



ООО «Наука»
194223, г. Санкт-Петербург,
ул. Курчатова, д. 6, корп. 4,
пом. 2Н лит. А

Тел: +7(812)346-61-49
Факс: +7(812)346-61-45
office@ntik.ru
www.ntik.ru

ПО «Система моделирования технологических процессов NAUKA.Proxima ver. 0.1»

Руководство администратора

ред. 1.0

2024 г.

Оглавление

Сокращения и упрощения	3
1 Общие сведения.....	4
2 Описание сервера	5
2.1 Операционная система:	5
2.2 Системные компоненты сервера	5
3 Архитектура ПО	6
3.1 Системная архитектура продукта	6
3.2 Сторонние компоненты.....	6
3.3 Собственные компоненты	7
4 Порядок первичной установки системы	8
4.1 Предварительные требования	8
4.2 Установка ПО.....	8
5 Расположение файлов ПО на сервере.....	11
6 Настройка параметров, передаваемых в backend.....	12
7. Управление приложением.....	13

Сокращения и упрощения

В руководстве используются следующие сокращения:

- ПК – персональный компьютер;
- ПО – программное обеспечение;
- СУБД – система управления базами данных;
- СМТП – система моделирования технологических процессов;
- ЛКМ — левая кнопка мыши.

В руководстве используются следующие упрощения в терминологии:

- щелчок — нажатие и немедленное отпускание кнопки мыши;
- щёлкнуть — произвести щелчок;
- нажать кнопку – щёлкнуть ЛКМ по кнопке;
- система - СМТП.

1 Общие сведения

СМТП «NAUKA.Proxima» предназначена для моделирования технологических процессов на производствах нефтеподготовки, нефтепереработки, нефтехимии, химии и других видах производств.

Система настраивается и разворачивается пользователем самостоятельно, после разворачивания заказчику передаются логин и пароль

При работе пользователя с системой предполагается наличие начальных знаний и навыков работы с персональным компьютером.

Руководство пользователя составлено в объеме, достаточном для обучения и последующей работы в Системе.

Система устанавливается двумя подготовленными разработчиком пакетами только на операционную систему Alt-linux.

2 Описание сервера

2.1 Операционная система:

- ALT Server 10.XX

2.2 Системные компоненты сервера

Java-17

- OpenJDK 17.0.10 2024-01-16
- OpenJDK Runtime Environment Temurin-17.0.10+7 (build 17.0.10+7)
- OpenJDK 64-Bit Server VM Temurin-17.0.10+7 (build 17.0.10+7, mixed mode, sharing)

Postgresql

- **Версия СУБД:** psql (PostgreSQL) 14

Nginx

- Ver. 1.16.1

3 Архитектура ПО

3.1 Системная архитектура продукта

Системная архитектура продукта представлена на **Ошибка! Источник ссылки не найден..**

