Руководство администратора

Платформа «Панда» Правообладатель — ООО «НЕЙРОПРЕМ» · info@neuroprem.ru · +7 995 007-17-57 Редакция от 17.07.2026

---

1. Назначение документа

Документ описывает установку, настройку, эксплуатацию, резервное копирование и обновление платформы «Панда». Предназначен для системного администратора, обладающего навыками работы с ОС Linux, Docker и СУБД PostgreSQL.

Порядок развёртывания проверочного экземпляра для экспертизы приведён в отдельном документе `05-Инструкция-по-установке-экземпляра-для-экспертизы.md`.

---

2. Архитектура развёртывания

Платформа поставляется в виде набора контейнеров Docker, управляемых Docker Compose.

Прикладные сервисы (собираются в единый образ `panda-app:dev`):

Сервис	Внутренний порт	Назначение
`api-gateway`	8000	REST API, аутентификация, авторизация
`agent-engine`	8001	обработка диалогов, RAG, интеграции, фоновые задачи
`connector-service`	8002	веб-хуки каналов, отправка сообщений, веб-виджет
`billing-service`	8003	платежи (ЮKassa)
`notification-service`	8004	уведомления по электронной почте
`migrate`	—	разовая задача применения миграций БД

Frontend: `web` (образ `panda-web:dev`, nginx со статикой личного кабинета), `site` (образ `panda-site:dev`, Next.js — публичный сайт, порт 3000).

Инфраструктурные компоненты:

Компонент	Образ	Назначение
PostgreSQL + pgvector	`pgvector/pgvector:pg16`	основная БД, векторный поиск, состояния диалогов
Redis	`redis:7-alpine`	кеш, шина событий, счётчики ограничения частоты
MinIO	`minio/minio:latest`	объектное хранилище документов базы знаний
Keycloak	`quay.io/keycloak/keycloak:24.0`	OIDC-провайдер
Centrifugo	`centrifugo/centrifugo:v5`	WebSocket
ClickHouse	`clickhouse/clickhouse-server:24-alpine`	аналитика
Caddy	`caddy:2`	обратный прокси, TLS

Дополнительные профили: `--profile monitoring` — Prometheus, Grafana, Alertmanager, postgres-exporter, redis-exporter; `--profile tools` — pgAdmin, n8n, Portainer.

Схема сети. Единственная публичная точка входа — Caddy (порты 80/443). Порты всех остальных сервисов в промышленной конфигурации закрыты от хоста (`ports: !reset []` в `docker-compose.prod.yml`).

---

3. Требования к аппаратному и системному обеспечению

3.1. Программные требования

- ОС: Linux x86-64. Проверенная конфигурация — **Ubuntu 24.04 LTS**.
- **Docker Engine** и **Docker Compose v2**.
- Доступ в интернет для загрузки образов и выпуска TLS-сертификатов, а также для обращения к API поставщиков языковых моделей (Yandex Foundation Models, GigaChat) и платёжного сервиса ЮKassa.
- Для варианта on-premise без доступа в интернет требуется предварительная загрузка образов и доступность LLM-сервисов внутри периметра.

3.2. Аппаратные требования

Параметр	Проверенная конфигурация	Комментарий
CPU	4 vCPU	
ОЗУ	**6 ГБ + 4 ГБ swap**	Файл подкачки на 4 ГБ обязателен при 6 ГБ ОЗУ; при 6 ГБ в покое занято ≈2,2 ГБ
Диск	80 ГБ	Основной потребитель — ClickHouse
Сеть	публичный IPv4, открытые порты 80/443	Необходимы для выпуска сертификатов Let's Encrypt

Рекомендуемая конфигурация для промышленной эксплуатации: 4 vCPU / 8–16 ГБ ОЗУ / 100 ГБ SSD. Указанная конфигурация 4 vCPU / 6 ГБ — минимальная проверенная; при включении профиля `monitoring` потребление возрастает.

Количество контейнеров: 14 в базовой промышленной конфигурации (+5 с профилем `monitoring`, +3 с профилем `tools`).

Отдельно учитывается требование Keycloak — не менее 512 МБ ОЗУ (ADR-003).

