production readiness	Готов к production	Требует доработки	Готов к production	Требует доработки
Интеграция с Kubernetes	Helm-чарты доступны	Требует настройки	Helm-чарты доступны	Требует доработки

После первичного анализа критериев по каждому из инструментов, можно увидеть, что больше всего подходят OpenMetaData и DataHub. Если говорить про остальные два варианта, то Apache Atlas отпадает в общем по тому, что заточен больше на системы обработки данных связанные с HADOOP, который не используется в отделе обработки данных ООО «Фарпост».

3.4 Обоснование выбора OpenMetadata

На основе проведенного сравнения для внедрения в информационную систему ООО «Фарпост» выбрана платформа OpenMetadata. Данный выбор обусловлен следующими факторами.

OpenMetadata обеспечивает нативную поддержку всех необходимых интеграций. Особенно важно наличие коннектора для ClickHouse, который позволяет автоматически обнаруживать не только схемы таблиц, но и детали движков, параметры партиционирования и индексы. Визуализация data lineage в OpenMetadata предоставляет сквозной просмотр пути данных от исходных MySQL-таблиц через Kafka и Altinity Sink до конечных результатов в Airflow.

Также платформа разработана с учетом облачных принципов и поставляется с официальными Helm-чартами [10], что обеспечивает бесшовную интеграцию в существующий Kubernetes-кластер без дублирования компонентов. Архитектура OpenMetadata поддерживает горизонтальное масштабирование всех сервисов, что гарантирует обработку метаданных от тысяч таблиц при сохранении отказоустойчивости.

отличие от Apache Atlas, который требует значительной кастомизации для работы с необходимыми платформами, и DataHub, фокусирующегося преимущественно на визуализации, OpenMetadata предлагает оптимальный баланс функционала и современных возможностей. Активное сообщество и корпоративная поддержка со стороны компаний-разработчиков обеспечивают регулярные обновления и быстрое решение проблем.

Платформа OpenMetadata позволяет реализовать все требования без необходимости интеграции нескольких специализированных решений, что значительно снижает стоимость владения и сложность эксплуатации. Готовые коннекторы почти для всех компонентов стека отдела данных ООО «Фарпост» минимизируют необходимость своей разработки и ускоряют время выхода на продуктивное использование.

Таким образом, OpenMetadata является оптимальным решением, полностью удовлетворяющим функциональным и нефункциональным требованиям проекта, при этом обеспечивая минимальные риски внедрения и максимальную отдачу от инвестиций в управление метаданными.

3.5 План внедрения

Внедрение системы управления метаданными будет осуществляться поэтапно с использованием методологии Agile. Общий срок реализации проекта составляет 14 недель. На рисунке 5 представлена диаграмма Ганта ключевых этапов внедрения.

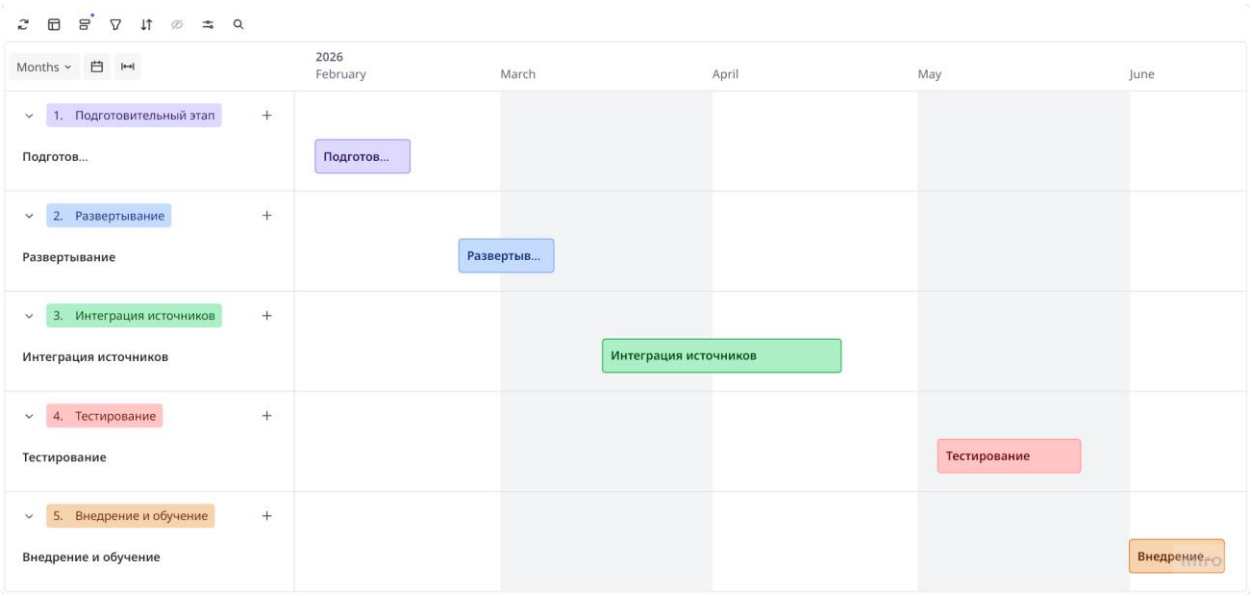

Рисунок 5 – Диаграмма Ганта ключевых этапов внедрения.

