Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЁТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №2 ПО ДИСЦИПЛИНЕ «ОСНОВЫ ОБЛАЧНЫХ ТЕХНОЛОГИЙ»**

**Тема: «Разработка распределённой системы для обеспечения уведомления пользователей о чрезвычайных ситуациях»**

Работу выполнил студент 44 группы Ерюшев И.Д.

Работу проверил канд. техн. наук, доц. каф. ИТ Полупанов А.А.

Краснодар

2024

**1. Введение**

Данное приложение представляет собой систему для работы с клиентами и контрактами. Оно использует ряд облачных технологий от Yandex Cloud для обеспечения надежности, безопасности и масштабируемости. Основные используемые облачные сервисы включают в себя **PostgreSQL**, **Virtual Private Cloud (VPC)**, **Cloud DNS**, **Lockbox**, и **Yandex MetaData Hub**. Каждый из этих сервисов играет важную роль в реализации функционала приложения.

**2. Использование облачных технологий**

**2.1 PostgreSQL**

**PostgreSQL** — это объектно-реляционная система управления базами данных (СУБД), используемая для хранения данных о клиентах и контрактах. В данном приложении PostgreSQL используется для:

* Хранения информации о клиентах (имя, контактные данные, дата создания).
* Хранения данных о контрактах (описание, цена, дата создания).
* Обеспечения запросов к базе данных, таких как добавление, обновление и извлечение данных о клиентах и контрактах.

Yandex Cloud предоставляет управляемую службу PostgreSQL, что гарантирует высокую доступность, автоматическое масштабирование и защиту данных без необходимости вручную управлять инфраструктурой базы данных.

**Преимущества использования PostgreSQL в Yandex Cloud**:

* **Высокая доступность:** Благодаря автоматическим репликациям и восстановлению данных.
* **Масштабируемость:** Легко масштабировать базу данных в зависимости от объема запросов.
* **Надежность и безопасность:** Поддержка транзакций, индексов, а также возможность шифрования данных в покое.

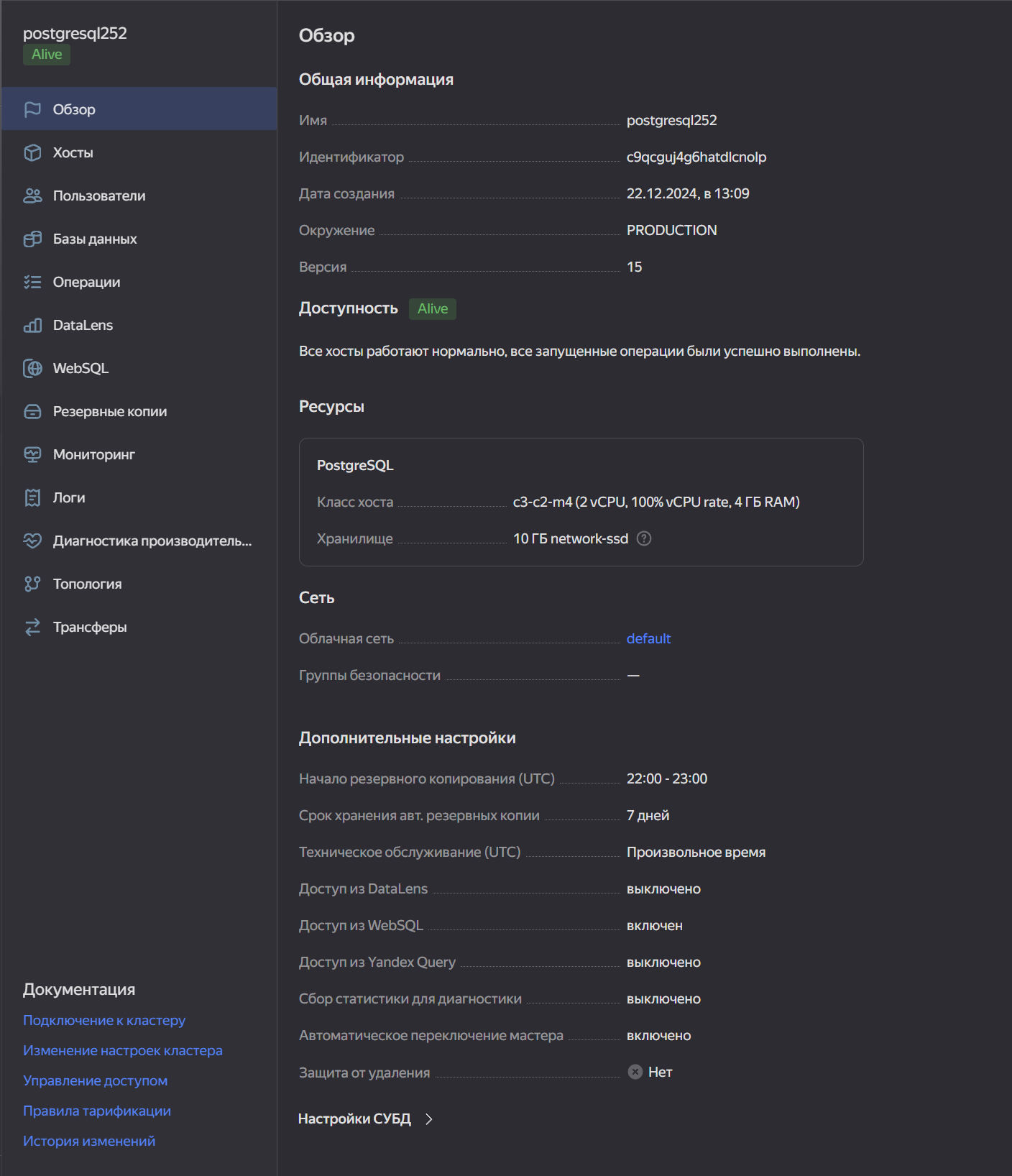


Рисунок 1 – консоль управления

**2.2 Virtual Private Cloud (VPC)**

**Virtual Private Cloud (VPC)** — это виртуальная сеть, которая используется для изоляции и управления сетевыми ресурсами в облаке Yandex Cloud. В данном приложении VPC используется для:

* Обеспечения безопасности и изоляции приложений от внешнего мира.
* Создания защищенных коммуникаций между различными сервисами, такими как база данных и сервер приложения.
* Разграничения доступа к различным частям инфраструктуры, что позволяет управлять политиками безопасности и доступом.