3.3. Требования к DNS

Требуется четыре записи типа A на адрес сервера (плюс `www`):

Поддомен	Назначение	Пример
основной домен	публичный сайт	`panda-ai.ru`
`www`	редирект 301 на основной	`www.panda-ai.ru`
`app`	личный кабинет и REST API	`app.panda-ai.ru`
`auth`	сервер аутентификации Keycloak	`auth.panda-ai.ru`
`hooks`	приём веб-хуков мессенджеров и ЮKassa, веб-виджет	`hooks.panda-ai.ru`

---

4. Установка

4.1. Подготовка

```
# Файл подкачки (обязателен при 6 ГБ ОЗУ)
fallocate -l 4G /swapfile && chmod 600 /swapfile && mkswap /swapfile && sw
echo '/swapfile none swap sw 0 0' >> /etc/fstab

# Docker и Docker Compose v2
apt-get update
apt-get install -y ca-certificates curl zstd docker.io docker-compose-v2 d
```

Разместите дистрибутив платформы в рабочем каталоге, например `/data/projects/panda-dev`.

4.2. Настройка переменных окружения

Создайте файл `.env` в корне проекта на основе `.env.example`. **Полный перечень переменных — раздел 5.** Минимально необходимо задать секреты, значения для которых не имеют безопасных значений по умолчанию (см. 5.1).

Генерация секретов:

```
bash scripts/gen-secrets.sh
# либо по отдельности:
python3 -c "import base64,os; print(base64.urlsafe_b64encode(os.urandom(32
openssl rand -hex 32      # INTERNAL_API_TOKEN, CENTRIFUGO_TOKEN_SECRET
openssl rand -base64 24   # пароли БД и хранилищ
```

4.3. Запуск (промышленная конфигурация)

```
cd /data/projects/panda-dev

docker compose \
  -f docker-compose.yml \
  -f docker-compose.app.yml \
  -f docker-compose.prod.yml \
  -f docker-compose.proxy.yml \
  up -d
```

Назначение файлов:

Файл	Роль
`docker-compose.yml`	инфраструктура (БД, кеш, хранилище, Keycloak, Centrifugo, мониторинг)
`docker-compose.app.yml`	прикладные сервисы и задача миграций
`docker-compose.prod.yml`	**промышленный слой:** обязательность секретов, закрытие портов от хоста, `APP_ENV=production`, healthcheck'и, режим Keycloak `start --import-realm`
`docker-compose.proxy.yml`	Caddy — TLS и маршрутизация

 Слой `docker-compose.prod.yml` обязателен для промышленной эксплуатации. Без него не действуют проверки обязательности секретов, порты сервисов остаются открытыми на хосте, а `APP_ENV` остаётся `development`, что отключает защитные проверки конфигурации. Файл `docker-compose.dev-override.yml` предназначен **только для локальной разработки** (он включает `DEV_NO_AUTH=true` — полный обход проверки токенов доступа) и в промышленном контуре применяться не должен.

Миграции применяются автоматически: задача `migrate` выполняет `alembic upgrade head`, а все прикладные сервисы стартуют только после её успешного завершения (`depends_on: migrate: condition: service_completed_successfully`).

4.4. Проверка после установки

```
docker compose ps                       # все контейнеры в состоянии runni

# Проверка работоспособности сервисов (внутри сети или при открытых портах
for p in 8000 8001 8002 8003 8004; do
  curl -s -m5 localhost:$p/health -o /dev/null -w ":$p %{http_code}\n"
done
curl -s localhost:8000/ready           # готовность (включая доступность Б
curl -s localhost:8001/health          # ожидается {"graph_ready": true}
```

Проверка конфигурации обратного прокси:

```
docker run --rm -v "$PWD/infra/proxy/Caddyfile:/etc/caddy/Caddyfile:ro" \
  caddy:2 caddy validate --config /etc/caddy/Caddyfile
```

5. Переменные окружения

5.1. Обязательные в промышленном режиме

При `APP_ENV=production` сервис **не запустится** (`ValueError` на старте), если не заданы или заданы небезопасно следующие переменные:

Переменная	Назначение
`DATABASE_URL`	строка подключения к PostgreSQL (единственная переменная без значения по умолчанию — обязательна всегда)
`ENCRYPTION_KEY`	ключ Fernet для шифрования учётных данных каналов и CRM (152-ФЗ)
`ENCRYPTION_KEYS`	список ключей через запятую для ротации (MultiFernet); имеет приоритет над `ENCRYPTION_KEY`
`INTERNAL_API_TOKEN`	токен межсервисного взаимодействия
`TRUSTED_PROXY_HOSTS`	доверенные адреса прокси; значение `*` в промышленном режиме запрещено
`CENTRIFUGO_API_KEY`, `CENTRIFUGO_TOKEN_SECRET`	должны совпадать с настройками Centrifugo
`ALLOWED_ORIGINS`	разрешённые источники CORS; `*`, `localhost`, `127.0.0.1` и пустое значение запрещены
`REDIS_PASSWORD`	пароль Redis (обязателен даже в базовом файле — Redis стартует с `--requirepass`)
`POSTGRES_PASSWORD`, `MINIO_ROOT_PASSWORD`, `CLICKHOUSE_PASSWORD`, `KEYCLOAK_ADMIN_PASSWORD`, `GRAFANA_ADMIN_PASSWORD`	пароли компонентов
`KC_HOSTNAME`	публичный домен Keycloak
`ACME_EMAIL`, `PANDA_APP_DOMAIN`, `PANDA_HOOKS_DOMAIN`, `PANDA_AUTH_DOMAIN`, `PANDA_LANDING_DOMAIN`	параметры Caddy (обязательны)

Дополнительно действует **список запрещённых значений по умолчанию**: старт в промышленном режиме блокируется, если секрет совпадает с известным демонстрационным значением (`dev-internal-token`, `minioadmin`, `minioadmin123`, `dev_token_secret_change_in_prod`, `dev_api_key_change_in_prod` и др.).

5.2. Базовые (все сервисы)

Переменная	По умолчанию
`REDIS_URL`	`redis://localhost:6379/0` (в Docker — `redis://:${REDIS_PASSWORD}@redis:6379/0`)
`APP_ENV`	`development` — в промышленном контуре **обязательно** `production`
`LOG_LEVEL`	`INFO`
`MINIO_ENDPOINT` / `MINIO_ACCESS_KEY` / `MINIO_SECRET_KEY`	`localhost:9100` / `minioadmin` / `minioadmin`
`KEYCLOAK_URL` / `KEYCLOAK_PUBLIC_URL` / `KEYCLOAK_REALM` / `KEYCLOAK_CLIENT_ID`	`http://localhost:8080` / — / `panda` / `panda-backend`
`CENTRIFUGO_URL`	`http://localhost:8100`

5.3. api-gateway

`ALLOWED_ORIGINS`, `REGISTRATION_OPEN` (по умолчанию `false` — открытая саморегистрация), `RATE_LIMIT` (`120/minute`), `WEBHOOK_BASE_URL` (публичный адрес приёма веб-хуков — **обязателен для автоматической регистрации веб-хука Telegram**), `BILLING_SERVICE_URL`, `AGENT_ENGINE_URL`, `APP_URL`, `PLATFORM_ADMIN_EMAILS`, `DEV_NO_AUTH` (**только разработка**; старт при `APP_ENV=production` блокируется), `TELEGRAM_BOT_TOKEN`, `TELEGRAM_LEAD_CHAT_ID`, `TELEGRAM_PROXY` (обращение к `api.telegram.org` с российского сервера требует прокси).

5.4. agent-engine (языковые модели)

Переменная	Назначение
`YANDEX_API_KEY`	**основной ключ.** Сервисный API-ключ Yandex Cloud. Требуется и для генерации ответов, **и для базы знаний**
`YANDEX_FOLDER_ID`	идентификатор каталога Yandex Cloud
`YANDEX_IAM_TOKEN`	альтернатива API-ключу для генерации (**для базы знаний не подходит** — требуется именно `YANDEX_API_KEY`)
`GIGACHAT_CREDENTIALS`	ключ авторизации GigaChat (резервный поставщик)
`GIGACHAT_SCOPE`	`GIGACHAT_API_PERS` / `_CORP` / `_B2B`
`GIGACHAT_VERIFY_SSL`	в промышленном контуре — `true` (требуется установка корневого сертификата Минцифры)
`DEFAULT_LLM_PROVIDER`	`yandexgpt` (значение по умолчанию)
`DEFAULT_LLM_MODEL`	модель по умолчанию (`yandexgpt-lite`)

Значение `DEFAULT_LLM_PROVIDER` по умолчанию — `yandexgpt` (российский поставщик-резидент); шаблоны `.env` содержат `YANDEX_API_KEY` как основной ключ.

