

Руководство по установке и эксплуатации «Руконтекст 2.0»

1. Требования к серверу

1.1. Аппаратные требования

Параметр	Минимальные	Рекомендуемые
Процессор	x86-64, 16 ядер	x86-64, 32+ ядер
Оперативная память	64 ГБ	256+ ГБ
Дисковое пространство	512 ГБ (SSD)	1+ ТБ (SSD/NVMe)
GPU (опционально)	—	NVIDIA с CUDA 11.7+ (детектор AI-текста) или CUDA 12.2+ (переводчик, выравнивание параграфов)

1.2. Программные требования

Компонент	Версия
Операционная система	Linux x86-64 (Astra Linux, ALT Linux, Ubuntu 22.04+)
Docker Engine	24.0 и выше
Docker Compose	v2 (входит в Docker Engine)
Драйвер NVIDIA (при использовании GPU)	525.0+ (детектор AI-текста) или 535.0+ (переводчик, выравнивание параграфов)
NVIDIA Container Toolkit (при использовании GPU)	1.14+

2. Подготовка к установке

2.1. Установка Docker

Для Astra Linux / Debian-based систем:

```
# Установка Docker Engine
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin

# Добавление текущего пользователя в группу docker
sudo usermod -aG docker $USER

# Проверка
docker --version
docker compose version
```

2.2. Подготовка каталогов

```
# Создание каталога для данных
sudo mkdir -p /incx-dir
sudo chown $USER:$USER /incx-dir

# Распаковка дистрибутива
tar -xzf icontext-standalone-<VERSION>.tar.gz -C /opt/icontext
cd /opt/icontext
```

2.3. Загрузка Docker-образов

Дистрибутив поставляется с предсобранными Docker-образами в формате tar-архивов:

```
# Загрузка всех образов
for img in images/*.tar; do
    docker load -i "$img"
done

# Проверка загруженных образов
docker images | grep incx
```

3. Конфигурация

3.1. Файл окружения (.env)

Отредактируйте файл `.env` в корневом каталоге установки:

```
# Каталог для хранения данных (тома Docker)
DATA_DIR=/incx-dir

# Каталог для импорта индексных коллекций с центрального сервера (опционально)
CENTRAL_BACKUP_DIR=/incx-dir/search-backup/central

# GPU-конфигурация (опционально, раскомментировать при наличии GPU)
# COMPOSE_PROFILES=gpu
# TRANSLATOR_GPU=0
# PARALIGN_GPU=0
# FAKEDETECTOR_DEVICE=cuda:0
```

3.2. Параметры организации

В файле `docker-compose.yml` отредактируйте параметры организации в сервисе `database-migrator`:

```
environment:
  ORG_TIN: '1234567890'           # ИНН организации
  ORG_SHORT_NAME: 'Название организации' # Краткое наименование
  ORG_FULL_NAME: 'Полное наименование'  # Полное наименование
  ENABLE_CROSSLANG_SEARCH: 'false'       # Кросс-языковой поиск (true/false)
```

3.3. Настройка email-уведомлений (опционально)

В файле конфигурации `configs/configuration.json` настройте SMTP-сервер:

```
{
  "EmailSenderSettings": {
    "SmtpHost": "smtp.example.com",
    "SmtpPort": 587,
    "SmtpUser": "noreply@example.com",
    "SmtpPassword": "password",
    "FromEmail": "noreply@example.com",
    "FromName": "InContext"
  }
}
```

3.4. Настройка HTTPS (рекомендуется)

Для промышленной эксплуатации рекомендуется настроить reverse proxy (nginx) с TLS-терминацией:

```

server {
    listen 443 ssl;
    server_name icontext.example.com;

    ssl_certificate      /etc/ssl/certs/icontext.crt;
    ssl_certificate_key  /etc/ssl/private/icontext.key;

    location / {
        proxy_pass http://localhost:5001;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # WebSocket для SignalR
    location /hubs/ {
        proxy_pass http://localhost:5001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

```

4. Установка и запуск

4.1. Первый запуск

```

cd /opt/incontext

# Запуск всех сервисов
docker compose up -d

# Наблюдение за процессом запуска
docker compose logs -f

```

При первом запуске система автоматически:

1. Создаст базу данных PostgreSQL и применит схему
2. Наполнит БД начальными данными (справочники, локализация, настройки)
3. Создаст поисковые индексы Elasticsearch с настроенными анализаторами
4. Настроит алиасы для индексов

Время первого запуска: 2–5 минут в зависимости от производительности сервера.

