Proto Observability Platform

РУКОВОДСТВО АДМИНИСТРАТОРА

Версия платформы: 203

Обозначение документа: ПРОТО.ОБП.203.И3.2

Вид документа: Руководство администратора

Реестровая запись: № 14369 от 03.08.2022 (единый реестр российских программ для ЭВМ и баз данных)

Листов: 61

Год: 2026

СОДЕРЖАНИЕ

1. Введение

1.1. Область применения

Настоящее руководство определяет порядок развёртывания, настройки, администрирования и восстановления работоспособности программного обеспечения **Proto Observability Platform** (далее — Система, платформа, ПО).

Документ предназначен для системных администраторов, администраторов средств мониторинга, инженеров эксплуатации и администраторов информационной безопасности организации-эксплуатанта.

Порядок работы с пользовательским интерфейсом изложен в документе **«Руководство пользователя»** (ПРОТО.ОБП.203.И3.1).

Документ подготовлен с учётом требований РД 50-34.698-90, ГОСТ 19.503-79 «ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению» и ГОСТ 34.201-2020.

> **О версиях в процедурах установки и обновления.** Функциональные возможности описаны по состоянию на версию 203. Порядок установки (раздел 4) и обновления (п. 8.3) приведён для дистрибутива **версии 202** — последнего, для которого вендором опубликованы состав дистрибутива, перечень образов и инструкция по обновлению. После публикации дистрибутива версии 203 имена файлов (`docker-compose-203.yaml`, `protoobp-backend-dist-singlehost-203.tar.gz`), перечень образов и процедуру обновления следует актуализировать по примечаниям к выпуску 203.

1.2. Уровень подготовки администратора

Администратор Системы должен обладать:

- навыками администрирования ОС семейства Linux (уровень уверенного администратора): управление службами `systemd`, работа с файловой системой, правами доступа, сетевой конфигурацией;
- практическим опытом работы с контейнеризацией: Docker Engine, Docker Compose, реестры образов;
- пониманием принципов работы обратного прокси, TLS-сертификатов и инфраструктуры открытых ключей;
- навыками администрирования служб каталогов (LDAP / Active Directory) и провайдеров единого входа (OIDC) — для настройки SSO;
- пониманием модели данных временных рядов и языка PromQL — для сопровождения правил алертинга;
- навыками администрирования Kubernetes и Helm — при развёртывании агентов в контейнерных средах;
- знанием требований организации в области защиты информации и порядка обработки персональных данных.

1.3. Термины и сокращения

Термины, применяемые в настоящем документе, соответствуют разделу 1.5 «Руководства пользователя». Дополнительно используются:

Термин / сокращение	Расшифровка
Backend (ProtoOBP Backend)	Серверная часть Системы — набор контейнеризованных компонентов сбора, обработки, хранения и представления данных

Термин / сокращение	Расшифровка
Агент (ProtoOBP Agent)	Программный компонент, устанавливаемый на контролируемый узел и осуществляющий сбор метрик, трейсов и логов
Трейсер	Библиотека инструментации, встраиваемая в приложение и формирующая данные распределённой трассировки
Cluster-агент	Компонент агента Kubernetes, собирающий данные уровня кластера
Retention	Срок хранения данных заданного вида
Обфускация	Замена значений, потенциально содержащих конфиденциальные данные, на символ-заполнитель
Realm	Область аутентификации в сервисе `proto-auth`
GMSA	Group Managed Service Account — управляемая учётная запись службы в Active Directory

1.4. Перечень эксплуатационной документации

Обозначение	Наименование
ПРОТО.ОБП.203.ИЗ.1	Руководство пользователя
ПРОТО.ОБП.203.ИЗ.2	Руководство администратора (настоящий документ)
—	Лицензионный сертификат (выдаётся вендором)
—	Описание функциональных характеристик ПО
—	Онлайн-документация вендора
—	Release notes соответствующей версии

