

УДК 004.65

**РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ РАБОТЫ
С РАСПРЕДЕЛЕННЫМ S3-СОВМЕСТИМЫМ ХРАНИЛИЩЕМ**

Голованов А.А., студент гр. ПИБ-221, IV курс,
Якунин Н.А., студент гр. ПИБ-221, IV курс
Научный руководитель: Славолубова Я.В., к.т.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

В статье представлена архитектура и реализация веб-сервиса для управления файловыми данными в распределенном S3-совместимом хранилище. Предложенный подход направлен на повышение безопасности за счет исключения прямого взаимодействия пользователей с объектным хранилищем и использования промежуточного уровня в виде специализированного Backend-сервиса и реляционной базы данных. Рассматриваются модель хранения метаданных, механизмы поиска по тегам и ключам, а также методы предотвращения несанкционированного доступа к объектам посредством защиты прямых ссылок.

Современная парадигма цифровизации бизнес-процессов характеризуется экспоненциальным ростом объемов хранимой информации. Традиционные иерархические файловые системы сталкиваются с ограничениями масштабирования до больших значений, особенно в условиях распределенных облачных вычислений. Объектные хранилища, работающие по протоколу S3, стали стандартом благодаря своей способности обеспечивать практически безграничную горизонтальную масштабируемость и высокую доступность данных [1].

Прямое использование S3-решений в пользовательских интерфейсах связано с рядом проблем. Как показывает практика, основными проблемами является отсутствие удобного поиска по метаданным, а также сложности в организации системы управления правами доступа и открытостью инфраструктурных ссылок. Для решения данных проблем была поставлена задача разработки программного решения, позволяющего обеспечить безопасное взаимодействие с распределенным хранилищем за счет использования промежуточного сервисного слоя.

При построении системы хранения больших объемов данных критически важным является разделение самих файлов и информации о них. Объектные хранилища (такие как MinIO) отлично справляются с хранением данных, но обладают ограниченными возможностями поиска [2]. Для решения этой задачи можно использовать реляционную СУБД (PostgreSQL), которая может обеспечить индексацию и хранение метаданных [3].

В качестве архитектурного подхода была выбрана модель с сервисом, который выступает в роли посредника между пользователем и объектным

хранилищем. Это позволяет не только повысить безопасность, но и абстрагироваться от конкретного поставщика облачных услуг, сохраняя S3-совместимость.

Разработанный веб-сервис функционирует как центральный узел управления и включает в себя следующие компоненты:

1. Backend-слой: реализует RESTful API для приема запросов на загрузку, чтение, модификацию и поиск объектов.
2. Слой хранения метаданных (PostgreSQL): в данной базе, представленной на рисунке 1, фиксируется жизненный цикл каждого объекта. Схема данных спроектирована таким образом, чтобы хранить не только физический путь к файлу, но и его логические атрибуты (теги).
3. Слой объектного хранения (MinIO): кластер распределенных узлов, обеспечивающий отказоустойчивость данных на физическом уровне.

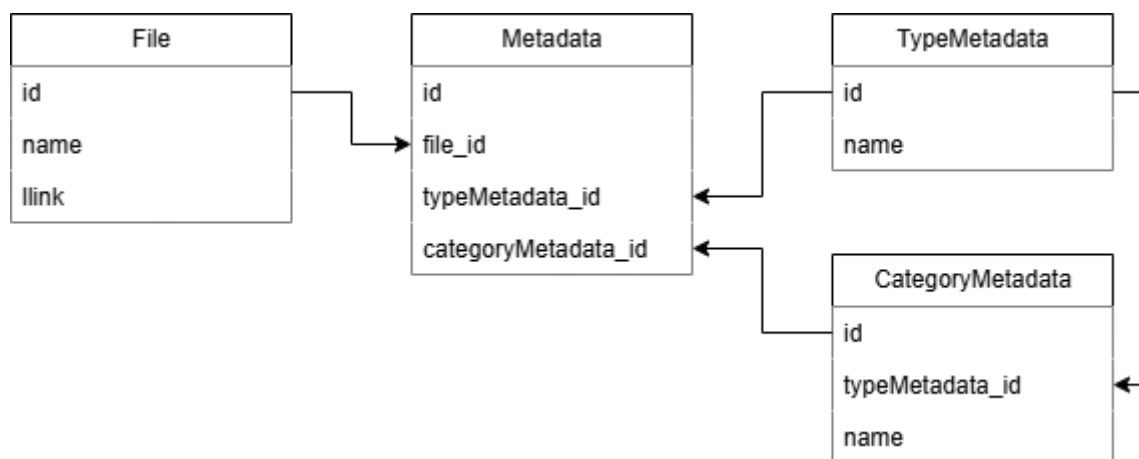


Рисунок 1 – Диаграмма сущность-связь базы данных

При загрузке файла веб-сервис выполняет следующий алгоритм:

1. Валидация и анализ метаданных: сервис анализирует метаданные запроса. Для каждого документа определяются два ключевых идентификатора: тип документа и номер счета.
2. Динамическая маршрутизация в S3: на основе этих ключей формируется логический путь внутри объектной среды. Такая структура позволяет эффективно использовать механизмы префиксов S3 для ускорения листинга объектов.
3. Регистрация в реестре: в PostgreSQL создается запись, содержащая: уникальный идентификатор объекта, сформированный путь в MinIO, системные атрибуты и пользовательские теги для быстрого поиска.