**Преимущества использования VPC**:

* **Изоляция и безопасность:** Возможность контролировать, кто и как может подключаться к сервисам.
* **Настроенная маршрутизация:** Позволяет создать собственные маршруты для передачи данных между различными подсетями.
* **Частные и публичные сети:** Возможность разделить трафик в частные и публичные подсети для улучшения безопасности.

**2.3 Cloud DNS**

**Cloud DNS** — это сервис для управления доменными именами и маршрутизации трафика в облаке. В данном приложении он используется для:

* Управления DNS-записями и обеспечения корректного разрешения доменных имен.
* Обеспечения доступности API и серверов приложения через доменные имена, что позволяет упростить доступ и масштабирование.

**Преимущества использования Cloud DNS**:

* **Высокая производительность:** Обеспечивает низкие задержки и высокую доступность.
* **Автоматизация:** Управление DNS-записями без необходимости вручную конфигурировать серверы.
* **Интеграция с другими сервисами Yandex Cloud:** Простота интеграции с другими сервисами облака, например, с VPC и другими вычислительными ресурсами.

**2.4 Lockbox**

**Lockbox** — это сервис для безопасного хранения и управления конфиденциальными данными, такими как ключи API, пароли, и другие чувствительные данные. В данном приложении Lockbox используется для:

* Хранения секретных данных, таких как ключи для подключения к базе данных или другим сервисам.
* Обеспечения безопасности при работе с конфиденциальной информацией.

**Преимущества использования Lockbox**:

* **Безопасность:** Хранение чувствительных данных в зашифрованном виде.
* **Управление доступом:** Возможность настроить доступ к секретам для определенных пользователей или сервисов.
* **Интеграция с приложением:** Легкая интеграция с приложением через API для получения секретных данных по запросу.

**Yandex MetaData Hub**

**Yandex MetaData Hub** — это сервис для работы с метаданными, который помогает организовать хранение и управление метаинформацией о данных, сервисах и инфраструктуре. В данном приложении MetaData Hub используется для:

* Управления метаинформацией, связанной с клиентами, контрактами и базой данных.
* Организации связей между различными данными и сервисами.

**Преимущества использования Yandex MetaData Hub**:

* **Централизованное управление метаданными:** Упрощает организацию и доступ к метаинформации.
* **Интеграция с другими сервисами Yandex Cloud:** Обеспечивает удобный способ работы с метаданными в рамках всех сервисов облака.

**2.6 WebSQL** WebSQL — это сервис, предоставляющий возможность работы с реляционными базами данных на стороне клиента, который может использоваться для кэширования данных и их быстрого извлечения в веб-приложениях. В данном приложении WebSQL используется для:

* **Локального кэширования данных**: Приложение может использовать WebSQL для временного хранения данных, таких как информация о клиентах и контрактах, которые часто запрашиваются, что позволяет ускорить их извлечение при последующих запросах.
* **Синхронизация с сервером**: Когда пользователь работает с приложением в оффлайн-режиме, данные сохраняются локально в WebSQL и синхронизируются с сервером при следующем подключении.
* **Оптимизация работы с данными**: WebSQL помогает уменьшить нагрузку на сервер и ускорить обработку запросов в случаях, когда база данных на сервере перегружена.

Преимущества использования WebSQL:

* **Высокая скорость работы**: Мгновенное извлечение локально хранимых данных улучшает отклик приложения и сокращает время ожидания для пользователей.
* **Поддержка оффлайн-режима**: WebSQL позволяет работать с данными даже без постоянного подключения к интернету, что улучшает пользовательский опыт в нестабильных сетевых условиях.
* **Минимизация нагрузки на сервер**: Локальное хранение часто запрашиваемых данных снижает частоту обращений к серверу.

