

РУКОВОДСТВО АДМИНИСТРАТОРА

**ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОНВЕРТАЦИИ ЗАПАСОВ УВС ИЗ
КАТЕГОРИЙ РКЗ В КЛАССЫ И ПОДКЛАССЫ РКООН И ОБРАТНО, ИЗ
КАТЕГОРИЙ PRMS В КЛАССЫ И ПОДКЛАССЫ РКООН И ОБРАТНО**

**МОСКВА
2025**

ЦЕЛЬ ДОКУМЕНТА

В документе описываются задачи и способы их выполнения для администраторов, отвечающих за функционирование, установку, настройку и обслуживание бесперебойной работы программного обеспечения для конвертации запасов УВС из категорий РКЗ в классы и подклассы РКООН и обратно, из категорий PRMS в классы и подклассы РКООН и обратно в соответствии с требованиями, определенными в ТЗ.

ЦЕЛЕВАЯ АУДИТОРИЯ:

- Руководитель проекта;
- Администраторы информационной системы;
- Члены проектной команды.

ИСТОРИЯ ИЗМЕНЕНИЙ

ВЕРСИЯ	ДАТА	КОММЕНТАРИИ	АВТОР
1	2	3	4
1	25.09.2024	Создан настоящий документ	Литуновский А.В.
2	26.09.2024	Дополнен документ	Смирнов Д.А.
3	26.09.2024	Дополнен документ	Нароставицына И.В.
4	09.06.2025	Дополнен документ	Нароставицына И.В.

СОДЕРЖАНИЕ

1. Основные понятия и определения	5
2. Назначение Системы.....	6
3. Общие сведения об администрировании Системы.....	7
4. Процедуры установки и настройки	8
4.1. Процедуры установки Системы.....	8
4.1.1. Установка и настройка ОС Linux.....	8
4.1.2. Установка и настройка PostgreSQL	8
4.1.3. Установка Dbeaver.....	9
4.1.4. Подключение базы данных PostgreSQL	9
4.1.5. Установка Nginx	11
4.1.6. Установка базовых программных компонентов	12
4.2. Процедуры настройки Системы.....	12
4.2.1. Подготовка к работе	12
4.2.2. Инсталляция и настройка	13
5. Мероприятия по обеспечению резервного копирования и восстановления Системы.....	21
5.1. Резервное копирование Системы.....	21
5.2. Восстановление Системы из резервных копий	21
5.3. Тестирование Системы после восстановления	21
5.4. Настройка архивации журналов событий Системы	21
6. Приложение А. Техническое обеспечение проведения испытаний.....	23

1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

В таблице 1 представлены основные понятия и определения.

Таблица 1. Термины и сокращения

№	Термины, сокращения	Определение
1	2	3
1.	АРМ	Автоматизированное рабочее место
2.	БД	База данных
3.	ЕОЛ	Единственное ответственное лицо
4.	ИБ	Информационная безопасность
5.	ИТ	Информационные технологии
6.	НК	Нефтяная компания
7.	ОС	Операционная система
8.	ПАО	Публичное акционерное общество
9.	ПКМ	Правая кнопка мыши
10.	ПО	Программное обеспечение
11.	ППО	Продукт программного обеспечения
12.	РКЗ	Российская классификация запасов
13.	РОН	Реестр отечественных номеров
14.	РКООН	Рамочная классификация ресурсов Организации Объединенных Наций 2019 года
15.	Конвертер запасов, Система	Программное обеспечение для конвертации запасов УВС из категорий РКЗ в классы и подклассы РКООН и обратно, из категорий PRMS в классы и подклассы РКООН и обратно.
16.	СБ	Служба безопасности
17.	СМН	Система Мониторинга Недропользования
18.	СУБД	Система управления базами данных
19.	ТЗ	Техническое задание
20.	УВС	Углеводородное сырье
21.	PRMS	Система управления ресурсами углеводородов (Petroleum Resources Management System, PRMS). Подготовлена при участии: Общества инженеров-нефтяников (SPE), Всемирного нефтяного совета (WPC), Американской ассоциации геологов-нефтяников (AAPG), Общества инженеров по оценке запасов нефти и газа (SPEE), Общества геофизиков-разведчиков (SEG), Общества петрофизиков и интерпретаторов каротажных диаграмм (SPWLA), Европейской ассоциации геологов и инженеров (EAGE). Версия 1.01. ISBN 978-1-61399-660-7

2. НАЗНАЧЕНИЕ СИСТЕМЫ

Проектируемая Система автоматизирует процесс конвертации запасов и ресурсов УВС из категорий РКЗ в классы и подклассы РКООН и обратно, из категорий PRMS в классы и подклассы РКООН и обратно.

Создание Конвертера запасов выполняется в целях:

- обеспечение сопоставимости оценок запасов и ресурсов УВС;
- снижение влияния субъективности при оценке запасов УВС;
- повышение интенсификации внутрироссийского и международного сотрудничества в сфере недропользования.

Данная инструкция предназначена для технических специалистов (администраторов Системы), обеспечивающих бесперебойную работу Системы.

3. ОБЩИЕ СВЕДЕНИЯ ОБ АДМИНИСТРИРОВАНИИ СИСТЕМЫ

Для Системы определены следующие привилегированные роли:

- Администратор;
- Техническая поддержка.

Обязанности и зоны ответственности каждой роли приведены далее.

Основными обязанностями Администратора являются:

- установка, обновление, настройка, мониторинг работоспособности, параметров СУБД;
- оптимизация функционирования СУБД;
- настройка механизмов резервного копирования и восстановления БД Системы из резервной копии средствами СУБД;
- резервное копирование БД вручную по требованию;
- проверка восстановления СУБД после восстановления из резервной копии;
- поддержка текущей конфигурации БД;
- проведение регламентных работ на серверах Системы;
- установка, обновление, настройка Системы;
- установка, обновление, настройка прикладного ПО Системы;
- решение инцидентов при работе с Системой;
- взаимодействие с разработчиками по вопросам технического обслуживания Системы;
- консультация пользователей по техническим аспектам функционирования Системы;
- выполнение заявок на создание/блокировку учётных записей пользователей;
- управление правами доступа пользователей (назначение ролей);
- рассылка информационных писем и уведомлений пользователям Системы;
- ведение учетных записей пользователей Системы.