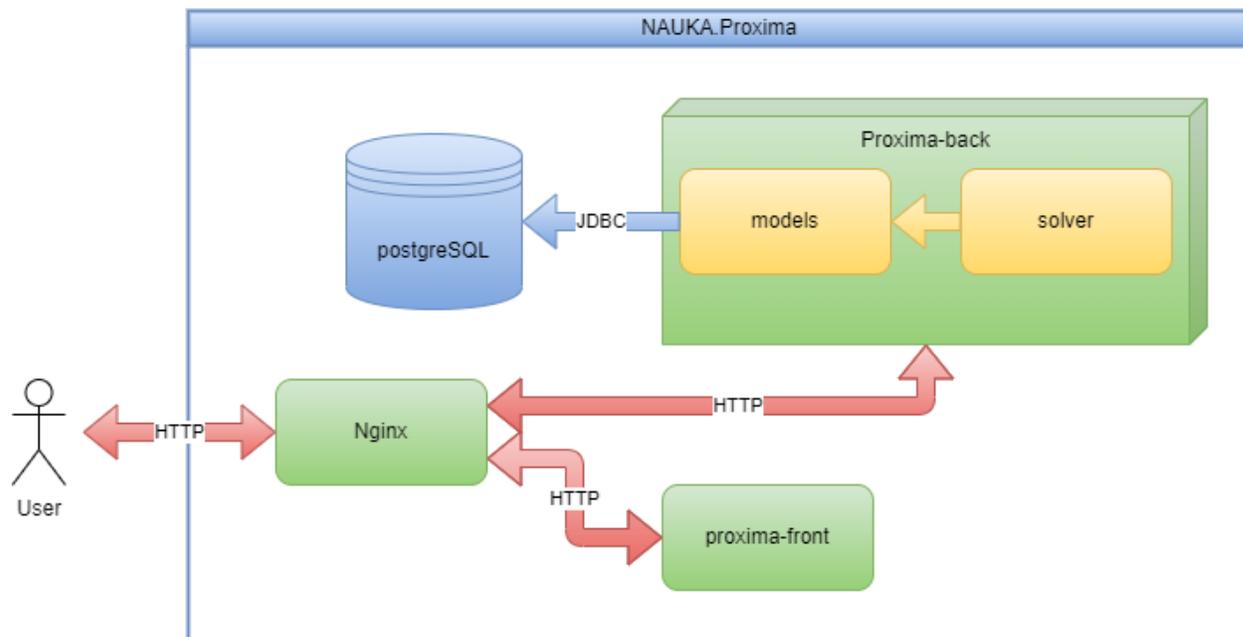


Рисунок 0.1 - Системная архитектура продукта

3.2 Сторонние компоненты

- **Alt-linux** - семейство дистрибутивов Linux, являющихся отдельной ветвью развития русскоязычного Linux.
- **PostgreSQL** - свободная объектно-реляционная система управления базами данных. Распространяется согласно разрешительной лицензии на свободное ПО. Используется для управления\хранения\обмена данными с ПО. В данном решении используется одна инсталляция PostgreSQL, в рамках которой используются 2 отдельные БД.
- **NGINX** - Веб-сервер с открытым исходным кодом, разработанный на языке программирования C, которое позволяет создавать и управлять HTTP-сервером и прокси-сервером. Используется для проксирования запросов от

клиента-веб-браузера к компонентам ПО, а также раздачи статических файлов.

- **Apache Tomcat** - комплект серверных программ от Apache Software Foundation, предназначенный для тестирования, отладки и исполнения веб-приложений на основе Java.
- **Java** - строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle).
- **Java Development Kit (сокращенно JDK)** — OpenJDK, бесплатно распространяемый компанией Oracle Corporation (ранее Sun Microsystems) комплект разработчика приложений на языке Java, включающий в себя компилятор Java, стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE). Используется для исполнения компонентов ПО в виде .jar файлов.

3.3 Собственные компоненты

- **Proxima-front** - главный веб-интерфейс ПО, в котором происходит работа пользователя. Предоставляет пользователю доступ ко всем компонентам ПО.
- **Models** – Пакет математических моделей технологических объектов. Содержит логику работы, а также математическую интерпретацию каждой модели.
- **Solver** - основной расчетный блок моделей. Содержит логику управления расчетами.

4 Порядок первичной установки системы

Система устанавливается на операционную систему Alt-linux двумя подготовленными разработчиком пакетами:

- наука-proxima-X.X.X.x86_64.rpm, где XXX – номер текущей версии ПО;
- наука-java17-X.X.X.x86_64.rpm, где XXX – номер текущей версии ПО.

4.1 Предварительные требования

На одном из серверов организации-пользователя СМТП должна быть установлена СУБД Postgre SQL 14.

4.2 Установка ПО

Перед установкой пакетов требуется их разместить в какой-либо папке на сервере и перейти в эту папку. Далее следует:

- набрать в командной строке «`sudo apt-get install наука-proxima - 9.XXXx86_64.rpm наука-java 17-17.XXXx86_64.rpm`» и нажать <Enter>;
- после этого на экран выводится сообщение об устанавливаемых пакетах, потребности количества дискового пространства в МВ и вопрос: Продолжить? [Y/n] (см. Рисунок 0.1);
- для продолжения установки нажать «Y» (см. Рисунок 0.1);
- после этого на экран выводится информация о процессе установки ПО (см. Рисунок 0.2)
- в окне «Enter tcp port for proxima (default http)» по умолчанию отображается порт 80 (см. Рисунок 0.3), наиболее часто используемый для web-приложений; если предполагается использование этого порта, то нажать кнопку <Ok>, если предполагается использование другого порта, то ввести его номер и нажать кнопку <Ok>;
- в окне «Enter postgresql database hostname or IP address:» ввести IP-адрес сервера, на котором установлена СУБД PostgreSQL (см. Рисунок 0.4);
- в окне «Enter PostgreSQL user» ввести имя пользователя СУБД (см. Рисунок 0.5);

- в окне «Enter postgresql password for ...:» ввести пароль для входа пользователя в СУБД (см. Рисунок 0.6);
- в окне «Enter jwt token secret» (см. Рисунок 0.7) ничего не изменяя нажать кнопку <Ok>; в дальнейшем данные в поле этого окна могут быть изменены разработчиком, поэтому оставлена возможность их изменения;
- в окне «Create iptables rule?» подтвердить нажатием кнопки «Да» разрешение использования tcp-порта web-приложением (см. Рисунок 0.8).