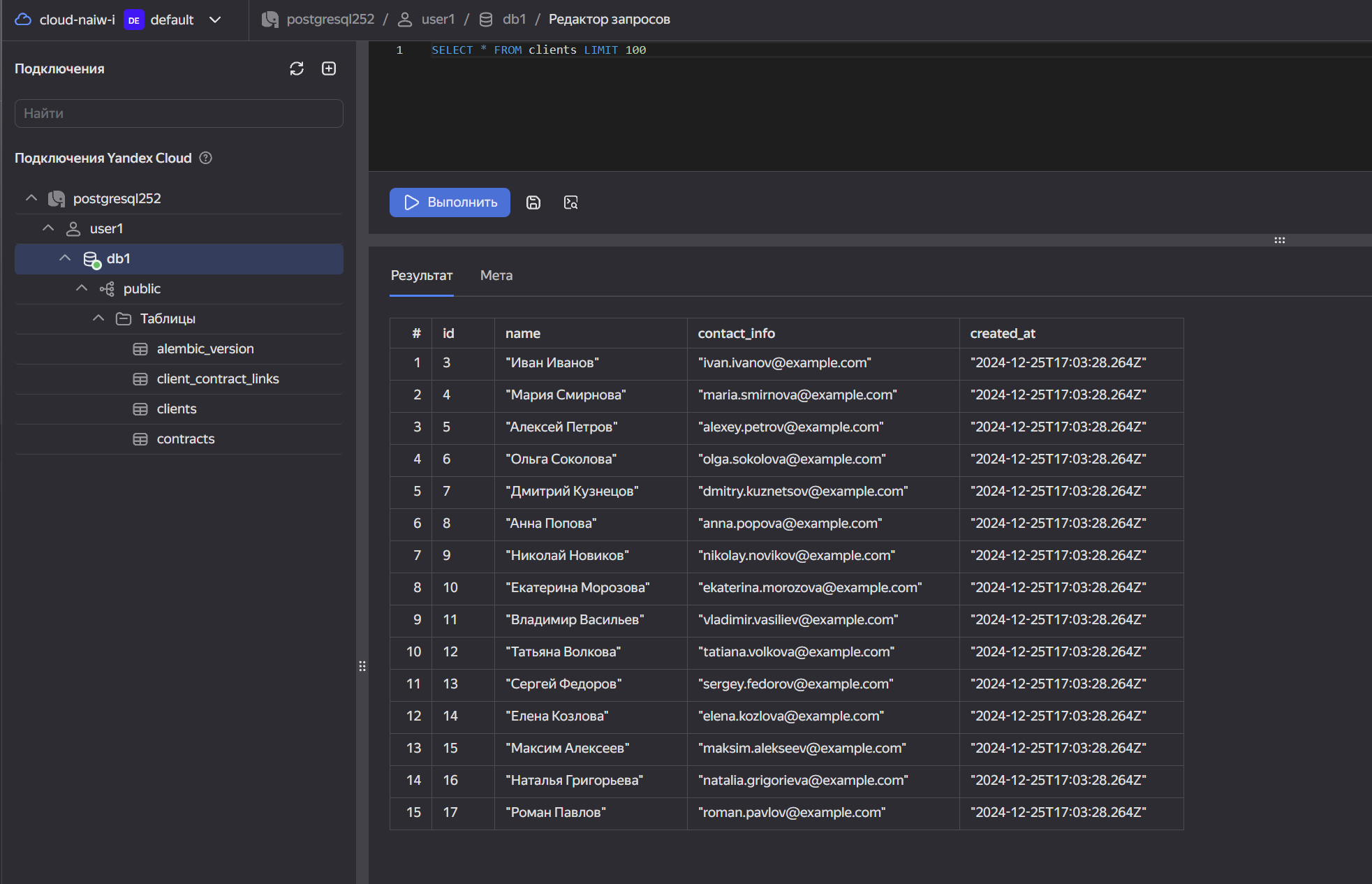


Рисунок 2.1 – окно настройки базы данных WebSQL

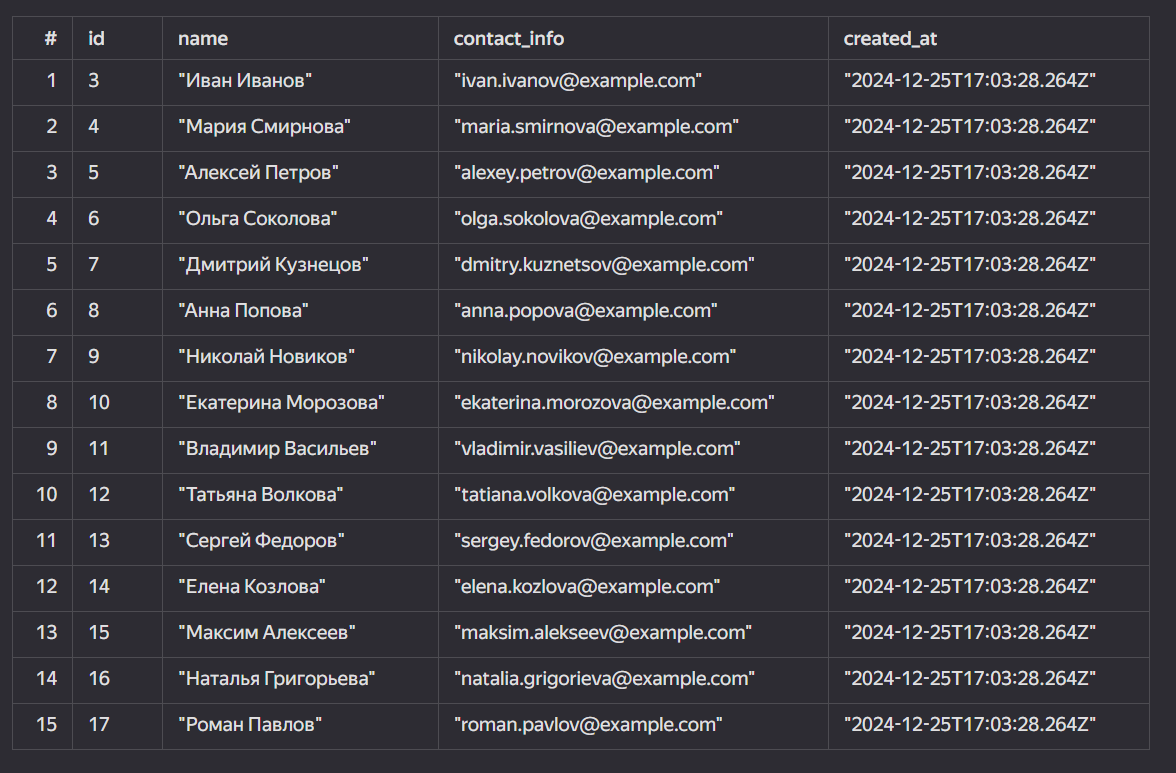


Рисунок 2.2 – Таблица Clients с тестовыми данными

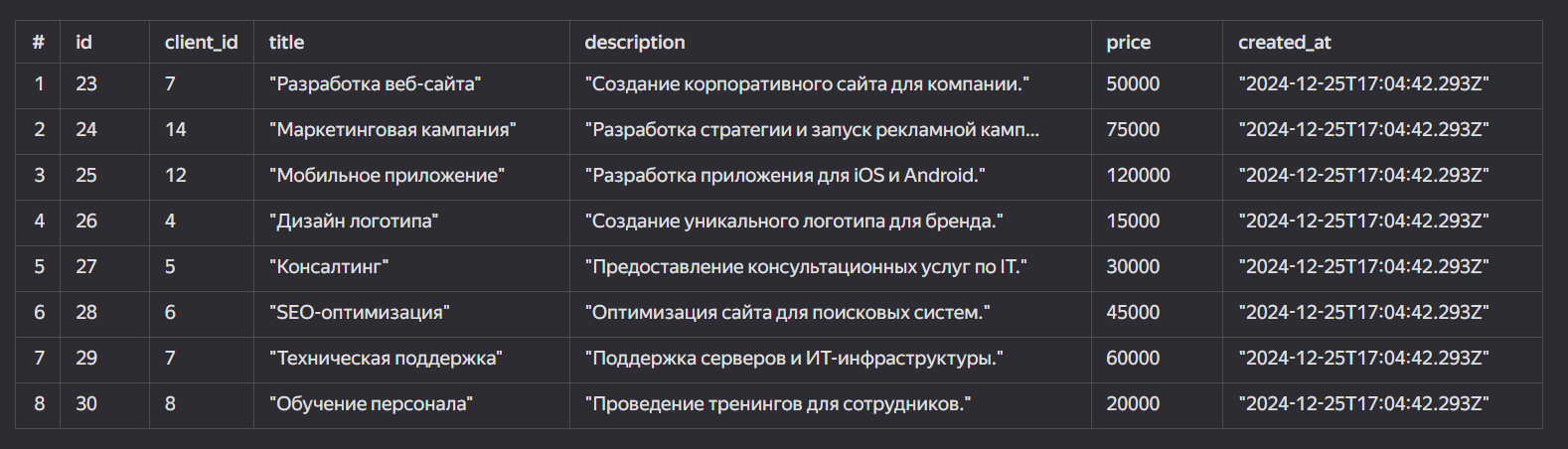


Рисунок 2.3 – Таблица Contracts с тестовыми данными

**2.7 Yandex Monitoring** Yandex Monitoring — это облачный сервис для мониторинга и управления состоянием инфраструктуры и приложений. В данном приложении он используется для:

* **Мониторинга состояния серверов и сервисов**: Yandex Monitoring позволяет отслеживать доступность и производительность серверов, баз данных и других компонентов приложения, что помогает оперативно реагировать на сбои и обеспечивать бесперебойную работу.
* **Сбор метрик и логов**: Приложение собирает данные о производительности, такие как время отклика, количество запросов, использование ресурсов, что позволяет проводить анализ и оптимизировать инфраструктуру.
* **Настройка оповещений**: В случае возникновения проблем, таких как отказ серверов или перегрузка базы данных, сервис автоматически отправляет уведомления, что позволяет быстро реагировать на инциденты и минимизировать время простоя.

Преимущества использования Yandex Monitoring:

* **Проактивный подход к управлению**: Мониторинг в реальном времени помогает предотвратить серьезные проблемы до их возникновения.
* **Гибкость в настройке оповещений**: Возможность настроить различные типы уведомлений для разных уровней серьезности позволяет оперативно реагировать на изменения в инфраструктуре.
* **Интеграция с другими сервисами Yandex Cloud**: Легкость в интеграции с другими сервисами Yandex Cloud, такими как VPC, позволяет получать комплексную информацию о работе всей инфраструктуры и сервисов.