Основными обязанностями роли Техническая поддержка являются:

- решение инцидентов при работе с Системой;
- техническое обслуживание Системы;
- анализ данных и действий пользователей для решения возникших инцидентов.

Основные ограничения и ответственность:

- к работе с Системой допускаются пользователи, ознакомленные с настоящей Инструкцией.
- запрещается подключение к Системе с использованием учетной записи другого пользователя или/и сервисных учетных записей.

4. ПРОЦЕДУРЫ УСТАНОВКИ И НАСТРОЙКИ

4.1. Процедуры установки Системы

Для установки и запуска Системы Администратору следует выполнить для установки ПО определённую последовательность действий на виртуальных серверах.

4.1.1. Установка и настройка ОС Linux

Для установки и настройки ОС Linux необходимо:

1. Установить и настроить ОС Linux необходимо как на сервере приложений, так и на сервере СУБД. Выбор конкретного дистрибутива остаётся на усмотрение системного администратора. Система тестировалась на дистрибутиве Linux Ubuntu 22.04.3 LTS.
2. Системный администратор выполняет установку ОС Linux в соответствии с официальным руководством по установке и развертыванию соответствующего дистрибутива.
3. После установки ОС Linux необходимо установить все актуальные обновления.

4.1.2. Установка и настройка PostgreSQL

В качестве системы управления данными в Системе используется СУБД PostgreSQL. Способ установки зависит от конкретной сборки ОС Linux. В данной инструкции рассмотрена сборка Linux Ubuntu 22.

В репозиториях Ubuntu уже включена PostgreSQL. Развертывание выполняется командой apt. До загрузки PostgreSQL необходимо обновить список пакетов (Рисунок 1).

```
sudo apt update && sudo apt upgrade -y
```

Рисунок 1. Команда обновления списка пакетов

Далее необходимо загрузить PostgreSQL с утилитой contrib (Рисунок 2).

```
sudo apt update && sudo apt upgrade -y
```

Рисунок 2. Команда загрузки PostgreSQL

Загрузятся драйверы PostgreSQL последней версии и развернутся необходимые компоненты на виртуальной машине с Ubuntu. Далее необходимо запустить соответствующий сервис (Рисунок 3).

```
sudo apt update && sudo apt upgrade -y
```

Рисунок 3. Запуск сервиса PostgreSQL

PostgreSQL применяет термин «Роль». Практически это тот же аккаунт в Ubuntu. При запуске СУБД роли сервиса привязываются к одноименным аккаунтам в Unix-системах. Другими словами, при наличии роли в PostgreSQL, войти в СУБД можно с аккаунтом Ubuntu. При запуске СУБД генерируется аккаунт postgres, привязываемый к роли PostgreSQL.

Необходимо войти в аккаунт postgres с sudo (Рисунок 4). Если в момент настройки выполнен вход в аккаунте postgres, нужно выйти, набрав exit. В этом варианте перейдем в аккаунт postgres с sudo.

```
sudo -u postgres psql
```

Рисунок 4. Вход в PostgreSQL

Аккаунт postgres обладает правами администратора. Необходимо написать «createuser», эта команда сообщает, что добавляется новая роль. Чтобы указать имя роли и выдать суперюзера, применяется флаг – «interactive» (Рисунок 5).

```
sudo -u postgres createuser --interactive
```

Рисунок 5. Создание пользователя

Любому созданному аккаунту привязывается БД с идентичным именем, то есть созданный tester начнет подключаться к БД tester. Командой «createdb» добавляется БД (поднять новую БД PostgreSQL на Ubuntu), назвав ее tester (Рисунок 6).

```
sudo -u postgres createdb tester
```

Рисунок 6. Создание базы данных

4.1.3. Установка Dbeaver

Для конфигурирования, управления и администрирования БД PostgreSQL с локального компьютера, как правило, используется утилита Dbeaver. Для того чтобы проверить ее наличие, необходимо открыть: «Пуск» («Start») - «Панель управления» («Control panel») - «Программы и компоненты» («Programs and Features») и найти ее в списке установленного на клиентском компьютере администратора ПО. Утилита Dbeaver распространяется бесплатно и доступна для скачивания с официального сайта <https://dbeaver.io/>.

4.1.4. Подключение базы данных PostgreSQL

Для подключения БД необходимо:

1. Запустить «Dbeaver», пройдя по пути «Пуск» – «Dbeaver» и подключиться к серверу БД (Рисунок 7).