Ключевые этапы и зависимости:

1) Подготовительный этап (недели 1-2, Февраль): Финализация функциональных требований и утверждение Логической архитектуры решения. Проектирование схем потоков данных для ingestion-процессов. Подготовка изолированного тестового окружения в Kubernetes для безопасного прототипирования.

2) Развертывание (недели 3-6, Февраль – Март): Реализация Интеграции в среду Kubernetes: настройка Helm-чартов, конфигурация Resource Quotas и Secrets. Развертывание ядра платформы OpenMetadata. Первичная Настройка структуры метаданных: создание доменов, сервисных аккаунтов и базового бизнес-глоссария перед подключением источников. Требует координации с DevOps-командой для выделения вычислительных ресурсов.

3) Интеграция источников (недели 5-10, Март – Апрель): Поэтапный запуск ingestion-процессов с наложением этапов для ускорения:

1) Недели 5-7: Подключение Kafka и Debezium (сбор схем топиков и CDC-событий);

2) Недели 6-8: Подключение коннектора Altinity Sink Connector;

3) Недели 7-9: Реализация кастомного извлечения данных из Apache Airflow и его запуск;

4) Недели 8-10: Сканирование системных таблиц ClickHouse.

4) Тестирование (недели 9-12, Апрель): Выполнение Плана валидации и тестирования:

1) Недели 9-10 (Функциональное): Проверка полноты и корректности собранных метаданных;

2) Неделя 11 (Нагрузочное): Эмуляция пиковых нагрузок для проверки стабильности Kubernetes-подов;

3) Недели 11-12 (Пользовательское тестирование): Привлечение аналитиков для проверки удобства навигации и актуальности графа происхождения данных.

5) Внедрение и обучение (недели 13-14, Май – Июнь): Пилотная эксплуатация. Информирование сотрудников о работе с бизнес-глоссарием и поиском данных. Валидация достижения Ожидаемых эффектов, перевод системы в режим нормальной эксплуатации.

3.6 Логическая архитектура решения

Логическая архитектура проектируемой системы управления метаданными построена на принципах микросервисной архитектуры и событийно-ориентированного взаимодействия компонентов. Решение интегрируется в существующую инфраструктуру ООО «Фарпост» с минимальным воздействием на текущие процессы обработки данных. Архитектура включает четыре основных слоя: источники данных, сбор и обработка метаданных, хранилище и управление, представление и взаимодействие.

Слой источников данных включает существующие компоненты инфраструктуры, из которых будут извлекаться метаданные:

1) Базы данных MySQL – источники транзакционных данных с включенным бинарным логгированием для CDC;

2) Apache Kafka – централизованная шина данных, содержащая события от Debezium с изменениями из MySQL;

3) Altinity Sink Connector – компонент, осуществляющий репликацию данных из Kafka в аналитическую СУБД ClickHouse;

4) Apache Airflow – платформа оркестрации, содержащая DAG-файлы с SQL-скриптами для преобразования данных;

5) ClickHouse – аналитическая база данных, хранящая результаты ETL-процессов.

Слой сбора и обработки метаданных представлен специализированными инжест-сервисами, которые работают в фоновом режиме и не влияют на производительность основных систем:

1) Kafka Connector– сервис, подключающийся к Kafka как read-only consumer для извлечения информации о топиках, схемах сообщений (через Schema Registry) и consumer groups;

2) Debezium Connector – компонент, опрашивающий REST API Debezium для получения конфигурации коннекторов, статусов репликации и позиций offset;

3) Altinity Sink Observer – модуль, собирающий метаданные о конфигурации репликации, маппинге таблиц между MySQL и ClickHouse, параметрах обработки схем;

4) Airflow Lineage Extractor – самописный плагин для Airflow для автоматического извлечения зависимостей между задачами, источниками данных и конечными результатами;

5) Airflow Connector – стандартный коннектор для получения общих метаданных из Airflow;

6) ClickHouse Connector – сервис, выполняющий периодическое сканирование системных таблиц ClickHouse для обнаружения схем, движков таблиц и параметров партиционирования.

Слой хранилища и управления реализован на базе платформы OpenMetadata и включает:

1) Ingestion Framework – распределенная система обработки задач по сбору метаданных с возможностью горизонтального масштабирования;

2) MySQL Database – основное хранилище метаданных;

3) Elasticsearch Cluster – индекс для обеспечения быстрого поиска и фильтрации метаданных.

Слой представления и взаимодействия обеспечивает пользовательский доступ к системе:

1) Веб-интерфейс OpenMetadata – реактивный UI для визуализации каталога данных, data lineage и управления глоссарием;

2) REST API – программный интерфейс для интеграции с другими системами компании;

3) Система уведомлений – сервис рассылки алертов о критических изменениях в схемах данных или проблемах с репликацией.

