

# Инструкция по установке программного обеспечения

**Наименование ПО:** NafanyaClassifierAI — Классификатор строительных ресурсов

**Версия:** 1.0

**Дата документа:** 02.07.2026

---

## 1. Общие сведения

Настоящая инструкция описывает порядок установки, настройки и запуска программного комплекса `res_cl`, состоящего из трёх сервисов и СУБД PostgreSQL.

Экземпляр программного обеспечения для проведения экспертной проверки также доступен в режиме SaaS без установки: <https://ainafanyacl.ru>

Демонстрационный дистрибутив для локальной установки: <https://disk.yandex.ru/d/dWMN0P91wXFgvQ>

## 2. Состав дистрибутива

Содержимое архива `res_cl-demo-1.0.tar.gz` после распаковки:

```
INSTALL_DEMO.md
res_cl_api/
embed_res_nom_api/
embed_res_group_api/
```

### В архив не включены:

- виртуальное окружение Python (`ksr_env/`) — создаётся при установке (раздел 4.3);
- обученные веса нейросетевых моделей (`model.safetensors`);
- документация — публикуется отдельно (вне архива дистрибутива).

После установки на сервере типовая структура каталога:

```
/opt/res_cl/
├─ INSTALL_DEMO.md
├─ res_cl_api/
├─ embed_res_nom_api/
├─ embed_res_group_api/
└─ ksr_env/                # создаётся при установке, в архив не входит
```

## 3. Предварительные требования

### 3.1. Программное обеспечение

| Компонент           | Версия                           |
|---------------------|----------------------------------|
| ОС                  | Linux (64-bit)                   |
| Python              | 3.12                             |
| PostgreSQL          | 14 или выше                      |
| Расширение pgvector | Совместимое с версией PostgreSQL |
| pip                 | Актуальная версия                |

Для GPU-инференса (рекомендуется):

- NVIDIA Driver + CUDA (совместимые с PyTorch);
- видеокарта с объёмом VRAM не менее 4 ГБ.

### 3.2. Сетевые порты

| Порт | Сервис              |
|------|---------------------|
| 4300 | res_cl_api          |
| 3010 | embed_res_nom_api   |
| 3011 | embed_res_group_api |
| 5432 | PostgreSQL          |

Убедитесь, что порты свободны или измените их в конфигурации.

## 4. Установка

### 4.1. Установка PostgreSQL и pgvector

```
# Пример для Debian/Ubuntu
sudo apt update
sudo apt install -y postgresql postgresql-contrib

# Установка pgvector (способ зависит от дистрибутива)
# См. https://github.com/pgvector/pgvector
```

Создайте пользователя и базу данных (укажите собственные имя пользователя, пароль и имя базы):

```
CREATE USER <имя_пользователя> WITH PASSWORD '<пароль>';
CREATE DATABASE <имя_базы> OWNER <имя_пользователя>;
```

```
\с <имя_базы>  
CREATE EXTENSION IF NOT EXISTS vector;
```

## 4.2. Распаковка дистрибутива

```
# Распакуйте архив в целевую директорию, например:  
sudo mkdir -p /opt/res_cl  
sudo tar -xzf res_cl-demo-1.0.tar.gz -C /opt/res_cl  
cd /opt/res_cl
```

## 4.3. Создание виртуального окружения Python

```
python3.12 -m venv ksr_env  
source ksr_env/bin/activate
```

## 4.4. Установка зависимостей

```
# Основной сервис  
pip install -r res_cl_api/requirements.txt  
  
# Embed-сервисы (включая PyTorch; при необходимости установите torch с CUDA  
отдельно)  
pip install -r embed_res_nom_api/requirements.txt  
pip install -r embed_res_group_api/requirements.txt
```

## 4.5. Проверка наличия моделей BERT

Убедитесь, что в дистрибутиве присутствуют каталоги моделей:

- `embed_res_nom_api/model/m33_nom_berta_best/epoch_19_20260403_203444/`
- `embed_res_group_api/model/m34_group_berta_best/epoch_18_20260403_203620/`

При необходимости скорректируйте пути в файлах `embed_res_nom_api/config.py` и `embed_res_group_api/config.py`.