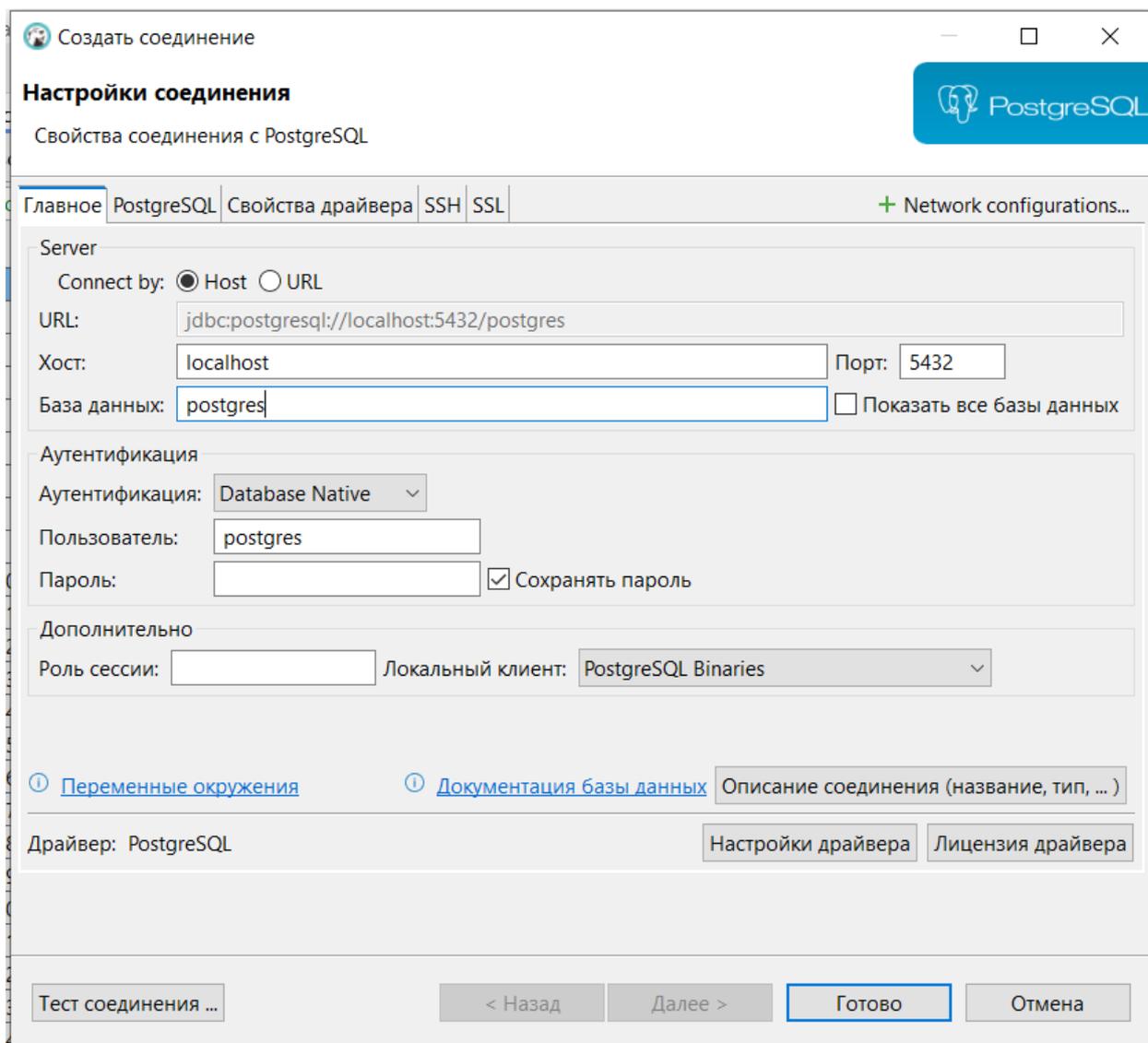


Рисунок 7. Подключение к серверу БД

2. Нажать ПКМ на папке «Connections» и в контекстном меню выбрать «Создать соединение...» (Рисунок 8).
3. В открывшемся окне указать параметры БД и подключиться к ней.
4. Таблицы БД для работы Системы создаются автоматически при первом запуске веб-интерфейса, создавать их отдельно не нужно.