После этого на экран выводится сообщение о завершении установки пакетов ПО (см. Рисунок 0.9).

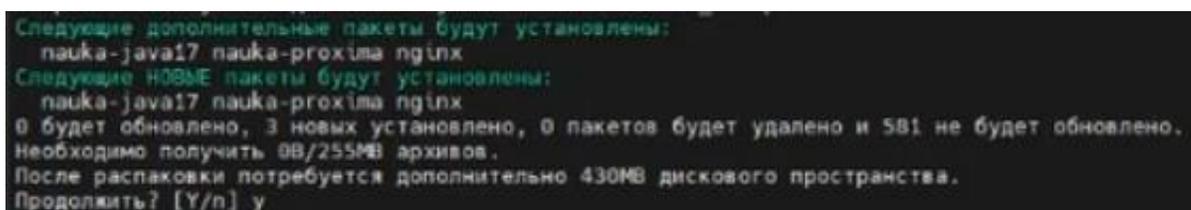


Рисунок 0.1 – Сообщения, выводимые на экран перед установкой пакетов



Рисунок 0.2 – Сообщения о процессе установки ПО

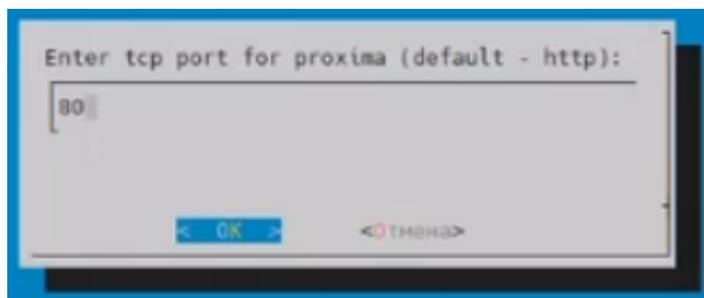


Рисунок 0.3 – Выбор tcp-порта для протокола http

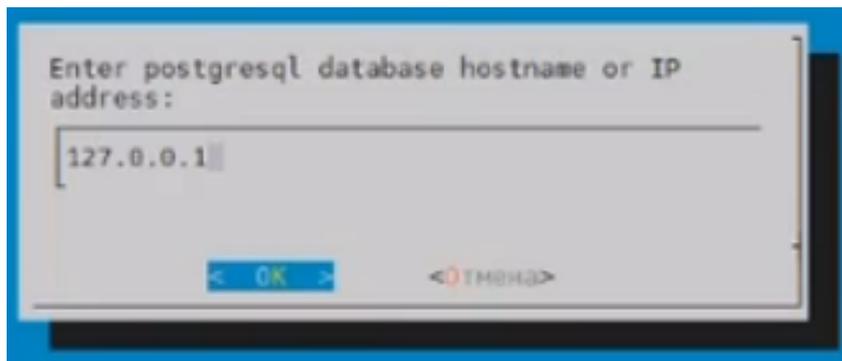


Рисунок 0.4 – Ввод IP-адреса сервера, на котором размещена СУБД PostgreSQL

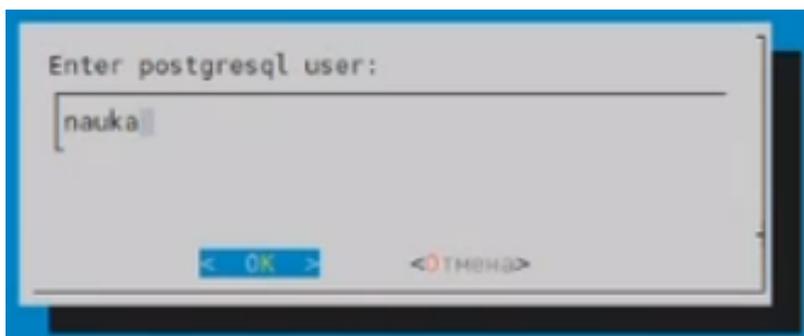


Рисунок 0.5 – Ввод имени пользователя СУБД

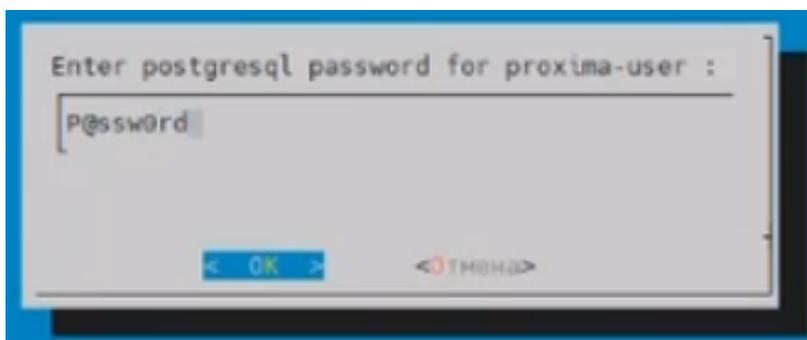


Рисунок 0.6 – Ввод пароля для входа в СУБД

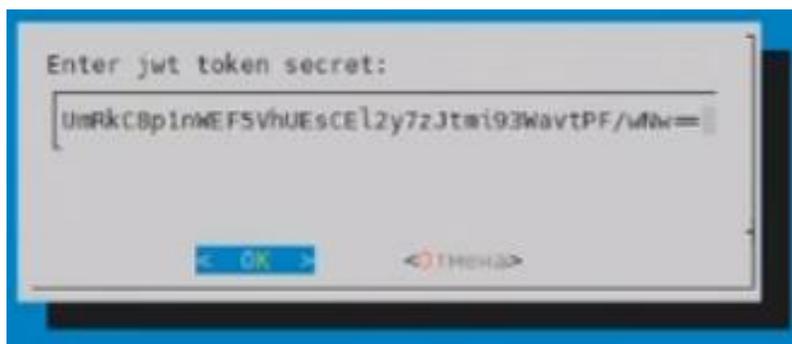


Рисунок 0.7 – Ввод данных в поле «Enter jwt token secret:»