152-ФЗ. Поддерживаются только поставщики — резиденты РФ (`yandexgpt`, `gigachat`). Иностранные поставщики из состава ПО удалены и в клиент не разрешаются; передача персональных данных в промптах зарубежным поставщикам исключена на уровне кода.

5.5. Прочие сервисы

connector-service: `TELEGRAM_BOT_TOKEN`, `VK_SERVICE_KEY`, `VK_CONFIRMATION_TOKEN`, `VK_SECRET`, `WHATSAPP_TOKEN`, `WEBHOOK_SECRET`, `SALUTESPEECH_CREDENTIALS`, `SALUTESPEECH_SCOPE`, `SALUTESPEECH_VERIFY_SSL` — все необязательные.

billing-service: `YOOKASSA_SHOP_ID`, `YOOKASSA_SECRET_KEY` (**при пустых значениях включается демонстрационный режим с имитацией платежей** — в промышленном контуре обязательны), `YOOKASSA_ALLOWED_IPS`, `YOOKASSA_VAT_CODE` (ставка НДС в чеке; `1` — «без НДС»).

notification-service: `SMTP_HOST` (`smtp.yandex.ru`), `SMTP_PORT` (`465`), `SMTP_USER`, `SMTP_PASSWORD`, `EMAIL_FROM`, `DASHBOARD_URL`.

Keycloak, вход через российские сервисы:
`YANDEX_CLIENT_ID`/`_SECRET`/`KC_IDP_YANDEX_ENABLED`, `VK_CLIENT_ID`/`_SECRET`/`KC_IDP_VK_ENABLED`, `MAILRU_CLIENT_ID`/`_SECRET`/`KC_IDP_MAILRU_ENABLED`. При пустых ключах провайдер отключается. Настройка — `make social-setup`.

Переменные времени сборки frontend. Переменные `VITE_*` (кабинет) и `NEXT_PUBLIC_*` (сайт) **встраиваются в код при сборке образа**. Их изменение требует **пересборки** соответствующего образа, перезапуск контейнера не даёт эффекта.

---

6. Эксплуатация

6.1. Основные команды

```
make help                    # перечень команд
make verify                  # проверка состояния стека (scripts/verify_st
make stack-logs              # журналы всех сервисов
docker compose logs -f api-gateway
make migrate                 # применение миграций вручную
make monitoring              # включить профиль мониторинга
```

6.2. Обновление версии

 Ключевая особенность. Пять прикладных сервисов используют **общий образ** `panda-app:dev`. Команда `docker compose up -d` (в том числе с `--build`) **не гарантирует** пересоздание контейнеров при общем образе — сервисы могут продолжить работу на старом коде.

Корректный порядок обновления кода Python:

```
make redeploy                        # пересборка образа + принудительное
make redeploy s=connector-service    # только один сервис
make verify-code SVCS="api-gateway"  # проверка: контейнер работает на акт
```

`make redeploy` выполняет пересборку, пересоздание с `--force-recreate` и автоматически запускает `verify-code`, который сравнивает идентификатор образа с идентификатором, на котором фактически запущен контейнер.

Frontend собирается отдельно: образы `panda-web:dev` и `panda-site:dev` (`docker compose ... build web` / `build site`, затем `up -d --force-recreate --no-deps ...`).

Рекомендуемый порядок обновления: 1. Резервная копия БД (раздел 7). 2. Обновление исходных файлов. 3. `make redeploy` (миграции применяются задачей `migrate` при старте). 4. `make verify` и проверка `/health` всех сервисов.

6.3. Ротация секретов

Процедура — `docs/runbooks/secret-rotation.md`.

Ротация `ENCRYPTION_KEY` требует перешифрования данных (`channels.credentials`, `crm_integrations.credentials`):

```
# 1. Резервная копия БД. 2. Остановить все 5 прикладных сервисов.
OLD_ENCRYPTION_KEY=... NEW_ENCRYPTION_KEY=... DATABASE_URL=... \
  python scripts/rotate_encryption_key.py            # проверочный прогон
OLD_ENCRYPTION_KEY=... NEW_ENCRYPTION_KEY=... DATABASE_URL=... \
  python scripts/rotate_encryption_key.py --apply    # применение — ровно
# 3. Обновить секрет. 4. Запустить сервисы. 5. Проверить /health.
```

Альтернатива без простоя — `ENCRYPTION_KEYS=новый,старый` (MultiFernet: запись новым ключом, чтение — новым и старым).