Все компоненты развертываются в существующем Kubernetes-кластере компании в отдельном namespace metadata-system с соблюдением принципов изоляции и безопасности. Коммуникация между сервисами осуществляется через:

1) Внутренние Kubernetes-сервисы для синхронного взаимодействия;

2) Kafka-топики для асинхронной передачи событий об изменении метаданных;

3) REST API для внешних интеграций.

Особое внимание уделено отказоустойчивости архитектуры: все инжест-сервисы поддерживают механизм checkpoint'ов для восстановления после сбоев, основные компоненты развернуты с множественными репликами.

Архитектура полностью соответствует принципам cloud-native разработки и оптимально использует существующую Kubernetes-инфраструктуру компании, что минимизирует затраты на внедрение и эксплуатацию системы.

3.7 Проектирование ingestion-процессов

Проектирование процессов сбора метаданных (ingestion) является критически важным этапом внедрения системы управления метаданными, так как от их надежности и эффективности напрямую зависит актуальность и полнота каталога данных. В ходе анализа существующих решений было установлено, что платформа OpenMetadata предоставляет наиболее полный набор инструментов для автоматизации ingestion-процессов, включая готовые коннекторы для

большинства компонентов инфраструктуры ООО «Фарпост», что позволяет минимизировать необходимость кастомной разработки.

Для извлечения метаданных из Apache Kafka используется штатный коннектор, работающий по принципу периодического опроса (pull-model). Настройка коннектора предполагает интеграцию с Confluent Schema Registry, что критически важно для интерпретации событий Debezium, передаваемых в формате Avro.

Коннектор настраивается в режиме инкрементальной загрузки (incremental ingestion) с интервалом в 15 минут. Ниже представлен пример конфигурации коннектора в формате YAML (рисунок 6).

```yaml
source:
  type: kafka
  serviceName: local_kafka_service
  serviceConnection:
    config:
      bootstrapServers: kafka-cluster:9092
      schemaRegistryURL: http://schema-registry:8081
      # Настройка Schema Evolution Tracking
      consumerConfig:
        group.id: openmetadata-ingestion-group
  sourceConfig:
    config:
      generateSampleData: true
      includeTags: true
      # Инкрементальная логика
      incremental:
        enabled: true
        offset: 15
```

Рисунок 6 – Конфигурация коннектора

Данная конфигурация позволяет автоматически фиксировать изменения в схемах (Schema Evolution) и сопоставлять их с версиями топиков, обеспечивая прозрачность изменений в структуре событий Debezium.

Учитывая наличие в отделе данных ООО «ФарПост» нестандартных структур DAG-файлов, стандартный коннектор Airflow был дополнен кастомным провайдером происхождения данных. Вместо базового сканирования используется комбинация двух подходов:

1) Airflow Lineage Backend: отправка метаданных в реальном времени при завершении каждого таска;

2) Custom Metadata Extractor: доработанный модуль, который анализирует специфичные атрибуты объектов в коде DAG для извлечения неявных зависимостей.

Пример настройки lineage_backend в airflow.cfg (рисунок 7).

```
[lineage]
backend = openmetadata_managed_apis.lineage.openmetadata.OpenMetadataLineage
openmetadata_api_endpoint = http://openmetadata-server:8585/api
openmetadata_auth_provider = openmetadata
```

Рисунок 7 – Конфигурация коннектора

Для корректной обработки "нестандартных" DAG-файлов в код загрузчика были внесены изменения, позволяющие парсить нестандартные аргументы, в которых инженеры данных фиксируют пути к источникам и приемникам.

Для ClickHouse будет задействован ClickHouse коннектор OpenMetadata, официально поддерживаемый сообществом. Коннектор обеспечивает:

1) Автоматическое обнаружение баз данных и таблиц;

2) Сбор детальной информации о движках таблиц;

3) Извлечение параметров партиционирования, TTL настроек и индексов;

4) Мониторинг размеров таблиц и количества строк;

5) Отслеживание изменений в схемах через system tables.

Особое внимание уделено специфике ClickHouse: коннектор корректно обрабатывает особенности движка ReplacingMergeTree, используемого в Altinity Sink, и автоматически определяет логические связи между таблицами-источниками и таблицами-приемниками. Ниже приведена конфигурация коннектора для Clickhouse (рисунок 8).

```yaml
source:
  type: clickhouse
  serviceName: clickhouse_analytics_prod
  serviceConnection:
    config:
      type: Clickhouse
      hostPort: "clickhouse-cluster.namespace.svc.cluster.local:8123"
      username: "openmetadata_user"
      password: "secure_password"
      database: "default"
  sourceConfig:
    config:
      type: DatabaseMetadata
      includeTables: true
      includeViews: true
      includeTags: true
      includeLineage: true
      databaseFilterPattern:
        includes: ["analytics_.*", "raw_data", "marts"]
        excludes: ["system", "information_schema", "default"]
      generateSampleData: true
      markDeletedTables: true
```