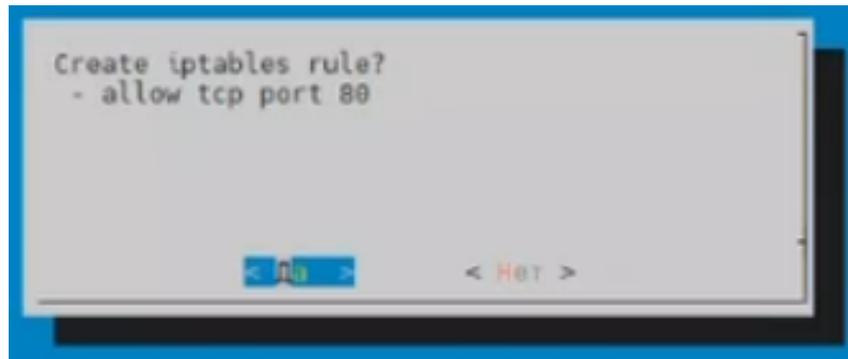


Рисунок 0.8 – Подтверждение разрешения использования заданного tcp-порта во внутреннем firewall'e

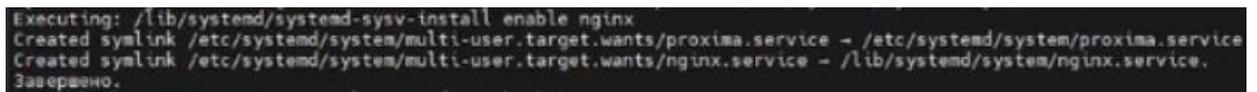


Рисунок 0.9 – Сообщение о завершении установки пакетов ПО

5 Расположение файлов ПО на сервере

После установки ПО основные его файлы находятся в директориях: /opt/nauka (см. Таблица 1).

Таблица 1

КомпONENTЫ	Расположение файла	Связанное ПО / Файлы
Proxima (backend)	/opt/nauka/proxima/version/proxima-X.X.X.jar	/opt/nauka/java/jdk-17/ /etc/systemd/system/proxima.service
Proxima (front)	/opt/nauka/proxima/version/front-X.X.X	/etc/nginx/sites-available.d/proxima.conf

Лог установки ПО можно посмотреть в файле opt/nauka/logs/proxima.log.