`INTERNAL_API_TOKEN` необходимо менять **одновременно на всех сервисах**, иначе межсервисные вызовы вернут 401. Секреты Centrifugo должны совпадать с обеих сторон.

6.4. Управление платформой

Роль `platform_admin` (realm-роль Keycloak либо перечень адресов в `PLATFORM_ADMIN_EMAILS`) даёт доступ к разделу «Админка»: просмотр тенантов и пользователей, сводные показатели, блокировка (заморозка) тенанта. Подробнее — `docs/ADMIN.md`.

---

7. Резервное копирование и восстановление

7.1. Регулярное копирование

Скрипт `scripts/backup.sh`.

Параметр	По умолчанию
`BACKUP_DIR`	`./backups`
`PG_CONTAINER` / `PG_USER` / `PG_DATABASES`	`panda-postgres` / `panda` / `panda keycloak`
`MINIO_CONTAINER`	`panda-minio`
`RETENTION_DAYS`	**14**

Что копируется: 1. **PostgreSQL** — `pg_dump -Fc` по каждой БД (`panda`, `keycloak`) в custom-формате. 2. **MinIO** — зеркалирование содержимого (`mc mirror`). 3. Устаревшие копии старше `RETENTION_DAYS` удаляются автоматически.

Регламент (cron):

```
0 3 * * *  cd /opt/panda && ./scripts/backup.sh >> /var/log/panda-backup.log 2>&1
```

 Не входит в состав регулярной копии: ClickHouse (аналитика) и Redis (кеш и очереди). Данные диалогов, база знаний, состояния LangGraph и данные Keycloak хранятся в PostgreSQL и MinIO и копируются. Потеря Redis приводит к потере необработанных событий в очередях, но не к потере сохранённых данных.

7.2. Восстановление

```
# Остановить прикладные сервисы, затем:
./scripts/restore.sh ./backups/postgres/panda_20260610-030000.dump [имя_БД
# внутри: docker exec -i panda-postgres pg_restore -U panda -d panda \
#           --clean --if-exists --no-owner < DUMP
# MinIO — вручную:
mc mirror ./backups/minio/<метка_времени> local
```

7.3. Полная холодная копия

Для переноса на другой сервер применяется скрипт `backup-panda.sh` — он дополнительно сохраняет тома Docker, образы (`docker save`), каталог проекта и контрольные суммы SHA-256. Порядок восстановления — `RESTORE.md`.

Особенности: - Том PostgreSQL восстанавливается только на том же мажорном релизе СУБД (PG 16); при несовпадении используется SQL-дамп `pg_dumpall`. - Keycloak не имеет собственных томов: конфигурация realm импортируется из `infra/keycloak/realm-export.json`, состояние хранится в PostgreSQL.

---

8. Мониторинг

Включается профилем: `make monitoring` (`docker compose --profile monitoring up -d`).

8.1. Метрики

Prometheus (`scrape_interval: 15s`) собирает метрики по именам сервисов Docker (работает и при закрытых портах хоста): `api-gateway:8000`, `agent-engine:8001`, `connector-service:8002`, `billing-service:8003`, `notification-service:8004`, `postgres-exporter:9187`, `redis-exporter:9121`. Путь — `/metrics`. Снаружи `/metrics` закрыт (Caddy отдаёт 403).

Grafana (порт 3001) — папка «Панда», преднастроенные панели `panda-overview`, `panda-demo-funnel`; источник данных Prometheus настраивается автоматически.

8.2. Проверки работоспособности

Сервис	Endpoint
api-gateway	`GET /health`, `GET /ready` (включает проверку БД)
agent-engine	`GET /health` → `{"graph_ready": true}`
connector-service / billing-service / notification-service	`GET /health`

8.3. Правила оповещения

Правило	Условие	Задержка	Важность
`PandaServiceDown`	`up{job=~"api-gateway|agent-engine|connector-service|billing-service|notification-service"} == 0`	2 мин	**critical**
`PandaDLQNotEmpty`	`panda_dlq_depth > 0`	1 мин	**critical**
`PandaConsumerLag`	`panda_group_pending > 100`	5 мин	warning
`PandaStreamBloat`	`panda_stream_length > 100000`	10 мин	warning
`PandaHighErrorRate`	доля ответов 5xx > 5%	5 мин	warning
`PandaHighLatencyP95`	p95 времени ответа > 1 с	10 мин	warning