Рисунок 8 – Конфигурация коннектора

Для Altinity Sink Connector готового коннектора в OpenMetadata нет, в том числе и у других платформ, которые были проанализированы выше, поэтому был разработан свой подход на основе OpenMetadata Ingestion Framework SDK. Решение включает:

1) Создание Python-модуля, обращающегося к REST API Altinity Sink Connector для получения конфигурации репликации;

2) Разработка парсера для извлечения информации о маппинге таблиц (MySQL → ClickHouse);

3) Интеграция с механизмом Custom Entities OpenMetadata для хранения специфичных параметров репликации;

4) Использование шаблонов Jinja для генерации бизнес-описаний на основе конфигурации коннектора.

Кастомный модуль спроектирован как расширение стандартного OpenMetadata Ingestion Framework, что гарантирует совместимость с общей архитектурой и возможность легкого обновления в будущем.

Все ingestion-процессы будут организованы по единой схеме distributed ingestion workers, реализованной в OpenMetadata:

1) Orchestrator Service – центральный компонент, отвечающий за планирование задач сбора метаданных на основе заданного расписания и приоритетов;

2) Worker Nodes – масштабируемые pod'ы в Kubernetes, выполняющие конкретные задачи сбора данных;

3) Task Queue – распределенная очередь задач (на основе Celery + Redis), обеспечивающая балансировку нагрузки и отказоустойчивость;

4) State Store – PostgreSQL база данных для хранения состояния выполнения задач и истории изменений.

Для каждого типа источника будет создана отдельная ingestion pipeline с настраиваемыми параметрами:

1) Частота сбора: для критически важных компонентов (Kafka топики с CDC) – каждые 15 минут, для статических источников (Airflow DAGs) – раз в сутки;

2) Глубина сканирования: полное сканирование при первом запуске, инкрементальное – при последующих;

3) Параллелизм: настройка количества одновременных подключений к источнику для минимизации нагрузки;

4) Retry policy: автоматические повторные попытки при временных сбоях с экспоненциальной задержкой.

Особое внимание уделено безопасности: все подключения к источникам данных будут использовать сервисные аккаунты с минимально необходимыми правами (read-only для метаданных), а учетные данные будут храниться в Kubernetes Secrets с автоматической ротацией.

Архитектура ingestion-процессов спроектирована с учетом требований к отказоустойчивости и масштабируемости:

1) Checkpointing mechanism – каждый ingestion worker сохраняет состояние обработки после завершения порции данных, что позволяет продолжить работу с последней точки при перезапуске;

2) Асинхронная обработка – длительные операции (сканирование больших таблиц ClickHouse) выполняются асинхронно с отправкой уведомлений по завершении;

3) Горизонтальное масштабирование – количество worker nodes автоматически регулируется через Kubernetes HPA (Horizontal Pod Autoscaler) на основе метрик загрузки CPU и очереди задач;

4) Изоляция источников – сбои в работе одного коннектора не влияют на работу других благодаря разделению на отдельные pipeline'ы.

Использование встроенных коннекторов OpenMetadata в сочетании с кастомной разработкой для Altinity Sink дает следующие преимущества:

1) Единообразие – все метаданные собираются по единой схеме и хранятся в согласованном формате;

2) Поддержка сообщества – встроенные коннекторы активно развиваются и тестируются сообществом OpenMetadata;

3) Гибкость – возможность быстрой адаптации под изменения в инфраструктуре через конфигурационные файлы.

Для обеспечения интеграции и автоматизации процессов документирования в рамках проектируемой системы необходимо детально рассмотреть структуру извлекаемых метаданных. Переход от ручного документирования к автоматизированному сбору (ingestion) базируется на способности платформы OpenMetadata интерпретировать «сырые» данные, поступающие из различных слоев инфраструктуры: брокеров сообщений, баз данных и конфигурационных файлов коннекторов. Ниже приведены примеры структур метаданных

Из Clickhouse требуется извлечь следующие метаданные, которые будут собранны в виде информации системных таблиц, содержащих информацию о структуре всех таблиц (Таблица 5):

Таблица 5 – Описание метаданных из Clickhouse

Тип метаданных	Виды метаданных
Структурные	Имена столбцов, типы данных, порядок и ключ сортировки таблицы, движок таблицы, ключ и тип партиционирования, время жизни данных.
Статистические	Количество строк, размер данных, количество партиций.

```
Row 1:
──────
database:                   base
name:                       t1
uuid:                       81b1c20a-b7c6-4116-a2ce-7583fb6b6736
engine:                     MergeTree
is_temporary:               0
data_paths:                 ['/var/lib/clickhouse/store/81b/81b1c20a-b7c6-4116-a2ce-7583fb6b6736/']
metadata_path:              /var/lib/clickhouse/store/461/461cf698-fd0b-406d-8c01-5d8fd5748a91/t1.sql
metadata_modification_time: 2021-01-25 19:14:32
dependencies_database:      []
dependencies_table:         []
create_table_query:         CREATE TABLE base.t1 (`n` UInt64) ENGINE = MergeTree ORDER BY n
engine_full:                MergeTree ORDER BY n
as_select:                  SELECT database AS table_catalog
partition_key:
sorting_key:                n
primary_key:                n
sampling_key:
storage_policy:             default
total_rows:                 1
total_bytes:                99
lifetime_rows:              NULL
lifetime_bytes:             NULL
comment:
has_own_data:               0
```