Одной из задач разработки является исключение прямого доступа клиента к объектному хранилищу MinIO. В предложенном решении реализована концепция виртуального проксирования ссылок. В базе данных PostgreSQL вместо прямой ссылки на S3-объект создается вторичный идентификатор. Когда пользователь запрашивает файл, процесс происходит следующим образом (рисунок 2):

1. Запрос к инфраструктуре: конечный пользователь инициирует запрос напрямую к Backend-сервису, передавая внутренний идентификатор требуемого файла.

2. Верификация и поиск метаданных: Backend-сервис выполняет запрос к PostgreSQL для получения атрибутов файла и проверки прав доступа пользователя в контексте текущей сессии. На этом этапе система сопоставляет права субъекта с политиками безопасности.

3. Генерация временного доступа: при успешной авторизации Backend-сервис обращается к MinIO с запросом на генерацию временной подписанной ссылки (pre-signed URLs).

4. Маршрутизация контента: все обращения по полученной ссылке обрабатываются через Backend-сервис, который взаимодействует с MinIO и обеспечивает передачу данных пользователю, не допуская прямого доступа к хранилищу.

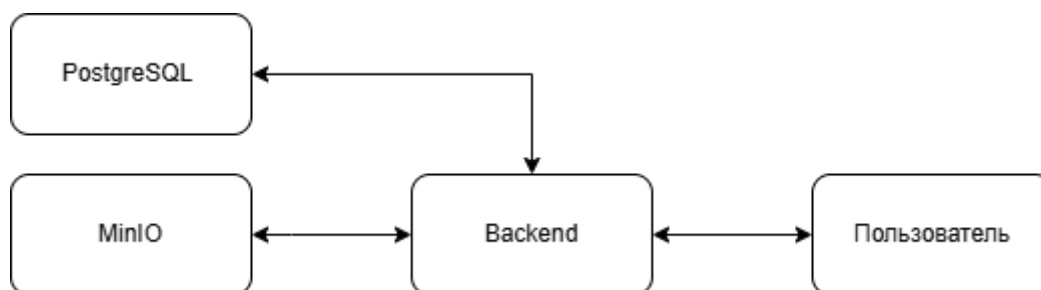


Рисунок 2 – Диаграмма взаимодействия компонентов веб-сервиса

Такой подход гарантирует, что физические адреса серверов хранения и учетные данные никогда не покидают защищенный контур серверной части. Это предотвращает возможность реализации атак типа перебор путей или несанкционированное скачивание данных в обход бизнес-логики.

Благодаря интеграции PostgreSQL, система поддерживает сложные поисковые запросы, которые невозможно выполнить стандартными средствами S3. Пользователи могут осуществлять поиск файлов по:

- свободным тегам, присвоенным вручную или автоматически;
- временным интервалам загрузки;
- комбинации типа документа и содержимого метаданных.

Распределенная природа MinIO позволяет системе масштабироваться горизонтально путем добавления новых узлов в кластер без изменения программного кода веб-сервиса. Использование PostgreSQL в качестве реестра метаданных позволяет в будущем внедрить механизмы автоматического удаления устаревших данных (TTL) и автоматизированную классификацию документов с использованием нейросетевых моделей на этапе загрузки.

Разработанный программный комплекс представляет собой эффективное решение задачи управления корпоративным контентом в распределенных средах. Интеграция S3-совместимого хранилища с реляционной базой данных под

управлением сервиса-посредника позволяет достичь синергетического эффекта: обеспечить надежность и масштабируемость хранения при сохранении высокого уровня безопасности и гибкости поиска. Предложенная архитектура может быть рекомендована к внедрению в информационные системы организаций с жесткими требованиями к защите данных и высокой интенсивностью файлообмена.

Список литературы:

1. Amazon S3 Documentation. Cloud Object Storage Concepts : [сайт]. URL: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html> (дата обращения: 25.03.2026).
2. MinIO Quickstart Guide. High Performance Object Storage : [сайт]. URL: <https://min.io/docs/minio/linux/index.html> (дата обращения: 25.03.2026).
3. PostgreSQL Documentation. The World's Most Advanced Open Source Relational Database : [сайт]. URL: <https://www.postgresql.org/docs/> (дата обращения: 26.03.2026).
4. Sharing objects with pre-signed URLs. Amazon Simple Storage Service User Guide : [сайт]. URL: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSignedURL.html> (дата обращения: 27.03.2026).
5. NGINX Documentation. High-Performance Load Balancing : [сайт]. URL: <https://docs.nginx.com/nginx/admin-guide/> (дата обращения: 27.03.2026).