## 4.6. Настройка конфигурации

### res\_cl\_api

```
cd res_cl_api  
cp .env.example .env
```

Отредактируйте `.env` — укажите параметры подключения к PostgreSQL, созданные на шаге 4.1:

```
APP_PORT=4300
```

```
DB_USER=<имя_пользователя>
```

```
DB_PASSWORD=<пароль>
```

```
DB_HOST=localhost
```

```
DB_PORT=5432
```

```
DB_NAME=<имя_базы>
```

```
DB_SCHEMA=<имя_схемы>
```

```
EMBED_GROUP_API_URL=http://127.0.0.1:3011/embed
```

```
EMBED_NOM_API_URL=http://127.0.0.1:3010/embed
```

```
DEFAULT_BASE_FILE=/opt/res_cl/res_cl_api/data/base.xlsx
```

```
CHANGES_GROUP_PATH=/opt/res_cl/res_cl_api/data/changes_group.xlsx
```

```
CHANGES_NOM_PATH=/opt/res_cl/res_cl_api/data/changes_nom.xlsx
```

## **embed\_res\_nom\_api и embed\_res\_group\_api**

В файлах `config.py` каждого embed-сервиса при необходимости укажите:

- `model_path` / `tokenizer_path` — пути к моделям;
- `visible_devices` — номер GPU ("0", "1" и т.д.) или оставьте для CPU;
- `device` — `auto`, `cpu` или `cuda`.

## **5. Запуск**

### **5.1. Ручной запуск (для проверки)**

Запустите в трёх отдельных терминалах:

```
# Терминал 1 – embed номенклатур
cd /opt/res_cl
source ksr_env/bin/activate
cd embed_res_nom_api && python main.py

# Терминал 2 – embed групп
cd /opt/res_cl
source ksr_env/bin/activate
cd embed_res_group_api && python main.py

# Терминал 3 – основной сервис
cd /opt/res_cl
source ksr_env/bin/activate
cd res_cl_api && python run.py
```

## 5.2. Проверка работоспособности

```
curl http://127.0.0.1:3010/health
curl http://127.0.0.1:3011/health
curl http://127.0.0.1:4300/health
```

Ожидаемый ответ: HTTP 200.

## 5.3. Установка как systemd-сервисов

Для production-эксплуатации:

```
cd /opt/res_cl/res_cl_api
# Отредактируйте пути в res_cl_api.service при необходимости
sudo cp res_cl_api.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl enable res_cl_api.service
sudo systemctl start res_cl_api.service
```

Аналогично для embed-сервисов (файлы `embed_res_nom_api.service`, `embed_res_group_api.service`).

Проверка статуса:

```
sudo systemctl status res_cl_api.service
```

## 6. Первоначальная настройка и загрузка базы

После установки и запуска сервисов выполните первоначальную загрузку базы классификатора.

### 6.1. Через веб-интерфейс

1. Откройте в браузере: `http://<адрес_сервера>:4300/base`
2. Нажмите «Загрузить базу по умолчанию» или загрузите свой файл (xlsx/csv/json).
3. Дождитесь завершения обработки (статус `ready`).

### 6.2. Через REST API

```
# Загрузка базы по умолчанию
curl -X POST http://127.0.0.1:4300/api/base/load-default

# Проверка статуса
curl http://127.0.0.1:4300/api/base/status
```

### 6.3. Загрузка пользовательского файла

```
curl -X POST http://127.0.0.1:4300/api/base/upload \  
-F "file=@/path/to/your_base.xlsx"
```

После успешной загрузки базы (статус `ready`) программное обеспечение готово к эксплуатации. Дальнейшие действия — см. документ «Руководство по эксплуатации».