2. Назначение и структура Системы

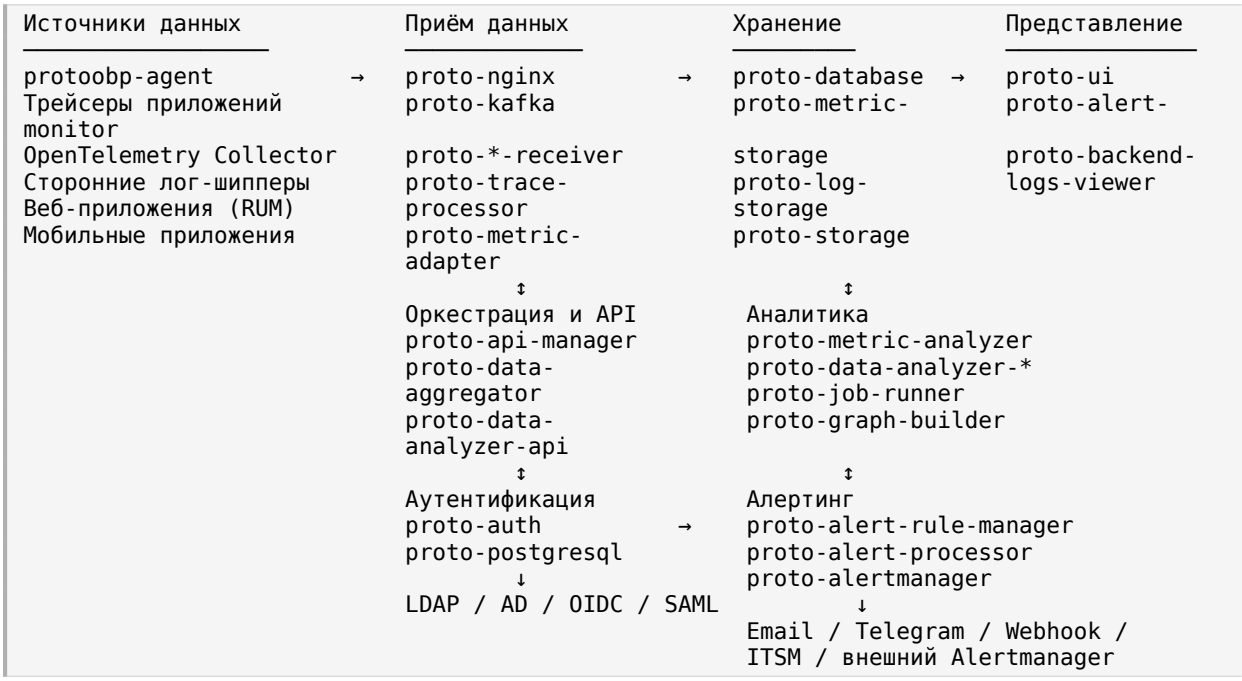
2.1. Назначение

Система предназначена для комплексного наблюдения за состоянием информационных систем организации и обеспечивает функции APM, RUM, распределённой трассировки, сбора и анализа логов, алертинга, мониторинга инфраструктуры и Kubernetes, анализа первопричин, выявления аномалий, контроля SLO и бизнес-показателей.

Система разворачивается **исключительно в контуре Заказчика** (on-premise). Передача собираемых данных во внешние облачные сервисы не производится.

2.2. Общая архитектура

Поток данных в Системе организован следующим образом:



2.3. Состав компонентов серверной части

Компонент	Назначение
`proto-nginx`	Маршрутизация запросов между компонентами Системы, терминирование TLS
`proto-kafka`	Буфер входящих данных
`proto-zookeeper`	Служебная координация
`proto-otel-collector`	Приём данных по протоколу OpenTelemetry
`proto-trace-receiver`	Потребление данных трейсов из буфера и их обработка
`proto-trace-processor`	Обработка трейсов, проверка лицензии
`proto-metric-receiver`	Обработка метрик
`proto-metric-adapter`	Преобразование данных логов и алертов и передача их другим компонентам
`proto-metric-exporter`	Преобразование метрик
`proto-metric-storage`	Хранилище метрик, API записи и чтения