Рисунок 9 – Пример метаданных таблицы из Clickhouse

Из Apache Airflow требуется извлечь следующие данные о связях между агрегациями, которые будут собраны с помощью нового созданного модуля. Данные будут представлены в виде записей об источниках агрегации, и финальной таблицы (Таблица 6). Эти данные будут собираться из разработанного для отдела класса DAG в котором есть информация об источниках данных для агрегации, а также результирующей таблицы, что и нужно для построения графа зависимостей и происхождения данных.

Таблица 6 – Описание метаданных из Apache Airflow

Тип метаданных	Виды метаданных
Структурные	Связи между источниками данных.

Из Debezium требуется извлечь метаданные коннекторов, включая конфигурацию источников, статусы репликации, схемы таблиц (Таблица 7):

Таблица 7 – Описание метаданных из Debezium,

Тип метаданных	Виды метаданных
Структурные	Имена таблиц и схем баз данных, колонки, типы данных, первичный ключ
Статистические	Статистика репликации, объем данных, количество операций по типам, количество пропущенных сообщений, количество коннекторов.

```
{
  "before": { "id": 101, "name": "Laptop", "price": 950.00 },
  "after": { "id": 101, "name": "Laptop", "price": 1050.00 },
  "source": {
    "version": "2.1.2.Final",
    "connector": "mysql",
    "name": "db_server_1",
    "ts_ms": 1674211200000,
    "snapshot": "false",
    "db": "inventory",
    "table": "products",
    "server_id": 223344,
    "gtid": null,
    "file": "mysql-bin.000003",
    "pos": 456,
    "row": 0,
    "thread": 15,
    "query": null
  },
  "op": "u",
  "ts_ms": 1674211200500,
  "transaction": {
    "id": "551",
    "total_order": 1,
    "data_collection_order": 1
  }
}
```


Рисунок 10 – пример метаданных из Debezium

Из Kafka необходимо извлечь метаданные в формате JSON схем сообщений, информацию о топиках, партициях и группах пользователей (Таблица 8):

Таблица 8 – Описание метаданных из Kafka

Тип метаданных	Виды метаданных
Структурные	Схемы топиков, конфигурация сохранения, схема сообщений, структура кластера.
Статистические	Размер топиков в байтах и количество сообщений, отставание обработки, средний размер сообщения.

```
{
  "metadata": {
    "offset": 450123,
    "partition": 2,
    "timestamp": 1705842000000,
    "timestampType": "CreateTime",
    "topic": "crm.public.orders",
    "key": "order_id_995"
  },
  "headers": {
    "correlation_id": "a1-b2-c3-d4",
    "source_system": "Magento_Ecom",
    "schema_version": "2",
    "content_type": "application/json"
  },
  "value": {
    "order_id": 995,
    "user_id": 10,
    "amount": 1500.50,
    "currency": "RUB"
  }
}
```


Рисунок 11 – пример метаданных из Kafka

Из Altinity Sink Connector требуется извлечь метаданные о конфигурации репликации, статусах синхронизации, маппинге таблиц между источником и ClickHouse, а также информации об автоматической обработке изменений схемы (Таблица 9):

Таблица 9 – Описание метаданных из Altinity Sink Connector

Тип метаданных	Виды метаданных
Структурные	Конфигурация маппинга таблиц, параметры движка таблиц.
Статистические	Статистика репликации, отставание репликации, количество и частота сбоев с деталями восстановления.

```
<clickhouse>
    <topic_names>crm.public.orders</topic_names>
    <table_name>orders_raw</table_name>
    <database_name>operational_data</database_name>
    <skip_database_on_startup>true</skip_database_on_startup>
    <column_names>order_id,amount,status,updated_at</column_names>
    <replace_char_with_underscore>.</replace_char_with_underscore>
</clickhouse>
```

Рисунок 12 – Пример метаданных Altinity Sink Connector

Таким образом, проектируемая архитектура ingestion-процессов полностью соответствует требованиям проекта и обеспечивает надежный, масштабируемый и отказоустойчивый сбор метаданных из всех компонентов инфраструктуры ООО «Фарпост».

3.8 Настройка структуры метаданных в OpenMetadata

Проектирование структуры метаданных в рамках внедрения OpenMetadata в отдел данных ООО «Фарпост» направлено на создание формализованной модели, описывающей технические активы и их бизнес-контекст. В данном разделе определяются конкретные параметры логической организации каталога.