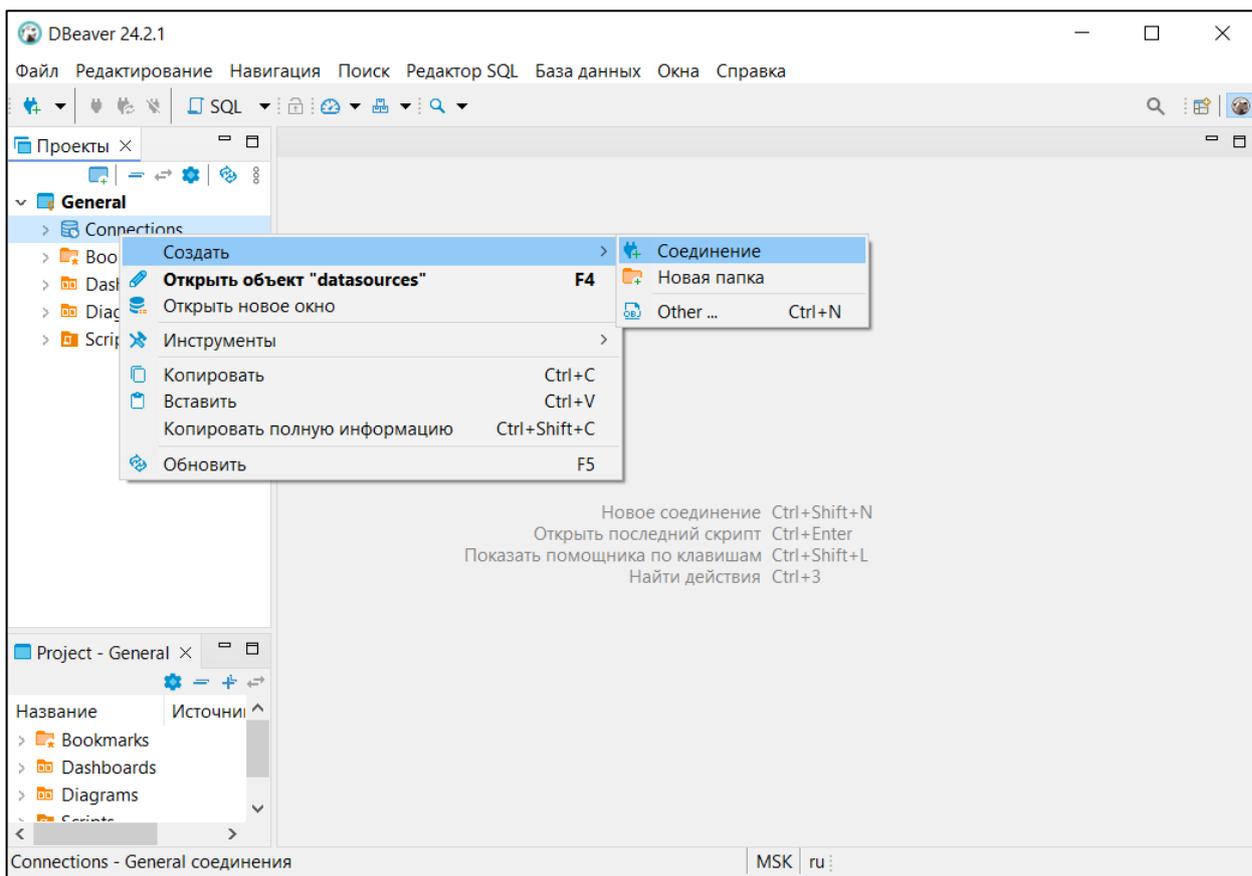


Рисунок 8. Подключение БД

4.1.5. Установка Nginx

Веб-сервер Nginx в Системе используется для хранения клиентского приложения, а также сертификатов HTTPS. Установить и настроить Nginx необходимо на сервере приложений.

Необходимо зайти на сервер приложений в терминале Linux выполнить команду установки Nginx из репозитория приложений (Рисунок 9). В случае отсутствия Nginx в репозитории рекомендуется установить его по руководству с официального сайта (<https://nginx.org/ru/docs/install.html>). Здесь и далее приведён пример установки Nginx для сборки Linux Ubuntu 22.

```
sudo apt update && sudo apt install nginx
```

Рисунок 9. Установка Nginx

Для инсталляции необходимо войти в систему учётной записью, имеющей привилегию на запуск команды «sudo» и последовательно выполнить определённые команды.

Первая команда выполняет обновление БД доступных для установки пакетов. Вторая выполняет установку элементов веб-сервера Nginx.

При настройке веб-сервера Nginx, первым делом следует уделить внимание брандмауэру, используемому ОС вашего сервера. Для защиты подключений к

Системе, с точки зрения простоты использования и надёжности, рекомендуется использовать инструмент Uncomplicated Firewall (UFW) (Рисунок 10).

```
sudo ufw allow 'Nginx Full'
```

Рисунок 10. Установка Nginx правил для UFW

4.1.6. Установка базовых программных компонентов

В установку базовых программных компонентов входит:

- Установка Docker. В большинстве современных сборок ОС Linux в репозиториях присутствует Docker. Например, в Ubuntu 22.04 для его установки необходимо запустить команду «apt install docker.io».
- Установка Node.js. В большинстве современных сборок ОС Linux в репозиториях присутствует Node.js. Например, в Ubuntu 22.04 для его установки необходимо запустить команду «apt install nodejs». Пакет nodejs необходим для сборки клиентской части Системы.

4.2. Процедуры настройки Системы

4.2.1. Подготовка к работе

Типовые операции администратора Системы можно объединить в следующие группы:

- подготовка к установке;
- инсталляция и настройка ПО;
- администрирование Системы.

Для обеспечения настройки политики безопасности и контроля доступа пользователей к Системе, функциям, инструментам, отчетам, данным и справочникам используется модуль «Администрирование». Разграничение прав доступа осуществляется в рамках служебных ролей. Пользователи, входящие в служебную роль «Администратор», могут создавать, редактировать и контролировать новые роли, а также добавлять или исключать из них пользователей Системы. Роль «Администратор» назначается привилегированному пользователю «Администратор ППО» с полным доступом к Системе. Пользователи, входящие в служебную роль «Аудитор» назначается привилегированным пользователям: «Администратор ИБ», «Контроль СБ», «Контроль ИБ» с правами для чтения к функциям, инструментам, отчетам и данным Системы.

Процесс установки и настройки включает в себя следующий порядок действий:

1. Подготовка инфраструктуры – установка базового ПО, заявленного в технических требованиях к Системе.
2. Подключение БД.
3. Копирование бинарных и конфигурационных файлов приложения.

4. Публикация веб-приложения.

В установочный пакет входят следующие компоненты:

- «ConvertReserves» – папка, содержащая исполняемые файлы серверной части и конфигурационные файлы Системы;
- «Client» – папка, содержащая исполняемые файлы клиентской части по технологии React и конфигурационные файлы Системы.

Технические требования к серверам для установки Системы приведены в приложении А.

4.2.2. Инсталляция и настройка

4.2.2.1. Настройка файлов конфигурации

На рисунке 11 представлена структура основных папок и файлов Системы. В папке «SGP.ConvertReserves.WebApi» находятся конфигурационные файлы Системы. Для того чтобы настроить проект, потребуется внести изменения в конфигурационные файлы «appsettings.Production.json» и «index.tsx».