Компонент	Назначение
`proto-log-storage`	Хранилище логов
`proto-database`	Хранилище трейсов (спанов), событий, алертов и конфигурации
`proto-storage`	Хранилище платформенных данных
`proto-postgresql`	Хранилище данных сервиса аутентификации
`proto-data-analyzer-api`, `-refresh`, `-router`, `-worker-1`, `-worker-2`	Семантический слой анализа данных: модели данных и API доступа для интерфейса
`proto-data-aggregator`	GraphQL API для пользовательского интерфейса
`proto-api-manager`	Управление программными интерфейсами
`proto-metric-analyzer`	Расчёт базовых линий метрик и обнаружение аномалий
`proto-job-runner`	Выполнение фоновых заданий
`proto-graph-builder`	Построение графов связей
`proto-alert-rule-manager`	API управления правилами, каналами и политиками алертинга
`proto-alert-processor`	Вычисление правил алертинга
`proto-alertmanager`	Маршрутизация и агрегация алертов
`proto-alert-monitor`	Отображение алертов в реальном времени
`proto-auth`	Аутентификация пользователей, подключения LDAP / OIDC / SAML
`proto-backend-logs-viewer`	API и интерфейс просмотра логов компонентов Системы
`proto-ui`	Основной веб-интерфейс
`protoobp-agent`	Агент, устанавливаемый в том числе на сам сервер серверной части
`proto-incident-manager`	Модуль инцидентов: ИИ-сводка и постмортем (с версии 201)
`proto-ai-agent-investigator`	ИИ-расследование первопричины (начиная с версии 201)
`proto-report-runner`	Планировщик отчётов: расписание, формирование задания, отправка письма получателям (начиная с версии 202)
`proto-report-renderer`	Отрисовка дашборда в PDF-документ для отчёта (начиная с версии 202)

Все компоненты серверной части поставляются в виде Docker-образов и управляются средствами `docker compose`.

2.4. Варианты развёртывания

Вариант	Описание	Применение
Один сервер	Все компоненты на одном физическом или виртуальном хосте	Пилотные проекты, небольшие промышленные установки. Описан в настоящем документе
Несколько серверов	Компоненты разнесены по нескольким хостам для повышения производительности и отказоустойчивости	Крупные промышленные установки. Конфигурация предоставляется вендором по запросу

3. Требования к техническим и программным средствам

3.1. Требования к серверу серверной части (один хост)

Параметр	Минимальное значение
Оперативная память	24 ГБ
Процессор	12 vCPU
Дисковая подсистема	500 ГБ SSD, не менее 5000 IOPS
Операционная система	Linux (Debian-based или CentOS 8+)
Docker Engine	24.x — 28.x
Docker Compose	v2.24.5 и выше
Доступ в сеть Интернет	Требуется для загрузки образов (либо применяется offline-установка)
Открытые порты	80, 443

> **Замечание по совместимости процессора.** При отсутствии поддержки инструкций AVX компоненты `proto-data-analyzer-router` и `proto-data-analyzer-worker-*` не запускаются. В этом случае применяются образы с суффиксом `-non-avx` (см. п. 10.2).

3.2. Требования к контролируемым узлам

Агент функционирует на всех основных DEB- и RPM-дистрибутивах Linux, а также на Windows.

Поддерживаемые ОС Linux: Astra Linux, Alt Linux, ROSA Linux, РЕД ОС (сертифицированная и стандартная редакции), Alpine Linux, Amazon Linux, CentOS, CentOS Stream, Debian, Fedora, Google Container-Optimized OS, Oracle Linux, Red Hat Enterprise Linux, Red Hat Enterprise Linux CoreOS, Rocky Linux, Ubuntu.

Поддерживаемые ОС Windows: Windows Desktop, Windows Server.

Контейнерные среды: Docker, CRI-O, containerd.

Оркестраторы: Kubernetes, OpenShift.

Языки и платформы автоматической инструментации: Java, PHP, Node.js, Python, .NET / .NET Core, Go. Дополнительно поддерживаются OpenTelemetry (включая zero-code инструментацию eBPF) и подключение приложений с трейсерами сторонних вендоров.

3.3. Требования к сетевому взаимодействию

Источник	Назначение	Порт / протокол	Описание
Сервер серверной части	Репозиторий вендора `registry.git.proto.group`	443/tcp	Загрузка дистрибутивов и образов
Контролируемые серверы (внутренняя сеть)	Сервер серверной части	80/tcp, 443/tcp	Передача данных агентами
Рабочие места пользователей	Сервер серверной части	80/tcp, 443/tcp	Доступ к веб-интерфейсу
Приложения на контролируемом сервере	Агент на этом же сервере (localhost)	8125/tcp, 8126/tcp, 8125/udp	Передача трейсов и метрик от трейсеров агенту
Точка сбора трафика	Сервер серверной части	443/tcp	Телеметрия от браузеров и

Источник	Назначение	Порт / протокол	Описание
RUM			мобильных устройств; трафик должен быть проксирован

Дополнительно для агента Kubernetes:

Источник	Назначение	Порт / протокол
Поды агента (`protoobp`, `protoobp-cluster-agent`)	Серверная часть	443/tcp
Под `protoobp`	Под `protoobp-cluster-agent`	5005/tcp
Поды приложений	Под агента `protoobp`	8125/udp (метрики), 8126/tcp (трейсы)
Control plane Kubernetes	Ноды кластера	8000/tcp (webhook автоматической инструментации)

3.4. Требования к рабочему месту администратора

- Терминальный доступ (SSH) к серверу серверной части и к контролируемым узлам;
- права `sudo` либо доступ к привилегированной учётной записи на сервере серверной части;
- веб-браузер для работы с интерфейсом Системы (требования — см. «Руководство пользователя», п. 2.2.1);
- при администрировании Kubernetes — настроенный `kubectl` и `helm` версии 3.x.

> `helm` версии 2.x не поддерживается — формат чартов отличается.

---

4. Установка Системы

4.1. Подготовительные мероприятия

1. Выделить сервер в соответствии с п. 3.1.
2. Обеспечить сетевые доступы в соответствии с п. 3.3.
3. Получить от вендора или компании-партнёра **лицензионный сертификат**. Сертификат содержит: имя компании для лицензии, лицензионный ключ, учётные данные (логин/имя пользователя и пароль/токен) для доступа к репозиторию артефактов.
4. Определить и зарегистрировать в DNS имя хоста, по которому агенты будут обращаться к серверной части, и имя, по которому пользователи будут обращаться к веб-интерфейсу (при их различии).
5. При использовании собственного TLS-сертификата — выпустить сертификат, у которого `Common Name` совпадает с именем, указываемым в параметре `UI_URL`.
6. Установить Docker Engine и Docker Compose требуемых версий.

4.2. Установка с использованием скрипта (рекомендуемый способ)

Способ применим при следующих ограничениях: установка на одном сервере (пилот или небольшая промышленная установка); загрузка образов выполняется напрямую из репозитория вендора.

```
curl -sSL https://docs.proto.group/setup_protoobp.sh -o setup_protoobp.sh \
  && chmod +x setup_protoobp.sh \
  && sudo ./setup_protoobp.sh
```

В интерактивном режиме введите данные из лицензионного сертификата и дождитесь завершения работы скрипта. При отдельных ошибках скрипт перезапускается самостоятельно.

> **Мера предосторожности.** Перед запуском скрипта, загружаемого из сети, убедитесь в подлинности источника в соответствии с регламентом организации. Рекомендуется предварительно сохранить скрипт, проверить его содержимое и только после этого выполнить.

4.3. Установка без использования скрипта

Команды выполняются от имени привилегированной учётной записи (`sudo su`).

Шаг 1. Загрузка и распаковка дистрибутива

```
curl --header "PRIVATE-TOKEN:<your_token>" \
"https://git.proto.group/api/v4/projects/125/packages/generic/protoobp-backend-dist/202/
protoobp-backend-dist-singlehost-202.tar.gz" \
--output protoobp-backend-dist-singlehost-202.tar.gz
```

Значение `<your_token>` — пароль из лицензионного сертификата.

```
mkdir -p /opt/protoobp && tar -xzvf protoobp-backend-dist-singlehost-202.tar.gz -C
/opt/protoobp
```

Шаг 2. Настройка параметров окружения

```
cd /opt/protoobp && mv env_template-202 .env
```

Задайте в файле `.env` обязательные параметры:

Параметр	Описание
`POBP_LICENSE_COMPANY_NAME`	Название компании-владельца лицензии из поля «Имя компании для лицензии» (кириллицей)
`POBP_LICENSE_KEY`	Лицензионный ключ. После копирования удалите все символы переноса строк

Параметр	Описание
	— значение должно быть в одну строку
`POBP_COMPUTE`	Имя хоста, доступное для подключения агентов (например, FQDN во внутренней сети)
`UI_URL`	Адрес доступа к веб-интерфейсу, если отличается от `POBP_COMPUTE`. Указывается с префиксом `https://`. **Недопустимы значения** `127.0.0.1`, `localhost`, `0.0.0.0`
`POBP_DATA_DIRECTORY`	Путь к каталогу хранения данных, если отличается от `/opt/protoobp/data`. Указанный каталог должен быть создан вручную

Примеры:

```
POBP_LICENSE_COMPANY_NAME=Супер Корпорация
POBP_LICENSE_KEY=b916e441450f...
POBP_COMPUTE=proto-compute.mycompany.local
UI_URL=https://my.proto.backend.local
```