Для обеспечения управляемости каталог разделяется на функциональные домены. Проектное решение предполагает создание трехуровневой структуры:

Домен «Infrastructure»: включает технические метаданные Kafka-кластеров, Kubernetes-сервисов и логов Debezium.

Домен «Operational Data»: содержит зеркальные таблицы MySQL в ClickHouse и соответствующие топики Kafka.

Домен «Analytics & Reporting»: содержит агрегированные таблицы ClickHouse и соответствующие DAG-файлы Airflow.

За каждым доменом закрепляется роль владельца, ответственного за верификацию метаданных.

Глоссарий проектируется как древовидная структура, исключающая дублирование терминов. В рамках проекта определены следующие категории:

Маркетинговые показатели: лиды, конверсия.

Технические сущности: User_ID, Order_ID, Event_Timestamp.

Статусы процессов: Order_Status, Deal_status.

Таблица 10 – Спецификация атрибутов термина глоссария

Атрибут	Тип данных	Обязательность	Описание
DisplayName	String	Да	Понятное бизнес-название термина
Description	Markdown	Да	Четкое определение формулы или смысла
Synonyms	List	Нет	Похожие названия в разных ИС
Related Terms	Link	Нет	Связи с другими терминами (ассоциации)
Status	Enum	Да	Draft/Approved/ Deprecated

Для автоматизации поиска и обеспечения безопасности проектируется две группы тегов (Classification):

Классификация «Data Sensitivity» (Конфиденциальность):

1) PII.Personal - Данные пользователей (ФИО, телефон).

2) PII.Sensitive - Финансовая информация.

3) Public: Общедоступные данные.

Классификация «LifeCycle» (Жизненный цикл):

1) Production: Актуальные данные, используемые в отчетах.

2) Sandbox: Тестовые таблицы.

3) Legacy: Устаревшие данные, запланированные к удалению.

Стандартная модель OpenMetadata расширяется через механизм Custom Properties для учета особенностей ClickHouse и Altinity Sink Connector.

Таблица 11 – Кастомные свойства для сущности «Table»

Свойство	Тип	Обязательность	Назначение
ch_engine_type	String	Да	Тип движка таблицы (ReplicatedMergeTree, Distributed и т.д.)
partition_key	String	Да	Ключ партиционирования для оптимизации выполнения запросов

replication_status	Boolean	Да	Наличие и статус активной репликации через Altinity Sink

Продолжение таблицы 11

Свойство	Тип	Обязательность	Назначение
ttl_policy	String	Нет	Правила (Time To Live) для автоматического удаления данных

Проектная модель Lineage строится на основе графа, где узлами являются сущности, а ребрами – процессы трансформации. Проектируемая цепочка выглядит следующим образом:

1) Source: MySQL.Table (Внешний источник);
2) Process: Debezium Connector (Захват изменений);
3) Entity: Kafka.Topic (Буфер событий);
4) Process: Altinity Sink Connector (Репликация в КХ);
5) Entity: ClickHouse.Table_Raw (Сырые данные);
6) Process: Airflow.DAG (Трансформация/Агрегация);
7) Entity: ClickHouse.Table_Final (Витрина).

Результатом данного проектирования является детальная спецификация, на основе которой на этапе реализации настраиваются JSON-схемы в OpenMetadata и конфигурируются профилировщики данных. Это гарантирует, что каждый технический объект в системе будет снабжен необходимым бизнес-контекстом и техническими параметрами.

3.9 Интеграция системы управления метаданными в существующую среду Kubernetes

Интеграция платформы OpenMetadata в инфраструктуру ООО «Фарпост» осуществляется с учетом уже используемой контейнерной среды Kubernetes [11] и направлена на минимальное влияние на существующие сервисы обработки данных. Основным принципом интеграции является изоляция системы управления метаданными от критически важных производственных компонентов при сохранении возможности масштабирования и отказоустойчивой работы.

Развертывание OpenMetadata выполняется с использованием официальных Helm-чартов (рисунок 13), что позволяет описать конфигурацию системы в декларативном виде и упростить дальнейшее сопровождение и обновление платформы. В рамках Helm-конфигурации настраиваются отдельные Deployment-ресурсы для API-сервера, веб-интерфейса и рабочих узлов, выполняющих ingestion-задачи. Все параметры подключения к внешним сервисам, включая базы

данных и источники метаданных, передаются через Kubernetes Secrets, что обеспечивает безопасное хранение учетных данных и их централизованное управление.

```yaml
global:
  namespace: openmetadata
  imageRegistry: docker.io

openmetadata:
  replicaCount: 2
  image:
    repository: openmetadata/server
    tag: 1.2.0

  resources:
    limits:
      cpu: "2"
      memory: "4Gi"
    requests:
      cpu: "1"
      memory: "2Gi"

  config:
    database:
      host: "mysql-db.openmetadata.svc.cluster.local"
      port: 3306
      dbName: "openmetadata_db"
      auth:
        username: "openmetadata_user"
        password:
          secretRef: openmetadata-secrets
          secretKey: mysql-password

    elasticsearch:
      host: "elasticsearch.openmetadata.svc.cluster.local"
      port: 9200
      auth:
        username: "admin"
        password:
          secretRef: openmetadata-secrets
          secretKey: elasticsearch-password
```