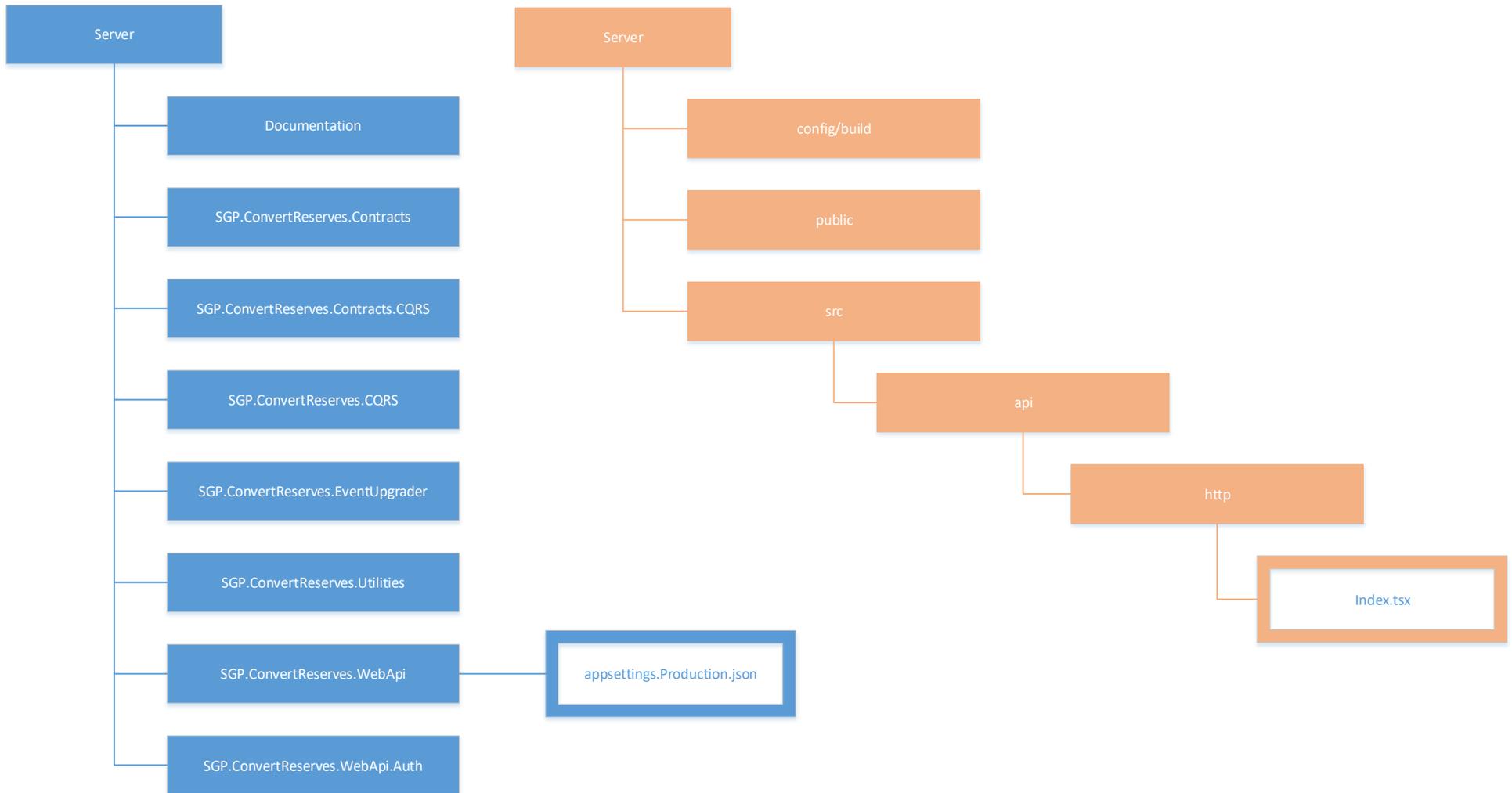


Рисунок 11. Структура файлов и папок проекта и пути к конфигурационным файлам

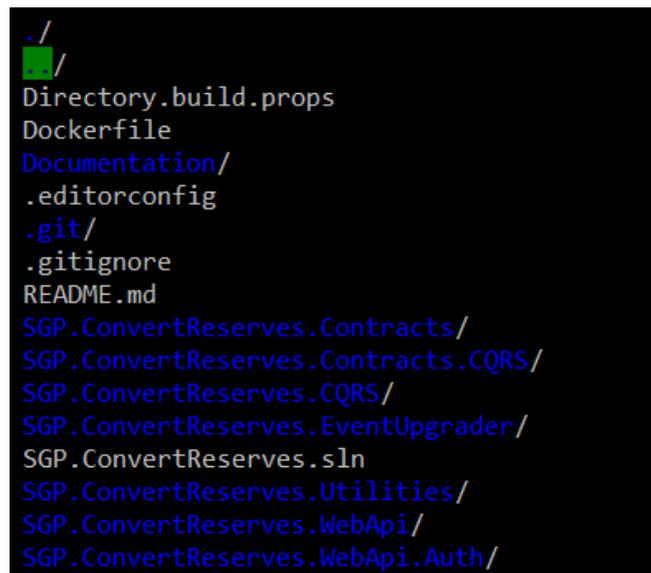
4.2.2.2. Настройка конфигурационных файлов серверной части проекта

Важно! Для редактирования конфигурационных файлов нельзя применять текстовые редакторы Word (word.exe) и WordPad (wordpad.exe), предпочтительный редактор – Блокнот (notepad.exe) в случае использования утилиты WinSCP, либо редактор nano в случае работы через терминал Linux.

Для корректной работы приложения на сервере приложений необходимо в проекте с исходным кодом указать следующие параметры:

- адрес сервера СУБД, на котором установлен PostgreSQL;
- порт, на котором работает PostgreSQL;
- название БД СУБД (по умолчанию converter);
- имя пользователя СУБД, который является владельцем БД;
- пароль соответствующего пользователя СУБД.

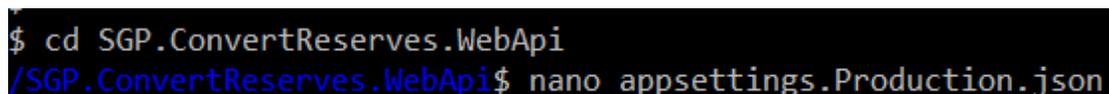
Здесь и далее в качестве примера используется работа с редактором nano терминала Linux. Для его использования сначала необходимо загрузить исходные коды на сервер приложений в папку, доступную для Docker. Структура проекта показана на рисунке 12.



```
./
./
Directory.build.props
Dockerfile
Documentation/
.editorconfig
.git/
.gitignore
README.md
SGP.ConvertReserves.Contracts/
SGP.ConvertReserves.Contracts.CQRS/
SGP.ConvertReserves.CQRS/
SGP.ConvertReserves.EventUpgrader/
SGP.ConvertReserves.sln
SGP.ConvertReserves.Utilities/
SGP.ConvertReserves.WebApi/
SGP.ConvertReserves.WebApi.Auth/
```

Рисунок 12. Структура проекта

С помощью команды «cd» необходимо перейти в директорию «SGP.ConvertReserves.WebApi». Затем открыть в редакторе nano файл «appsettings.Production.json» (Рисунок 13).



```
$ cd SGP.ConvertReserves.WebApi
/SGP.ConvertReserves.WebApi$ nano appsettings.Production.json
```

Рисунок 13. Запуск редактирования настроечного файла

В случае успешного открытия файла, его содержимое должно отобразиться на экране.