6 Настройка параметров, передаваемых в backend

В процессе установки ПО создаётся и регистрируется файл `proxima.service` (см. связанное ПО/файлы, Таблица 1), в котором содержатся параметры, передаваемые в backend (см. Рисунок 0.1).

```

user@alt-linux-10 proxima $ cat /etc/systemd/system/proxima.service
[Unit]
Description=nauka -- proxima-prod

[Service]
SuccessExitStatus=143
User=nauka
Group=nauka
Type=simple

Environment="SPRING_PROFILES_ACTIVE=manual"
Environment="SPRING_DATASOURCE_URL=jdbc:postgresql://127.0.0.1:5432/proxima?stringtype=unspecified"
Environment="SPRING_DATASOURCE_USERNAME=proxima-user"
Environment="SPRING_DATASOURCE_PASSWORD=password"
  
```

Рисунок 0.1 – Параметры файла `proxima.service`, передаваемые в backend

Этими параметрами являются:

- IP-адрес сервера, на котором установлена СУБД (см. 1, Рисунок 0.1);
- имя пользователя СУБД (см. 2, Рисунок 0.1);
- пароль для входа в СУБД (см. 3, Рисунок 0.1).

Все три параметра можно при необходимости здесь же изменить.

При запуске приложения в `opt/nauka/java/jdk-17/bin/java -jar` – `jar` передаются ещё несколько параметров, в том числе `jwt.token.secret`, который при необходимости можно здесь изменить (см. Рисунок 0.2).

```

ExecStart=/opt/nauka/java/jdk-17/bin/java -jar \
-Dserver.port=20992 \
-Dspring.datasource.driver-class-name=org.postgresql.Driver \
-Dspring.jpa.hibernate.ddl-auto=none \
-Djwt.token.secret=03X/h2xwfh4xMuZ01DjXZY+wJ58mx52M+kNFmcBsLs35UmRkCBpInwEFSVhUESCEl2y7zJtmi93WavtPF/wNw== \
-Djwt.token.expiration=864000 \
-Dupload.path=/opt/nauka/proxima/uploads \
/opt/nauka/proxima/proxima.jar > /opt/nauka/logs/proxima.log 2>&1
  
```

Рисунок 0.2 – `jwt.token.secret`

7. Управление приложением

Управление установленным приложением производится с помощью утилиты «systemctl». Основными командами данной утилиты являются:

- «status» - просмотр статуса приложения;
- «stop» - остановка приложения;
- «start» - запуск приложения;
- «restart» перезапуск приложения.

После команды указывается название приложения, с которым должна работать команда. Пример просмотра статуса приложения с помощью данной утилиты представлен на Рисунок 0.1.

```

user@alt-linux-10 proxima:~$ systemctl status proxima
* proxima.service - nauka - proxima-prod
   Loaded: loaded (/etc/systemd/system/proxima.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Wed 2024-04-17 14:17:51 MSK; 9min ago
   Process: 5066 ExecStart=/opt/nauka/java/jdk-17/bin/java -jar -Dserver.port=20992 -Dspring.datasource.driver-class-name=org.postgresql.Driver -Dspring.jpa.hibernate.ddl-auto=none
   Main PID: 5066 (code=exited, status=1/FAILURE)
   CPU: 18.069s

anp 17 14:17:51 alt-linux-10 java:5066 : at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:327) ~[na:na]
anp 17 14:17:51 alt-linux-10 java:5066 : at java.base/java.net.Socket.connect(Socket.java:633) ~[na:na]
anp 17 14:17:51 alt-linux-10 java:5066 : at org.postgresql.core.PGStream.createSocket(PGStream.java:243) ~[postgresql-42.6.0.jar!/:42.6.0]
anp 17 14:17:51 alt-linux-10 java:5066 : at org.postgresql.core.PGStream.<init>(PGStream.java:98) ~[postgresql-42.6.0.jar!/:42.6.0]
anp 17 14:17:51 alt-linux-10 java:5066 : at org.postgresql.core.v3.ConnectionFactoryImpl.tryConnect(ConnectionFactoryImpl.java:132) ~[postgresql-42.6.0.jar!/:42.6.0]
anp 17 14:17:51 alt-linux-10 java:5066 : at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:258) ~[postgresql-42.6.0.jar!/:42.6.0]
anp 17 14:17:51 alt-linux-10 java:5066 : -- 129 common frames omitted
anp 17 14:17:51 alt-linux-10 systemd: proxima.service: Main process exited, code=exited, status=1/FAILURE
anp 17 14:17:51 alt-linux-10 systemd: proxima.service: Failed with result 'exit-code'.
anp 17 14:17:51 alt-linux-10 systemd: proxima.service: Consumed 18.069s CPU time.
user@alt-linux-10 proxima:~$ systemctl stop proxima
  
```

Рисунок 0.1 – Просмотр статуса приложения Proxima