

02 —

Инфраструктура

Раздел базы знаний ITG

- [02 — Инфраструктура](#)
- [06 — ZimaOS и Docker](#)
- [11 — Реестр сервисов ITG](#)
- [16 — Docker: безопасное обновление](#)

02 — Инфраструктура

Инфраструктура ITG

Компоненты

- **ZimaOS:** домашний сервер для контейнерных сервисов.
- **Docker Compose:** управление сервисами.
- **Nginx Proxy Manager:** обратный прокси.
- **Cloudflare:** DNS и внешний доступ к доменам.
- **OpenWrt:** маршрутизация локальной сети.
- **BookStack:** <https://wiki.karpiev.ru> — документация.
- **Twenty CRM:** <https://crm.itgsystem.online>.

Стандарт документации сервиса

1. Назначение и адрес сервиса.
2. Путь к Compose-конфигурации.
3. Сети, тома, зависимости.
4. Процедура обновления и отката.
5. Резервное копирование и восстановление.
6. Мониторинг, логи, алерты.

Безопасность

Не публиковать ключи, токены, пароли, файлы `.env` и приватные SSH-ключи. Хранить секреты отдельно в менеджере секретов.

06 — ZimaOS и Docker

ZimaOS и Docker — ITG

Назначение

Домашняя инфраструктура для приложений ITG: CRM, документация, автоматизация и MCP-сервисы.

Известная конфигурация

- ZimaOS и Docker Compose.
- Nginx Proxy Manager для обратного прокси.
- Cloudflare для DNS и внешнего доступа.
- OpenWrt для локальной сети.
- Для Docker на ZimaOS ранее требовался `sudo`.
- Рабочий каталог Docker CLI: `/DATA/AppData/docker-cli-config`.

Команды обслуживания

```
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker ps
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose ps
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker logs --tail 50 ИМЯ_КОНТЕЙНЕРА
```

Выполнять `docker compose` из каталога конкретного сервиса.

Стандарт контейнерного проекта

- Один постоянный каталог проекта в `/DATA/AppData/`.
- Compose-файл и отдельный `.env` с ограниченными правами.
- Том для данных, который сохраняется при пересоздании контейнера.
- Healthcheck и политика перезапуска, где применимо.
- Резервная копия конфигурации и данных до изменений.

Диагностика HTTP

1. Проверить процесс/контейнер.
2. Проверить адрес, который слушает приложение внутри контейнера: `0.0.0.0` для Docker port mapping.
3. Проверить проброс портов и Docker-сети.
4. Проверить Nginx Proxy Manager.
5. Проверить DNS, TLS и правила Cloudflare.

Безопасность

Не публиковать административные порты, Docker API и внутренние сервисы без авторизации. Не размещать секреты в BookStack.

Требует актуализации: полный инвентарь контейнеров, томов, бэкапов и схема сетей.

11 — Реестр сервисов ITG

Реестр сервисов ITG

Подтверждённые адреса и назначения

Сервис	Адрес	Назначение
BookStack	https://wiki.karpiev.ru	База знаний
ITG BookStack MCP	https://mcp-wiki.karpiev.ru/mcp	Инструменты BookStack для ChatGPT
Twenty CRM	https://crm.itgsystem.online	Управление продажами
ITG сайт	https://itgsystem.ru	Публичный сайт студии

Проверка доступности

11.10.2026

- <https://wiki.karpiev.ru/books/itg-baza-znaniy>: HTTP 200.
- <https://mcp-wiki.karpiev.ru/mcp> без авторизации: HTTP 401; проверка OAuth через авторизованный клиент выполнена отдельными вызовами MCP.
- <https://itgsystem.ru>: HTTP 200.
- <https://crm.itgsystem.online>: HTTP 404 при GET и HEAD. Работоспособность CRM не подтверждена; проверить маршрут прокси и актуальный адрес. Это результат внешней проверки, внутреннее состояние контейнеров не проверялось.

Инфраструктура (по предыдущей конфигурации)

- ZimaOS: контейнерные приложения.
- Nginx Proxy Manager: обратный прокси.
- Cloudflare: DNS и проксирование.
- OpenWrt: домашняя сеть.

Шаблон инвентаризации

Поле	Значение
Имя сервиса	
Назначение	
URL	
Каталог Compose	
Docker-образы и версии	
Тома данных	
База данных	
Ответственный	
Резервная копия	
Последняя проверка	

Важно: адреса не являются подтверждением текущей доступности. Перед эксплуатационными действиями проверять реальное состояние.

16 — Docker: безопасное обновление

Docker — безопасное обновление сервиса ITG

Область применения

Контейнерные сервисы ZimaOS. Процедура не заменяет инструкции конкретного приложения.

Перед обновлением

1. Уточнить каталог Compose и текущую версию образа.
2. Проверить доступное место и статус томов.
3. Сохранить резервные копии конфигурации и данных.
4. Зафиксировать план отката и окно обслуживания.

Команды проверки

```
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose ps
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose config --quiet
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose images
```

Обновление

Выполнять из существующего каталога проекта, только после проверки резервных копий:

```
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose pull
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose up -d
```

Проверка

```
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose ps
sudo env DOCKER_CONFIG=/DATA/AppData/docker-cli-config docker compose logs --tail 100
```

Проверить реальную бизнес-операцию приложения.

Откат

Вернуть зафиксированную версию образа и совместимые данные из резервной копии. Не откатывать схему БД вслепую: миграции могут быть необратимыми.

Статус: инструкция подготовлена, на боевом сервисе не выполнялась.