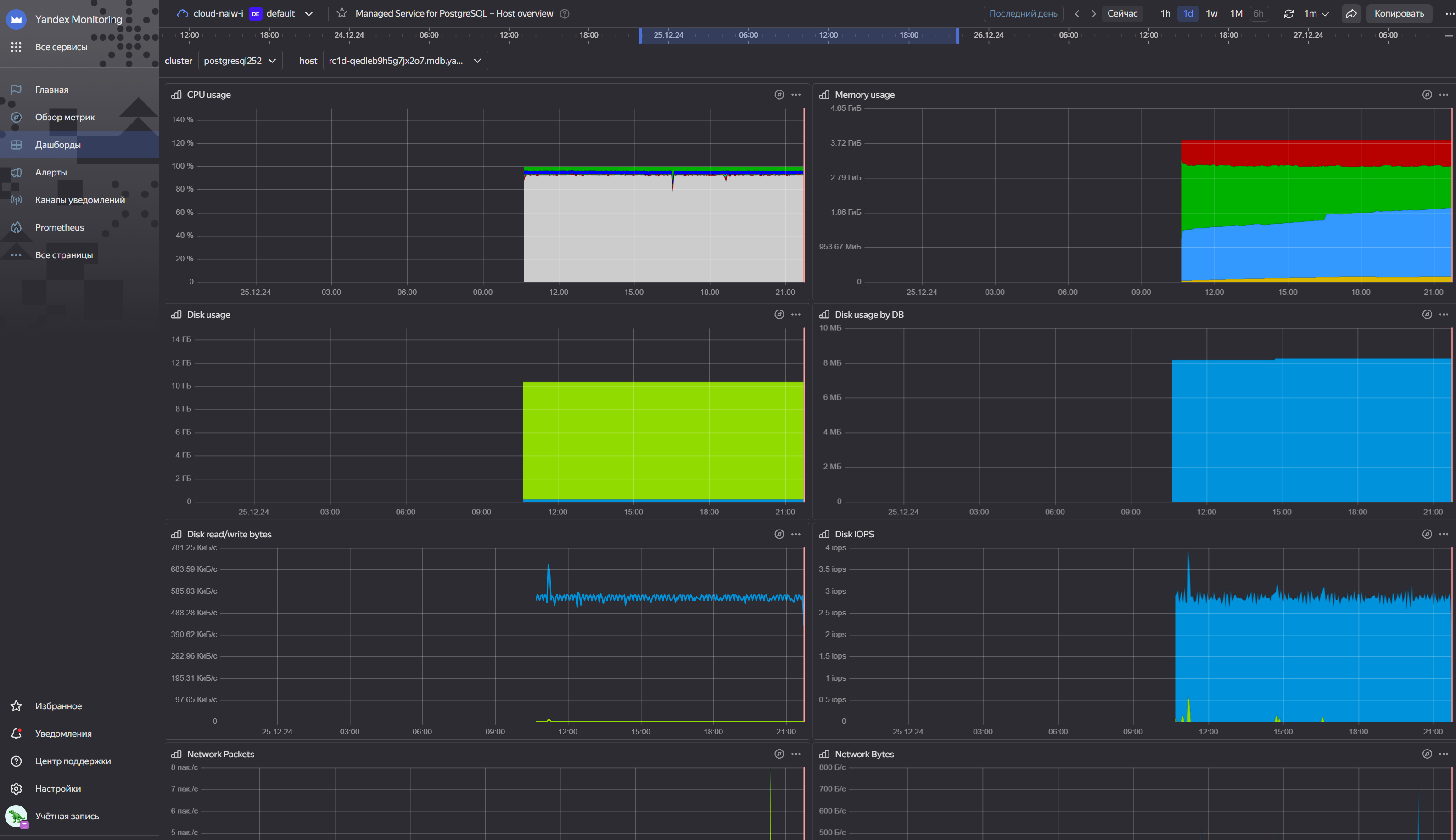


Рисунок 3 – окно мониторинга.

**3. Функционал приложения**

Приложение, описанное в данном отчете, представляет собой систему для эффективного добавления, управления и обработки информации о клиентах и контрактах. Каждый функциональный блок приложения ориентирован на обеспечение удобства работы пользователей, безопасности данных и интеграцию с облачными технологиями для оптимизации масштабируемости и доступности. Рассмотрим ключевые функциональные блоки приложения подробнее:

**3.1 Добавление и управление клиентами**

Добавление и управление клиентами является основным функционалом приложения, которое позволяет не только создавать новые записи о клиентах, но и эффективно их редактировать и искать.

* **Процесс добавления клиента**:
  + Пользователь может добавить нового клиента, предоставив его имя и контактную информацию. Эти данные включают различные контактные данные, такие как номер телефона, адрес электронной почты, физический адрес и другие элементы, которые могут понадобиться для дальнейшего взаимодействия с клиентом.
  + Добавление нового клиента происходит через API, который передает данные на сервер, где они сохраняются в базе данных PostgreSQL.
  + С помощью этих данных создается новая запись в таблице клиентов, а также генерируется уникальный идентификатор (ID) для каждого клиента.
* **Метаинформация о клиентах**:
  + Для каждого клиента в приложении сохраняется не только основная информация (имя, контактные данные), но и метаинформация, которая включает дополнительные атрибуты, такие как дата создания записи, история взаимодействий с клиентом и другие параметры, которые могут быть полезны для фильтрации и поиска клиентов.
  + Метаинформация о клиентах предоставляет гибкость в поиске и аналитике, позволяя отслеживать изменения и взаимодействия, связанные с каждым клиентом.
* **Поиск и фильтрация клиентов**:
  + Приложение позволяет выполнять поиск клиентов по различным критериям, таким как имя, контактная информация или дата последнего обновления. Это достигается с помощью SQL-запросов к базе данных PostgreSQL, которые фильтруют клиентов на основе предоставленных параметров.
  + Благодаря гибкости базы данных PostgreSQL и ее способности эффективно работать с большими объемами данных, поиск клиентов осуществляется быстро, даже при большом количестве записей в базе данных.
* **API для управления клиентами**:
  + Приложение использует API, который позволяет не только добавлять новых клиентов, но и обновлять или удалять существующие записи. Эндпоинты API предоставляют возможность взаимодействовать с клиентскими данными через CRUD-операции (Create, Read, Update, Delete).
  + Все операции с клиентами проходят через защищенные API-запросы, что гарантирует безопасность данных.

**3.2 Добавление и управление контрактами**

Контракты, как неотъемлемая часть взаимодействия с клиентами, занимают важное место в функционале приложения. В этом блоке предусмотрены функции для создания, редактирования и удаления контрактов, а также для их управления.

* **Процесс добавления контракта**:
  + Пользователь может создать новый контракт, связанный с конкретным клиентом. Для каждого контракта необходимо указать такие данные, как название контракта, описание, цена и дата заключения.
  + Контракты хранятся в отдельной таблице базы данных PostgreSQL, где каждый контракт связан с уникальным клиентом по идентификатору клиента.
  + Создание нового контракта также происходит через API, где переданные данные (название, описание, цена) записываются в базу данных.
* **Обновление и удаление контрактов**:
  + Приложение позволяет пользователям обновлять данные контрактов. Например, можно изменить условия контракта, обновить цену или срок действия контракта.
  + Также предусмотрена возможность удаления контракта, если это необходимо (например, если контракт был отменен или выполнен).
  + Эти операции выполняются через API-запросы, которые обновляют или удаляют данные в базе данных PostgreSQL.
* **Связь контрактов с клиентами**:
  + Контракты в приложении всегда связаны с конкретным клиентом. Это реализуется через внешний ключ в базе данных, который указывает на идентификатор клиента. Такой подход позволяет легко отслеживать, какие контракты относятся к конкретному клиенту.
  + Приложение позволяет просматривать все контракты для выбранного клиента, а также выполнять поиск и фильтрацию контрактов по различным параметрам (например, по цене, дате заключения или статусу выполнения).
* **Работа с контрактами через API**:
  + API приложения предоставляет полный набор функционала для работы с контрактами: создание, чтение, обновление и удаление данных. Также через API можно получать информацию о контрактах, связанных с конкретным клиентом, или извлекать список всех активных контрактов.