Параметры подключения внешней языковой модели (версия 201). Требуются для работы ИИ-функций модуля «Инциденты». Без них платформа запустится, но ИИ-функции будут отключены:

Параметр	Описание
`POBP_AI_LLM_BASE_URL`	Базовый адрес внешней модели (OpenAI-совместимый протокол)
`POBP_AI_LLM_API_KEY`	Ключ доступа, передаётся как `Authorization: Bearer`. Является секретом, в журналы не выводится
`POBP_AI_LLM_MODEL`	Имя модели без префикса провайдера

Эндпоинт используется сервисами `proto-incident-manager` (ИИ-сводка и постмортем) и `proto-ai-agent-investigator` (ИИ-расследование первопричины), а начиная с версии 203 — также разделом **«AI-аналитик»** (чат по данным платформы на естественном языке). Без подключённой модели все перечисленные функции недоступны; платформа при этом работает штатно.

Параметры почтового сервера отчётности (начиная с версии 202). В версии 202 добавлены два сервиса отчётности — `proto-report-runner` и `proto-report-renderer`. Чтобы отчёты доставлялись получателям, задайте:

Параметр	Описание
`POBP_REPORT_RUNNER_SMTP_HOST`	Адрес почтового сервера
`POBP_REPORT_RUNNER_SMTP_USER`	Учётная запись отправителя
`POBP_REPORT_RUNNER_SMTP_PASSWORD`	Пароль отправителя (секрет)

Других обязательных переменных нет: порт (587), режим TLS, адрес отправителя, таймауты и интервалы имеют рабочие значения по умолчанию. Без параметров почтового сервера платформа запускается штатно, но отчёты отправляться не будут — запуски завершатся с ошибкой, видимой в истории запусков отчёта.

Параметры почтового сервера отчётности **не связаны** с параметрами канала оповещения по электронной почте (см. п. 6.4) и с параметрами SMTP для приглашений пользователей (п. 6.4) — это отдельный сервис с собственной конфигурацией.

> **Требование безопасности.** Служебная учётная запись отчётности заводится автоматически при старте, но её пароль одинаков во всех инсталляциях. В промышленной эксплуатации его следует перевыпустить и указать новое значение в переменной `POBP_REPORT_RUNNER_KEYCLOAK_CLIENT_SECRET`.

> **Требование информационной безопасности.** Подключение внешней языковой модели означает передачу за пределы контура организации фрагментов телеметрии (описаний алертов, наименований сервисов и ресурсов). Перед включением ИИ-функций согласуйте это с подразделением информационной безопасности и, при обработке сведений ограниченного доступа, используйте языковую модель, размещённую в контуре организации.

Шаг 3. Создание каталогов хранения данных

Команды приведены для значения `POBP_DATA_DIRECTORY` по умолчанию:

```
mkdir -p /opt/protoobp/data/database && chown -R 101:101 /opt/protoobp/data/database
mkdir -p /opt/protoobp/data/opensearch && chown -R 1000:1000
/opt/protoobp/data/opensearch
mkdir -p /opt/protoobp/data/logs && chown -R 1000:1000 /opt/protoobp/data/logs
mkdir -p /opt/protoobp/data/metrics && chown -R 1000:1000 /opt/protoobp/data/metrics
mkdir -p /opt/protoobp/data/kafka && chown -R 65532:65532 /opt/protoobp/data/kafka
mkdir -p /opt/protoobp/data/zookeeper && chown -R 1001:1001 /opt/protoobp/data/zookeeper
```

Шаг 4. Создание файлов конфигурации алертинга

```
touch /opt/protoobp/alertrules.yml /opt/protoobp/alertpolicies.yml && \
chown 65532:65532 /opt/protoobp/alertrules.yml /opt/protoobp/alertpolicies.yml
```

Шаг 5. Загрузка образов и запуск

```
docker login --username <USERNAME> registry.git.proto.group
cd /opt/protoobp
docker compose -f docker-compose-202.yaml pull
docker compose -f docker-compose-202.yaml up -d
```

> **Штатное поведение при первом запуске.** Компоненты `proto-storage`, `proto-data-aggregator`, `proto-trace-processor` и `proto-metric-receiver` при первом запуске выполняют инициализацию и перезапускаются, что приводит к выводу ошибки. Прервите текущую команду (`Ctrl+C`) и повторите `docker compose ... up -d`. Полный запуск занимает 2–4 минуты.