4.2. Проверка состояния

```
# Статус всех контейнеров
docker compose ps
```

```
# Ожидаемый результат: все контейнеры в статусе "Up" (healthy)
```

Контейнер	Ожидаемый статус
incx-standalone	Up (healthy)
incx-database	Up (healthy)
incx-search	Up (healthy)
incx-plagiarism	Up (healthy)
incx-dprocessor	Up (healthy)
incx-fakedetector	Up (healthy)
incx-checker	Up (healthy)
incx-reports	Up (healthy)
Контейнер	Ожидаемый статус
incx-sender	Up (healthy)
incx-rabbitmq	Up (healthy)
incx-redis	Up (healthy)
incx-etcd	Up (healthy)
incx-database-scheduler	Up (healthy)
incx-search-init	Exited (0)
incx-database-migrator	Exited (0)

4.3. Доступ к веб-интерфейсу

После успешного запуска откройте в веб-браузере:

```
http://<IP-адрес-сервера>:5001
```

5. Управление системой

5.1. Остановка

```
docker compose down
```

5.2. Перезапуск

```
docker compose restart
```

5.3. Обновление

```
# Остановка текущей версии
docker compose down

# Загрузка новых образов
for img in images/*.tar; do
    docker load -i "$img"
done

# Запуск обновлённой версии
docker compose up -d
```

Данные (БД, индексы, документы) сохраняются в томах Docker и не теряются при обновлении.

5.4. Просмотр журналов

```
# Все сервисы
docker compose logs -f

# Конкретный сервис
docker compose logs -f incx-standalone
docker compose logs -f incx-plagiarism

# Последние 100 строк
docker compose logs --tail=100 incx-standalone
```

6. Резервное копирование

6.1. База данных

```
# Создание резервной копии
docker exec incx-database pg_dump -U incx incx_database > backup_$(date
+%Y%m%d).sql

# Восстановление
docker exec -i incx-database psql -U incx incx_database < backup_20260223.sql
```

6.2. Поисковые индексы

```
# Создание снимка
./scripts/search-backup.sh

# Восстановление из снимка
./scripts/search-restore.sh <имя_снимка> <имя_индекса>
```

6.3. Загруженные документы

Каталог `/incx-dir` содержит все персистентные данные. Рекомендуется включить его в регулярное резервное копирование.

7. Устранение неполадок

Контейнер не запускается

```
# Проверка журналов
docker compose logs <имя-сервиса>
```

```
# Проверка ресурсов
docker stats --no-stream
```

Не хватает оперативной памяти

Elasticsearch требует значительного объёма памяти. При необходимости уменьшите heap size в файле `elasticsearch/jvm.options`:

```
-Xms16g
-Xmx16g
```

База данных не инициализируется

```
# Проверка журналов мигратора
docker compose logs incx-database-migrator

# Ручная переинициализация (удалит все данные!)
docker compose down
docker volume rm $(docker volume ls -q | grep database-data)
docker compose up -d
```

8. Сетевые порты

Порт	Протокол	Сервис	Доступ
5001	HTTP	Веб-приложение	Внешний (пользователи)

5433	TCP	PostgreSQL	Внутренний / для администрирования
9200	HTTP	Elasticsearch	Внутренний
15672	HTTP	RabbitMQ Management	Внутренний / для администрирования

Все остальные порты используются только внутри Docker-сети и не требуют открытия на межсетевом экране.