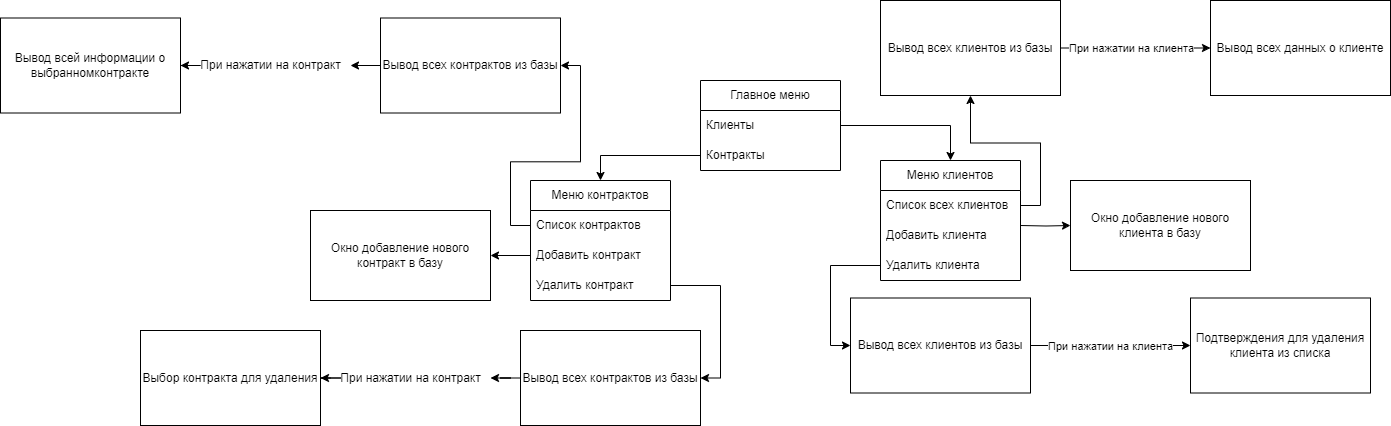


Рисунок 4.1 – схема навигации телеграмм бота

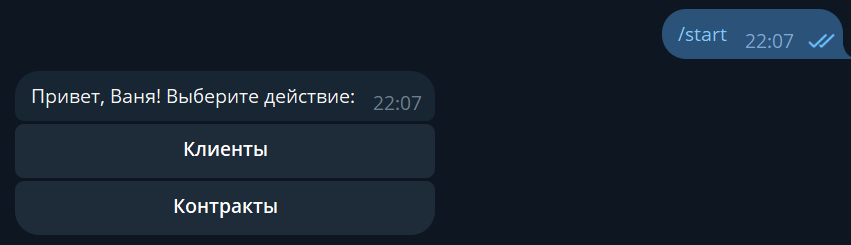


Рисунок 4.2 – главное меню управления

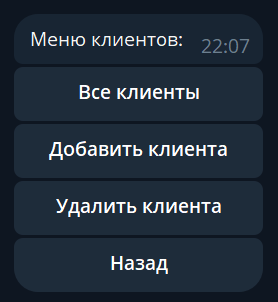


Рисунок 4.1 – Меню клиентов

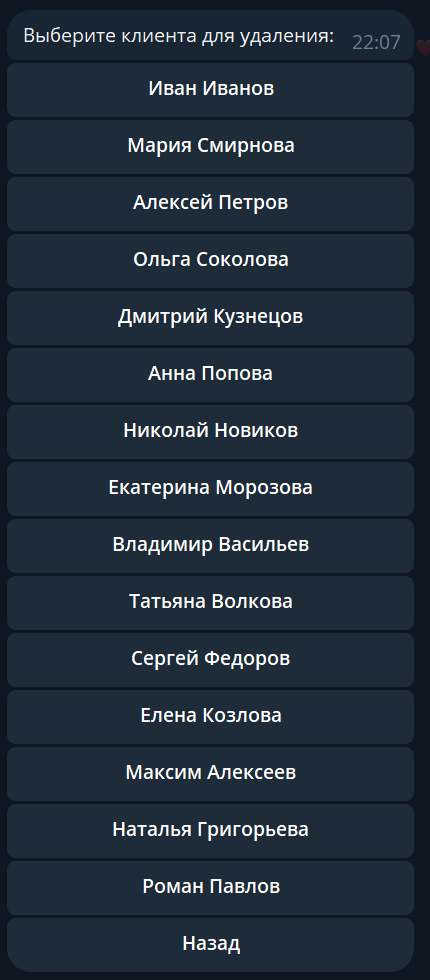


Рисунок 4.3 – меню удаления клиентов

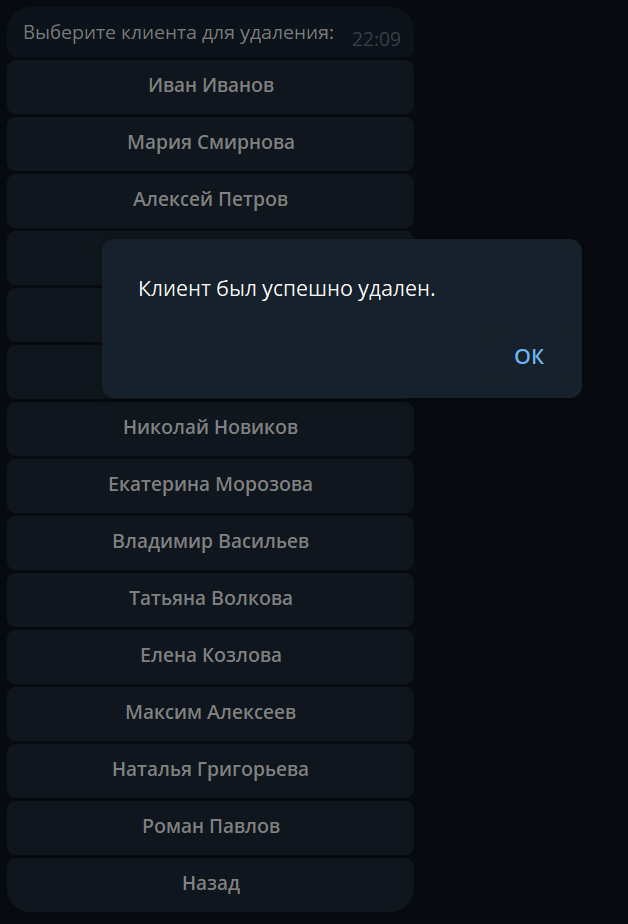


Рисунок 4.4 – Удаление клиента из базы

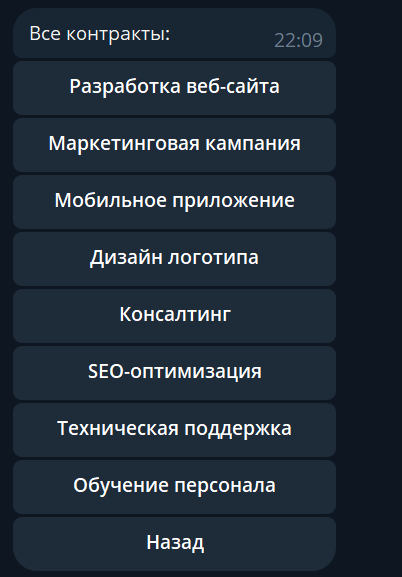


Рисунок 4.5 – Меню контрактов

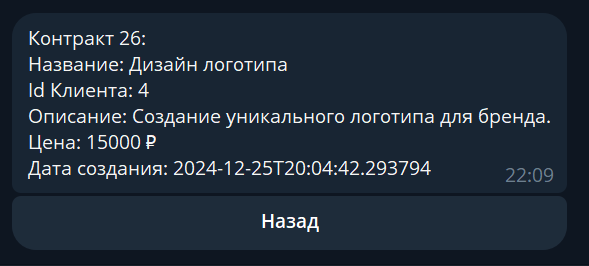


Рисунок 4.6 – Подробная информация о выбранном контракте

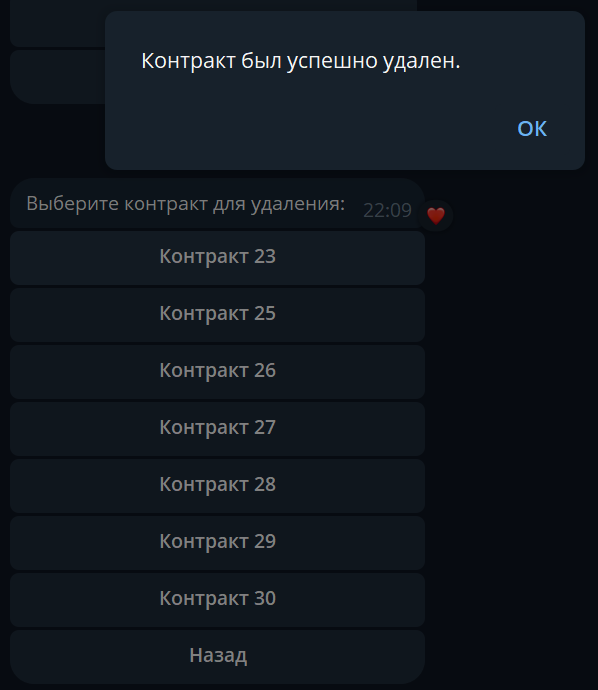


Рисунок 4.7 – Удаления контракта из БД

**3.3 Интеграция с облачными сервисами**

Интеграция с облачными сервисами, использующимися в приложении, представляет собой фундаментальную основу для обеспечения его безопасности, надежности, масштабируемости и высокой производительности. Облачные технологии позволяют хранить данные, управлять ресурсами, гарантировать безопасный доступ к сервисам и масштабировать приложение в зависимости от роста потребностей бизнеса. Давайте более детально рассмотрим все аспекты облачной интеграции, применяемые в этом приложении.

**Использование PostgreSQL для хранения данных**

* **Облачная база данных**:
  + В приложении все данные о клиентах и контрактах сохраняются в **управляемой PostgreSQL базе данных**, которая размещена в **Yandex Cloud**. PostgreSQL, как одна из самых мощных и популярных реляционных систем управления базами данных, используется для хранения, извлечения и обработки всех данных приложения.
  + База данных PostgreSQL гарантирует целостность данных и их доступность, что является критически важным для работы приложения, особенно при обработке больших объемов информации. Она эффективно поддерживает как малые, так и большие рабочие нагрузки.
* **Высокая производительность и надежность**:
  + PostgreSQL в облачной среде Yandex Cloud обеспечивает отличную производительность, которая поддерживает высокие требования к запросам и обработке данных. Это включает выполнение сложных SQL-запросов, агрегирование больших данных и поддержание высокой скорости работы системы при нагрузке.
  + К тому же, облачные сервисы позволяют автоматически распределять нагрузку на несколько серверов, что повышает устойчивость и доступность данных.
* **Поддержка ACID-транзакций**:
  + PostgreSQL полностью поддерживает принципы **ACID** (Atomicity, Consistency, Isolation, Durability), что обеспечивает надежную работу с базой данных и гарантию того, что все транзакции выполняются корректно, а данные остаются в консистентном состоянии, даже в случае сбоя.