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "ConnectionStrings": {
    "Connection": "User ID=...;Password=...;Server=...;Port=...;Database=...;Integrated Security=true;Pooling=true;"
  },
  "Jwt": {
    "Key": "12345678901234567890",
    "Issuer": "convert-reserves-app"
  }
}

```

Рисунок 14. Настроечный файл приложения

Настроечный файл разбит на следующие блоки (Рисунок 14):

- "Logging" – содержит параметры логирования;
- "Logging" – содержит подключения к БД, в данном случае единственное подключение, которое необходимо отредактировать в своём параметре "Connection"
- "Jwt" – настроечную информацию для формирования jwt-токенов при аутентификации и авторизации

В блоке "Logging" в параметре "Logging" записана строка подключения к БД. В примере выше вместо троеточия необходимо указать реальные данные, касающиеся сервера СУБД. Описание параметров строки подключения к БД представлено в таблице 2.

Таблица 2. Описание параметров строки подключения к БД

Параметр	Описание	Предполагаемое значение
User ID	Имя пользователя СУБД, который является владельцем БД	-
Password	Пароль пользователя СУБД	-
Server	Адрес сервера СУБД (IP-адрес, или доменное имя)	-
Port	Порт, на котором функционирует СУБД PostgreSQL на сервере СУБД	5432
Integrated Security	Устанавливает проверку подлинности	true
Pooling	Регулирует объединение подключений в пул	true

4.2.2.3. Публикация веб-приложения средствами Docker

Публикация приложения на сервере приложений осуществляется средствами Docker. При этом, публикацию можно разделить на несколько этапов:

1. Загрузка исходных файлов на сервер приложения.
2. Сборка Docker-образа.
3. Запуск Docker-контейнера на основе собранного Docker-образа.

Для загрузки исходных файлов на сервер приложения рекомендуется использовать утилиту WinSCP (<https://winscp.net/eng/download.php>). Файлы с исходным кодом необходимо положить в папку, доступную для Docker. Альтернативным решением является загрузка исходников в доступный с сервера приложений репозиторий Gitlab и последующая загрузка их оттуда на сервер приложений.

Для сборки Docker-образа необходимо через терминал Linux зайти в папку с исходными кодами и в ней запустить команду «`docker build -t converter-image .`» (Рисунок 15).

```
docker build -t converter-image .
```

Рисунок 15. Команда сборки образа

В результате работы команды должен быть создан образ «`converter-image`» и помещён в локальный репозиторий образов на сервере приложений. Если всё в порядке, то при использовании команды «`docker images ls`» (Рисунок 16) можно будет увидеть название созданного образа.

```
docker images ls
```

Рисунок 16. Команда для вывода списка образов

Запуск контейнера на основе созданного образа можно осуществлять из любого места на сервере приложений, не обязательно из папки с исходниками. Перед запуском контейнера необходимо определиться с портом, которому он будет сопоставлен. Порт должен быть открыт для HTTP и HTTPS-запросов. Здесь и далее в качестве примера используется порт 8083. Для запуска контейнера необходимо в терминале Linux запустить команду «`docker run --rm -d -p 8083:80 -e "ASPNETCORE_ENVIRONMENT=Production" --name converter converter-image`» (Рисунок 17).

```
docker run --rm -d -p 8083:80 -e "ASPNETCORE_ENVIRONMENT=Production" --name converter converter-image_
```

Рисунок 17. Команда запуска контейнера

Для проверки факта запуска контейнера необходимо использовать команду «`docker ps`». В случае, если контейнер запущен, его название отобразится в соответствующем списке.

4.2.2.4. Публикация клиентской части веб-приложения

Клиентская часть Системы представляет собой приложение на ReactJS, для его запуска потребуется Nginx, а для его сборки Node.js.

Рекомендуется вначале запустить серверную часть Системы, чтобы точно понимать на каком порту развёрнут соответствующий ей контейнер. Перед сборкой клиентской части необходимо указать путь к уже действующей серверной части. Для этого следует отредактировать файл index.tsx (Рисунок 18). Путь к действующей серверной части необходимо прописать в переменную «API_PROD_URL».

```
/* eslint-disable @typescript-eslint/no-explicit-any */
import axios, { AxiosError } from "axios";
import { dictionary } from "@dictionary";
import { useAuthStore } from "@store/auth/authStore";
import i18n from "@i18n/config";
import { ITokenModel } from "@api/generated/data-contracts";
import { EventEmitter } from "@utils/eventEmitter";
import { useErrorManager } from "@utils/useErrorManager";

const { LS } = dictionary;

const API_DEV_URL = "https://dev-api-converter.sgp72.ru";
const API_PROD_URL = "https://api-converter.sgp72.ru";
```

Рисунок 18. Файл index.tsx

Для сборки приложения необходимо средствами терминала Linux зайти в папку с исходниками приложения и выполнить команду сборки «npm build:deploy-prod» (Рисунок 19).