Шаг 6. Проверка запуска

```
docker ps -a --format "table {{.Names}}\t{{.Status}}" | sort
```

Все компоненты должны находиться в состоянии `Up`; компоненты, для которых определены проверки, — в состоянии `healthy`. Циклические перезапуски недопустимы.

Шаг 7. Первый вход

Откройте в браузере адрес, указанный в `UI_URL`.

Стандартная учётная запись: логин `admin`, пароль `protoobp`.

> **Обязательное требование безопасности.** Немедленно после первого входа смените пароль встроенной учётной записи администратора. Эксплуатация Системы с паролем по умолчанию недопустима.

4.4. Установка при отсутствии доступа в сеть Интернет (offline)

4.4.1. Подготовка на промежуточной машине

На машине с доступом к репозиторию вендора выполните авторизацию и загрузку образов:

```
docker login --username <USERNAME> registry.git.proto.group
```

Сохраните и выполните скрипт, формирующий единый архив образов:

```
#!/usr/bin/env bash
set -euo pipefail

REGISTRY="registry.git.proto.group/protoobp/protoobp-artifacts"
TAG="202"
ARCHIVE="protoobp-images-${TAG}.tar.gz"

IMAGES=(
  "proto-ai-agent-investigator:${TAG}" "proto-alert-monitor:${TAG}" "proto-alert-
processor:${TAG}"
  "proto-alert-rule-manager:${TAG}" "proto-alertmanager:${TAG}" "proto-api-manager:$
{TAG}"
  "proto-auth:${TAG}" "proto-backend-logs-viewer:${TAG}" "proto-data-aggregator:${TAG}"
  "proto-data-analyzer-api:${TAG}" "proto-data-analyzer-refresh:${TAG}"
  "proto-data-analyzer-router:${TAG}" "proto-data-analyzer-router:${TAG}-non-avx"
  "proto-data-analyzer-worker-1:${TAG}" "proto-data-analyzer-worker-1:${TAG}-non-avx"
  "proto-data-analyzer-worker-2:${TAG}" "proto-data-analyzer-worker-2:${TAG}-non-avx"
  "proto-database:${TAG}" "proto-graph-builder:${TAG}" "proto-incident-manager:${TAG}"
  "proto-job-runner:${TAG}" "proto-kafka:${TAG}" "proto-log-storage:${TAG}"
  "proto-metric-adapter:${TAG}" "proto-metric-analyzer:${TAG}" "proto-metric-exporter:$
{TAG}"
  "proto-metric-receiver:${TAG}" "proto-metric-storage:${TAG}" "proto-nginx:${TAG}"
  "proto-otel-collector:${TAG}" "proto-postgresql:${TAG}"
  "proto-report-renderer:${TAG}" "proto-report-runner:${TAG}"
  "proto-storage:${TAG}" "proto-trace-processor:${TAG}" "proto-trace-receiver:${TAG}"
  "proto-ui:${TAG}" "proto-zookeeper:${TAG}" "protoobp-agent:${TAG}"
)

FULL=()
for img in "${IMAGES[@]}"; do FULL+=("${REGISTRY}/${img}"); done

for img in "${FULL[@]}"; do docker pull "${img}"; done
docker save "${FULL[@]}" | gzip > "${ARCHIVE}"
echo "Готово: ${ARCHIVE}"
```

Дополнительно загрузите архив дистрибутива (`docker-compose-202.yaml`, шаблон `.env` и др.).

4.4.2. Перенос и загрузка на целевом сервере

```
scp protoobp-images-202.tar.gz protoobp-backend-dist-singlehost-202.tar.gz \
  user@protoobp-backend:/opt/protoobp/
```

На целевом сервере:

```
docker load -i /opt/protoobp/protoobp-images-202.tar.gz
docker images | grep protoobp-artifacts
```

4.4.3. Завершение установки

Далее выполняется установка по п. 4.3 со следующими отличиями:

- архив дистрибутива уже перенесён — распакуйте его в `/opt/protoobp`, повторная загрузка не требуется;
- шаги `docker login` и `docker compose ... pull` **пропускаются**;
- параметр `DOCKER_REGISTRY` в файле `.env` **не изменяется**, чтобы имена образов совпадали с указанными в `docker-compose-202.yaml`;
- запуск выполняется командой `docker compose -f docker-compose-202.yaml up -d`.