**Масштабируемость базы данных**

* **Автоматическое масштабирование**:
  + При размещении PostgreSQL в **Yandex Cloud**, инфраструктура может автоматически масштабироваться в зависимости от текущих потребностей. Если объем данных увеличивается или наблюдается высокая нагрузка на систему, можно без труда расширить вычислительные ресурсы, такие как процессоры и память, что позволяет поддерживать высокую производительность приложения.
  + **Автоматическое масштабирование** в Yandex Cloud помогает избежать перегрузок, предоставляя необходимую мощность по мере роста данных. Система автоматически распределяет запросы между несколькими инстансами, обеспечивая бесперебойную работу.
* **Гибкость при росте данных**:
  + Это важное преимущество, поскольку рост объема данных и числа пользователей может значительно повлиять на производительность, если система не будет масштабироваться. Yandex Cloud поддерживает не только вертикальное масштабирование (увеличение ресурсов на одном сервере), но и горизонтальное масштабирование (распределение нагрузки между несколькими серверами), что позволяет без проблем работать с огромными объемами данных.
  + Масштабирование не требует значительных затрат времени и усилий со стороны пользователей, так как оно происходит автоматически, на основе текущей нагрузки и настроек инфраструктуры.

**Использование Lockbox для безопасности данных**

* **Хранение конфиденциальных данных**:
  + В рамках обеспечения безопасности приложения используется сервис **Lockbox**, предоставляемый Yandex Cloud. Lockbox служит для хранения чувствительных данных, таких как **ключи для подключения к базе данных**, **учетные данные API**, **пароли** и другие параметры, требующие защиты.
  + **Lockbox** гарантирует, что данные хранятся в зашифрованном виде и доступны только авторизованным сервисам и пользователям. Этот сервис значительно снижает риски утечек конфиденциальной информации.
* **Управление доступом**:
  + В **Lockbox** возможно настроить детализированное управление доступом, определяя, какие сервисы и пользователи могут получить доступ к конкретным данным. Для этого применяются политики доступа, которые позволяют ограничить использование чувствительных данных только для тех систем и сервисов, которые действительно нуждаются в этих данных.
  + Lockbox также поддерживает **ротацию ключей** и автоматическую генерацию новых ключей по расписанию, что помогает минимизировать риски безопасности и устаревания данных.
* **Интеграция с приложением**:
  + Приложение может динамически запрашивать секреты из Lockbox в реальном времени, что позволяет избежать хранения конфиденциальной информации непосредственно в коде приложения или на сервере. Таким образом, вся конфиденциальная информация хранится в защищенной среде, доступной только авторизованным компонентам системы.

**Использование Virtual Private Cloud (VPC)**

* **Изоляция и безопасность инфраструктуры**:
  + Вся инфраструктура приложения размещена в **Yandex Cloud** в рамках **Virtual Private Cloud (VPC)**. Это позволяет создать полностью изолированную сеть, в которой взаимодействуют все сервисы приложения, такие как база данных, серверы приложения и другие компоненты.
  + Благодаря VPC ресурсы приложения работают в закрытой сети, что защищает данные от внешних угроз и атак, обеспечивая максимальную безопасность и контроль над данными.
* **Защищенная передача данных**:
  + Через VPC осуществляется **шифрованная передача данных** между различными сервисами и компонентами системы, такими как сервер приложения, база данных и API. Это исключает возможность перехвата данных в процессе их передачи, что критически важно для сохранения конфиденциальности информации.
  + Все соединения внутри VPC имеют строгие правила доступа, что предотвращает несанкционированный доступ и минимизирует риски утечек.
* **Гибкость настроек безопасности**:
  + VPC позволяет гибко настраивать параметры безопасности для каждого ресурса и компонента, например, устанавливать **межсетевые экраны** (firewalls) для блокировки ненужного трафика и подключения только с определенных IP-адресов или сервисов.

**Интеграция с другими сервисами Yandex Cloud**

* **Мониторинг и логирование**:
  + Приложение интегрировано с другими сервисами **Yandex Cloud**, такими как **Yandex Monitoring**, **Yandex Logbroker** и **Yandex Cloud Functions**, для обеспечения постоянного мониторинга работы системы и сбора логов.
  + **Yandex Monitoring** отслеживает производительность и состояние инфраструктуры в реальном времени, автоматически уведомляя администраторов о потенциальных проблемах или аномалиях в системе.
  + **Yandex Logbroker** используется для сбора и анализа логов, что помогает в оперативном обнаружении и устранении ошибок.
* **Обеспечение отказоустойчивости**:
  + Интеграция с **Yandex Cloud** предоставляет доступ к таким возможностям, как **резервное копирование**, **восстановление после сбоев** и автоматическое **перераспределение ресурсов** при сбоях. Это минимизирует время простоя системы и обеспечивает стабильную работу приложения в любой ситуации.
* **Высокая доступность и защита данных**:
  + Взаимодействие между всеми сервисами в Yandex Cloud происходит через защищенные каналы связи, что исключает возможность несанкционированного доступа. Все сервисы настроены для обеспечения **высокой доступности** и **безопасности данных**, что гарантирует, что даже в случае сбоя одного из сервисов, приложение будет продолжать работать без потери данных.

**4. Ссылки**

GitHub репозиторий проекта: https://github.com/Joja2Hub/Bot-BD-Assist