```
npm build:deploy-prod
```

Рисунок 19. Команда сборки клиентского приложения

В результате работы команды в той же папке появится папка «build», где и будет находиться сборка клиентского приложения, готовая к работе. Далее, содержимое папки «build» необходимо поместить в папку «/var/www/html», откуда Nginx берёт файлы для запуска основного сайта.

4.2.2.5. Log-файлы Системы

Система регистрирует внутренние события с помощью стандартных средств Docker, а также средств администратора, предоставляемых самим приложением внутри контейнера.

Docker регистрирует внутренние события приложения в log-файлах, которые по умолчанию располагаются на системном диске сервера приложений по пути: «/var/lib/docker/containers». Другим способом доступа к логам является использование команды docker logs.

События Системы, которые регистрирует само приложение, хранятся в БД на сервере СУБД. Просмотр данных событий доступен администратору в интерфейсе приложения (Рисунок 20).

Дата	Пользователь	Событие	Идентификатор объекта	Объект	Параметры	Сообщения об ошибке
30.05.2025, 10:08:59	Vera-do@mail.ru	добавление нового проекта	cfcc9a62-00c5-4edf-9925-bf5117246a7		Имя проекта: Новый проект конвертации	
30.05.2025, 10:08:52	Vera-do@mail.ru	просмотр проекта	010f54e0-87af-4f97-9ad2-8c3681d55291			
30.05.2025, 10:08:49	Vera-do@mail.ru	редактирование проекта	65773114-d352-45c0-ad67-973774abe34b		Новое имя проекта: Проект-1 Новое тип проекта: РКЗ	
30.05.2025, 10:08:47	Vera-do@mail.ru	редактирование проекта	33af5383-d52d-4601-b68e-879b755d8846		Новое имя проекта: Проект Новое тип проекта: РКЗ	
30.05.2025, 10:08:45	Vera-do@mail.ru	редактирование проекта	2a4b74a5-a18e-43be-9bb7-f66fd2c1979		Новое имя проекта: Новая групп Новое тип проекта: РКЗ	
30.05.2025, 10:08:39	Vera-do@mail.ru	редактирование проекта	1278fb0c-3562-479e-b81c-6b6b7a73e342		Новое имя проекта: Проект Новое тип проекта: РКЗ	
30.05.2025, 10:08:33	Vera-do@mail.ru	просмотр объекта конвертации	85f37cec-8b9e-4fce-a6d1-ae22c65510c0			
30.05.2025, 10:08:33	Vera-do@mail.ru	просмотр проекта	ebac9f5e-1b6a-4d80-9dcd-f150c769023b		ID: d02ce3a8-6783-46f2-8f73-bd99be455ca7 Название: Новый проект конвертации	
30.05.2025, 10:08:33	Vera-do@mail.ru	просмотр объекта конвертации	084467b6-ab05-421b-ab30-4d9bba2ad784			
30.05.2025, 10:08:33	Vera-do@mail.ru	просмотр проекта	69fdb5fe-3154-414e-aa49-3e8fb82fbb7e			
30.05.2025, 10:08:33	Vera-do@mail.ru	добавление нового проекта	5259ed7e-e2ef-42ad-8b7b-80973c0ca205		Имя проекта: Новый проект конвертации	
30.05.2025, 10:08:06	Vera-do@mail.ru	просмотр проекта	b3634895-cda1-4183-9f2c-d31d0a1257b5			
30.05.2025, 10:08:04	Vera-do@mail.ru	просмотр объекта конвертации	3038abb9-63ce-43c1-845b-208ce5c656a6			

Рисунок 20. Форма просмотра событий Системы

4.2.2.6. Установка Системы на клиентское рабочее место пользователя

Для того чтобы пользователь смог осуществить вход в Систему необходимо:

1. На компьютере пользователя установить любой браузер из списка: Internet Explorer (версия 9 и выше), Mozilla Firefox (версия 5 и выше), Google Chrome (версия 10 и выше), Apple Safari (версия 5 и выше).
2. Создать учетную запись пользователя в модуле «Администрирование» согласно инструкции, из раздела «Администрирование Системы».
3. На рабочем месте пользователя запустить браузер и перейти по ссылке: «https://имя_сервера_приложений/login». Откроется форма авторизации пользователя (Рисунок 21).

Конвертер запасов

Авторизация

Почта
Введите почту

Пароль
Введите пароль

Запомнить меня [Забыли пароль?](#)

Войти

Или

[Запросить доступ](#) [Войти как Гость](#)

Рисунок 21. Форма авторизации

Пользователю необходимо ввести в открывшейся форме свой логин и пароль и нажать на кнопку «Вход».

Если пользователь ввел действительные параметры учетной записи, будет выполнен вход в Систему, и откроется главная страница приложения (Рисунок 22).

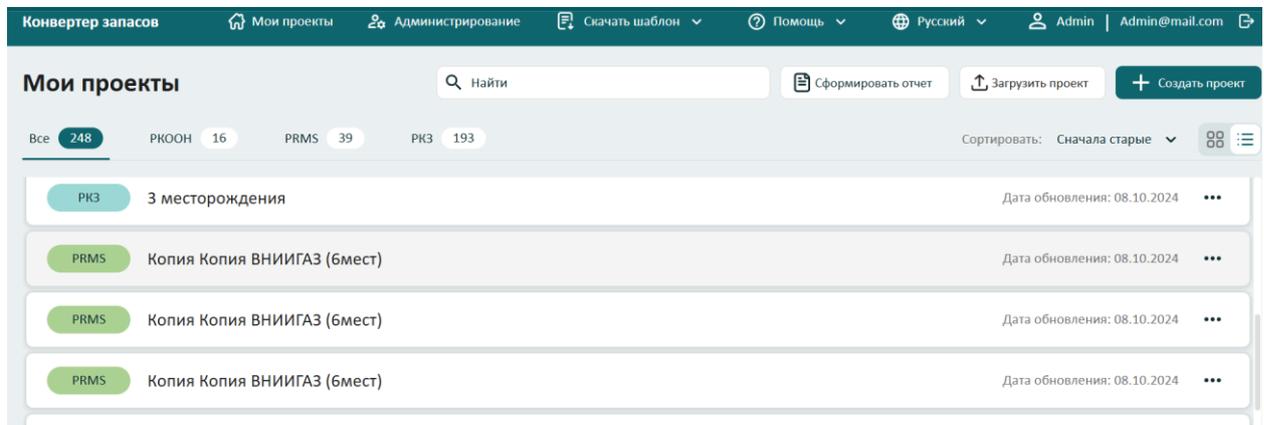


Рисунок 22. Главная страница

Если пользователь выполнил попытку авторизации под недействующей учетной записью, Система выдаст предупреждение (Рисунок 23).

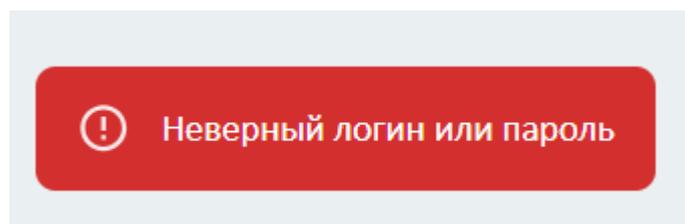


Рисунок 23. Попытка входа под недействующей учетной записью