Рисунок 13 – Пример Helm chart OpenMetadata

Для логической и ресурсной изоляции компонентов OpenMetadata в кластере создается отдельное пространство имен (namespace), предназначенное исключительно для системы управления метаданными. В рамках данного namespace задаются квоты и лимиты на использование вычислительных ресурсов, что предотвращает негативное влияние процессов сбора и обработки метаданных на работу аналитических систем, таких как ClickHouse и Kafka.

Ограничение по памяти особенно важно при обработке крупных схем данных и метаданных Kafka-топиков, где возможны пиковые нагрузки.

Доступ пользователей к веб-интерфейсу OpenMetadata организуется через Ingress-контроллер с включенным TLS-шифрованием. Это обеспечивает безопасный доступ к системе из корпоративной сети и позволяет интегрировать OpenMetadata в существующую инфраструктуру сетевой безопасности компании. Внутреннее взаимодействие между компонентами платформы осуществляется через стандартные Kubernetes Service, что обеспечивает прозрачную балансировку нагрузки и отказоустойчивость при перезапуске отдельных pod’ов.

Для обеспечения наблюдаемости и оперативного реагирования на сбои система управления метаданными интегрируется с существующим стеком мониторинга компании. Метрики состояния API-сервера, времени выполнения ingestion-процессов и загрузки рабочих узлов экспортируются в Prometheus и визуализируются в Grafana. Это позволяет DevOps-команде контролировать стабильность работы OpenMetadata и своевременно выявлять проблемы, связанные с актуализацией метаданных или ростом нагрузки.

3.10 План валидации и тестирования системы управления метаданными

Для обеспечения корректной эксплуатации системы управления метаданными в рамках проекта разработан план валидации и тестирования, направленный на проверку полноты, актуальности и корректности собираемых метаданных. Валидация рассматривается как обязательный этап внедрения, позволяющий подтвердить соответствие реализованного решения проектным требованиям и выявить потенциальные расхождения между фактическим состоянием инфраструктуры и отображаемыми в каталоге метаданными.

Валидация полноты метаданных направлена на подтверждение того, что все ключевые источники данных и процессы обработки представлены в системе OpenMetadata. В рамках данного этапа проверяется наличие в каталоге всех активных Kafka-топиков с CDC-событиями, таблиц ClickHouse, задействованных в аналитических витринах, а также DAG-файлов Apache Airflow, участвующих в преобразовании данных. Полнота считается достигнутой, если для каждого элемента инфраструктуры существует соответствующая сущность в каталоге метаданных, и она корректно привязана к домену и источнику данных.

Проверка актуальности метаданных ориентирована на контроль своевременного обновления информации при изменениях в инфраструктуре. В рамках тестирования моделируются типовые изменения, такие как добавление новой колонки в таблицу MySQL, изменение схемы Kafka-сообщения или модификация DAG в Airflow. После внесения изменений проверяется, что система управления метаданными фиксирует обновления в рамках заданного интервала ingestion-процессов и отображает актуальное состояние схем, связей и параметров

хранения данных. Дополнительно контролируется отсутствие устаревших или дублирующихся объектов в каталоге.

Валидация корректности метаданных включает проверку точности отображаемых связей и атрибутов. Особое внимание уделяется проверке data lineage: для выбранных аналитических таблиц ClickHouse анализируется соответствие отображаемого пути данных фактическим процессам репликации и обработки. Корректность считается подтвержденной, если цепочки происхождения данных корректно отражают источники, промежуточные этапы обработки и конечные результаты без логических разрывов и некорректных связей. Также проверяется корректность заполнения пользовательских атрибутов, таких как тип движка таблицы и параметры партиционирования.

Завершающим этапом валидации является пользовательское тестирование с участием аналитиков и разработчиков. В рамках данного этапа оценивается удобство навигации по каталогу, понятность структуры доменов и глоссария, а также практическая применимость системы для поиска данных и анализа их происхождения. Результаты пользовательского тестирования используются для корректировки структуры метаданных и уточнения проектных решений перед переводом системы в режим промышленной эксплуатации.

Таким образом, разработанный план валидации и тестирования обеспечивает контроль качества системы управления метаданными и подтверждает ее готовность к использованию в корпоративной информационной системе ООО «Фарпост».

3.11 Оценка ожидаемых эффектов от внедрения системы управления метаданными

Внедрение системы управления метаданными, спроектированной в рамках данной курсовой работы, ориентировано на достижение измеримых организационных и технологических эффектов в деятельности отдела обработки данных ООО «Фарпост». Оценка ожидаемых эффектов проводится на основе анализа текущего состояния процессов работы с данными и предполагаемых изменений после внедрения централизованного каталога метаданных.