 Требует настройки. В `infra/monitoring/alertmanager/alertmanager.yml` активны только получатели-веб-хуки n8n (`http://n8n:5678/webhook/alerts`), при этом n8n относится к профилю `tools`, а Alertmanager — к профилю `monitoring`. Получатели для Telegram и электронной почты присутствуют в конфигурации в закомментированном виде. **Без настройки получателя оповещения никуда не доставляются.**

---

9. Безопасность

- **TLS** терминируется на Caddy; сертификаты Let's Encrypt выпускаются автоматически. Издатель принудительно ограничен ACME/Let's Encrypt (`cert_issuer acme`), поскольку ZeroSSL недоступен с серверов, размещённых в РФ. Сертификаты сохраняются в томе `caddy_data` — том не следует удалять (ограничения Let's Encrypt по частоте выпуска).
- **Административный API Caddy отключён** (`admin off`) — исключает переконфигурирование прокси из сети Docker.
- Заголовки безопасности (HSTS, `X-Content-Type-Options`, `Referrer-Policy`, CSP для кабинета) устанавливаются на прокси и дублируются приложением.
- Все секреты хранятся в `.env` и передаются в контейнеры через окружение. **Файл `.env` содержит рабочие секреты — он должен иметь права `600` и не подлежит передаче третьим лицам и включению в общедоступные архивы.**
- Учётные данные каналов и CRM хранятся в БД в зашифрованном виде (Fernet).
- Ограничение частоты запросов — 120/мин по умолчанию, для публичных методов 10–15/мин.
- Журнал действий — таблица `audit_log`.

---

10. Устранение неисправностей

Признак	Вероятная причина и действия
Сервисы не стартуют, задача `migrate` завершилась с ошибкой	Проверьте доступность PostgreSQL (`docker compose ps`, healthcheck) и корректность `DATABASE_URL`. В промышленном слое `migrate` использует отдельный `DATABASE_URL` — он должен быть задан
Сервис падает при старте с `ValueError`	Сработала защитная проверка конфигурации: не задан `ENCRYPTION_KEY`/`INTERNAL_API_TOKEN`, `TRUSTED_PROXY_HOSTS='*'` либо секрет совпадает с демонстрационным значением
`agent-engine: graph_ready: false`	Checkpointer не может подключиться к PostgreSQL. Проверьте `DATABASE_URL` и наличие схемы `langgraph`
Агент не отвечает	Не задан `YANDEX_API_KEY` / `YANDEX_FOLDER_ID`; либо `DEFAULT_LLM_PROVIDER` содержит неподдерживаемое значение (допустимы только `yandexgpt` и `gigachat`)
Документы базы знаний массово получают статус «Ошибка»	Не задан **`YANDEX_API_KEY`** (IAM-токен для эмбеддингов не подходит) либо недоступен MinIO
Telegram-бот не отвечает	Проверьте токен бота в канале и доступность исходящих запросов к Telegram (приём сообщений работает через long-polling; при сетевых ограничениях задаётся `TELEGRAM_PROXY`). Один бот должен быть подключён только к одному агенту
Развёрнут новый код, но поведение прежнее	Контейнер работает на старом образе. Выполните `make redeploy` и `make verify-code`
Изменены `VITE_*` / `NEXT_PUBLIC_*`, эффекта нет	Требуется пересборка образа `web` / `site`
Оплата не проходит, шлюз возвращает 400	Проверьте `ALLOWED_ORIGINS` — адрес возврата отвергается при несовпадении
Не работает WebSocket	Проверьте совпадение `CENTRIFUGO_TOKEN_SECRET` с `CENTRIFUGO_TOKEN_HMAC_SECRET_KEY`, пароль Redis в конфигурации Centrifugo и список `allowed_origins`
Не приходят оповещения мониторинга	Не настроен получатель в Alertmanager (см. 8.3)

Диагностика: `make stack-logs`, `docker compose logs -f <сервис>`. Журналы — структурированные (structlog).

11. Известные особенности текущей версии

1. **Разграничение доступа на уровне строк БД (RLS) не применяется** — изоляция организаций обеспечивается на уровне приложения.
2. **Рекуррентные списания не реализованы**; возврат платежа доступен только на стороне billing-service (метод не опубликован в API-шлюзе).
3. Каталоги `infra/k8s`, `infra/helm`, `infra/terraform` пусты — развёртывание в Kubernetes не поддерживается.