5. МЕРОПРИЯТИЯ ПО ОБЕСПЕЧЕНИЮ РЕЗЕРВНОГО КОПИРОВАНИЯ И ВОССТАНОВЛЕНИЯ СИСТЕМЫ

5.1. Резервное копирование Системы

Для резервного копирования сервера приложений предусмотрено сохранение Docker-образа в файл с помощью команды «docker save». Файл может быть размещён в любом доступном месте сервера приложений.

Резервное копирование БД осуществляется с помощью стандартных средств создания бэкапа PostgreeSQL. Перед обновлением производится внеплановое резервное копирование.

5.2. Восстановление Системы из резервных копий

Восстановление БД на сервере СУБД осуществляется стандартными средствами СУБД PostgreeSQL. Необходимо запустить утилиту «pg_restore» в соответствии с официальной документацией (<https://postgrespro.ru/docs/postgresql>).

5.3. Тестирование Системы после восстановления

Тестирование Системы после восстановления производится системным администратором.

После восстановления Системы выполняются следующие шаги:

- проверка доступности интерфейса Системы;
- проверка доступности основных объектов Системы в интерфейсе Системы;
- проверка доступности отчетов Системы;
- проверка наличия данных и визуальный контроль наличия данных, представленных в отчетах Системы;
- проверка быстродействия Системы;
- проверка стабильности работы Системы;
- выполняется контроль целостности компонентов Системы;
- выполняется проверка корректности восстановления ролей и полномочий в соответствии с матрицами доступа.

5.4. Настройка архивации журналов событий Системы

Процедура сжатия и хранения сжатых архивов журналов событий Системы:

1. Журналы событий Docker могут быть выгружены в несжатый файл архива в любой доступный каталог ОС с помощью команды «docker logs».
2. Рекомендуется ежедневно запускать сценарий выгрузки логов и сжатия файлов архивов журналов (*.zip) с последующим переносом сжатых файлов в долгосрочное хранилище.

3. Журналы событий СУБД при достижении ими предельного размера (2 Гб) выгружаются в несжатый файл архива в любой доступный каталог сервера приложений.
4. Рекомендуется ежедневно запускать сценарий выгрузки журналов событий и сжатия файлов архивов журналов (*.zip) с последующим переносом сжатых файлов в долговременное хранилище.

Процедура резервного копирования настраивается администраторами при участии профильных специалистов отдела системного администрирования, отвечающих за настройку ОС Linux.

6. ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОВЕДЕНИЯ ИСПЫТАНИЙ

Минимальные требования к программному и аппаратному обеспечению пользователя для проведения тестирования представлены в таблице А1. Технические требования к оснащению серверов представлены в таблице А2.

Таблица А1. Минимальные требования к аппаратному обеспечению

№ п/п	Параметр	Минимальные требования	Рекомендуемая разработчиком
1.	Количество ядер процессоров (CPU)	16	2
2.	Тактовая частота	2,0 ГГц	2,8+ ГГц
3.	Оперативная память (RAM)	32 Гб	4 Гб
4.	Дисковая подсистема (HDD)	600 Гб	250 Гб
5.	Сетевое оборудование	Адаптер Ethernet 10 Мбит	Адаптер Ethernet 100 Мбит
6.	Операционная система (OS)	Linux (kernel > 3.2)	Linux (kernel > 3.2)
7.	Браузер Internet	Chrome (version > 110)	Chrome (version > 110)

Таблица А2. Технические требования к оснащению серверов

Сервер	Аппаратные характеристики		ПО сервера
	Параметр	Необходимое значение	
Сервер приложений / веб-сервер	Число ядер	8 и более	Linux (ядро версии 3.2 и выше), 64 бита, с поддержкой виртуализации Docker (версии 20 и выше)
	Тактовая частота	2+ ГГц	
	Оперативная память (RAM)	16Гб и более	
	Дисковая подсистема	100Гб	
	Сетевое оборудование	Адаптер Ethernet 1Гб	
Сервер БД	Число ядер	8 и более	Linux (ядро версии 3.2 и выше), 64 бита PostgreSQL (версия 16 или выше)
	Тактовая частота	2+ ГГц	
	Оперативная память (RAM)	16Гб и более	
	Дисковая подсистема, БД (важен быстрый доступ)	500Гб	
	Сетевое оборудование	Адаптер Ethernet 1Гб	

Сервер	Аппаратные характеристики		ПО сервера
	Параметр	Необходимое значение	
Сервер БД	Тактовая частота	2+ ГГц	Linux (ядро версии 3.2 и выше), 64 бита PostgreSQL (версия 16 или выше)
	Оперативная память (RAM)	8Гб	
	Дисковая подсистема, БД (важен быстрый доступ)	20 терабайт и более	
	Сетевое оборудование	Адаптер Ethernet 1Гб	