Одним из ключевых ожидаемых эффектов является повышение прозрачности данных. Централизованный каталог с поддержкой графа происхождения данных позволяет сотрудникам оперативно определять происхождение данных, их взаимосвязи и степень надежности. Это снижает зависимость от неформальных знаний отдельных специалистов и уменьшает риски, связанные с кадровыми изменениями в команде.

Значительный эффект ожидается в области повышения эффективности аналитической и инженерной деятельности. Сокращается время поиска нужных таблиц, понимания их структуры и назначения, а также анализа влияния изменений в источниках данных. Аналитики получают

возможность быстрее подключаться к новым витринам данных, а разработчики – принимать более обоснованные решения при изменении ETL-процессов и схем хранения данных.

Важным результатом внедрения является повышение качества данных. Наличие централизованных метаданных, информации о схемах, правилах обработки и показателях качества создает предпосылки для системного контроля данных и более быстрого выявления ошибок в процессах репликации и агрегации. Отображение происхождения позволяет оперативно локализовать источник проблем и сократить время их устранения.

Таким образом, ожидаемые эффекты от внедрения системы управления метаданными выражаются в повышении прозрачности, управляемости и качества данных, а также в снижении операционных затрат на сопровождение аналитической инфраструктуры. Это подтверждает практическую целесообразность и ценность реализуемого проектного решения.

Заключение

В рамках данной курсовой работы было выполнено проектирование внедрения системы управления метаданными в корпоративную информационную систему ООО «Фарпост». В ходе работы проведен анализ существующей ИТ-инфраструктуры и текущих процессов управления метаданными, выявлены ключевые проблемы, связанные с фрагментацией, неактуальностью и отсутствием сквозной прозрачности данных.

На основе проведенного анализа были сформулированы требования к системе управления метаданными и выполнен сравнительный анализ современных open-source решений. В качестве целевой платформы выбрана OpenMetadata, наиболее полно соответствующая техническим и организационным требованиям проекта. Разработана логическая архитектура решения, спроектированы процессы сбора метаданных, структура каталога данных и модель интеграции в существующую Kubernetes-среду.

Особое внимание в работе уделено проектированию структуры метаданных, бизнес-глоссария и механизмов отображения происхождения данных, что обеспечивает практическую применимость решения в условиях реальной корпоративной инфраструктуры. Также разработан план валидации и тестирования системы и выполнена оценка ожидаемых эффектов от ее внедрения.

Результаты курсовой работы подтверждают, что внедрение системы управления метаданными является обоснованным и эффективным шагом для повышения качества управления данными, прозрачности аналитических процессов в ООО «Фарпост». Разработанный проект может быть использован в качестве основы для практической реализации решения и дальнейшего развития корпоративной платформы работы с данными.

Список использованных источников

1 Apache Kafka Documentation – Текст: электронный // Apache Software Foundation: [сайт]. – 2025. – URL: https://kafka.apache.org/documentation/ (дата обращения: 09.01.2026).

2 Debezium Documentation – Текст: электронный // Debezium: [сайт]. – 2025. – URL: https://debezium.io/documentation/ (дата обращения: 09.01.2026).

3 ClickHouse Documentation – Текст: электронный // ClickHouse: [сайт]. – 2025. – URL: https://clickhouse.com/docs/ (дата обращения: 09.01.2026).

4 Altinity Sink Connector for ClickHouse: official documentation – Текст: электронный // GitHub: [сайт]. – 2025. – URL: https://github.com/Altinity/clickhouse-sink-connector (дата обращения: 11.01.2026).

5 Apache Airflow Documentation – Текст: электронный // Apache Software Foundation: [сайт]. – 2025. – URL: https://airflow.apache.org/docs/ (дата обращения: 09.01.2026).

6 OpenMetadata Documentation – Текст: электронный // OpenMetadata: [сайт]. – 2025. – URL: https://docs.open-metadata.org/ (дата обращения: 09.01.2026).

7 Apache Atlas Documentation – Текст: электронный // Apache Software Foundation: [сайт]. – 2025. – URL: https://atlas.apache.org/ (дата обращения: 09.01.2026).

8 DataHub Documentation – Текст: электронный // DataHub Project: [сайт]. – 2025. – URL: https://datahubproject.io/docs/ (дата обращения: 09.01.2026).

9 Amundsen Documentation: official portal – Текст: электронный // Amundsen: [сайт]. – 2025. – URL: https://www.amundsen.io/amundsen/ (дата обращения: 09.01.2026).

10 Helm Documentation – Текст: электронный // Helm: [сайт]. – 2025. – URL: https://helm.sh/docs/ (дата обращения: 09.01.2026).

11 Kubernetes Documentation – Текст: электронный // Kubernetes: [сайт]. – 2025. – URL: https://kubernetes.io/docs/ (дата обращения: 09.01.2026).