4.5. Использование локального реестра Docker-образов

Параметры подключения к репозиторию вендора:

- адрес репозитория, используемый в конфигурации по умолчанию: `registry.git.proto.group/protoobp/protoobp-artifacts`;
- адрес для настройки proxy-репозитория: `https://registry.git.proto.group`;
- Project ID: `protogroup/protoobp-artifacts`;
- требуется сетевой доступ от локального реестра к `https://git.proto.group` (аутентификация) и `https://registry.git.proto.group` (загрузка образов).

jFrog Artifactory. `Administration → Repositories → Add Repositories → Remote Repository`, тип `Docker`. Укажите `Repository Key` (например, `protoobp-proxy`), `URL = https://registry.git.proto.group/`, `Project ID = protogroup/protoobp-artifacts`, имя пользователя и токен из лицензионного сертификата. Затем в `.env`:

```
DOCKER_REGISTRY="artifactory.local/protoobp-proxy"
```

Sonatype Nexus. `Repository → Repositories → Create repository`, рецепт `docker (proxy)`. Укажите имя, `Remote storage = https://registry.git.proto.group`, тип аутентификации `Username` с реквизитами из сертификата. Затем в `.env`:

```
DOCKER_REGISTRY="nexus.local/protoobp/protoobp-artifacts"
```

Перед первым использованием реестра может потребоваться выполнить `docker login`.

4.6. Подключение собственного HTTPS-сертификата

По умолчанию используется самоподписанный сертификат. Для подключения собственного:

1. Выпустите сертификат, у которого `Common Name` совпадает с именем домена из `UI_URL`. Перенесите файлы сертификата и ключа в формате `.pem` на сервер.
2. В файле `/opt/protoobp/.env` раскомментируйте и задайте:

```
POBP_SSL_CERT=/opt/protoobp/ssl/cert.pem
POBP_SSL_CERT_KEY=/opt/protoobp/ssl/cert.key
```

3. В `docker-compose-202.yaml` раскомментируйте секции `volumes` у сервисов `proto-nginx` и `proto-auth`:

```
proto-nginx:
  volumes:
    - ${POBP_SSL_CERT}:/home/nginx/cert.pem
    - ${POBP_SSL_CERT_KEY}:/home/nginx/cert.key

proto-auth:
  volumes:
    - ${POBP_SSL_CERT}:/home/nonroot/cert.pem
    - ${POBP_SSL_CERT_KEY}:/home/nonroot/cert.key
```

4. Примените изменения:

```
docker compose -f docker-compose-202.yaml stop proto-nginx proto-auth
docker compose -f docker-compose-202.yaml rm proto-nginx proto-auth
docker volume rm protoobp_postgres_data
docker compose -f docker-compose-202.yaml up -d
```

> **Внимание.** Команда `docker volume rm protoobp_postgres_data` удаляет том с данными сервиса аутентификации: локальные учётные записи, созданные во встроенном провайдере, будут утрачены. Выполняйте её на этапе первоначальной настройки либо после резервного копирования тома (п. 8.4). Для установки, находящейся в промышленной эксплуатации, порядок замены сертификата согласуйте со службой технической поддержки вендора.

4.7. Установка на нескольких серверах

Развёртывание серверной части на нескольких серверах для повышения отказоустойчивости и распределения нагрузки поддерживается. Инструкции и конфигурационные файлы предоставляются вендором по запросу.

5. Установка и настройка агентов

5.1. Назначение агента

Агент устанавливается на контролируемый узел и обеспечивает автоматический сбор метрик, трейсов и логов с последующей передачей на серверную часть. Сбор трейсов приложений выполняется трейсерами, автоматически инструментирующими код на языках Java, PHP, Node.js, Python, .NET / .NET Core, Go. Сбор метрик выполняется автоматически для всех поддерживаемых технологий.

5.2. Агент для Linux

5.2.1. Установка

Актуальная версия агента — **7.80.2**.

DEB-дистрибутивы (Ubuntu, Debian, Astra Linux и др.):

```
curl --header "PRIVATE-TOKEN:<your_token>" \
"https://git.proto.group/api/v4/projects/125/packages/generic/protoobp-agent/7.80.2/
protoobp-agent_7.80.2-1_amd64.deb" \
--output protoobp-agent_7.80.2-1_amd64.deb
apt install ./protoobp-agent_7.80.2-1_amd64.deb
```

RPM-дистрибутивы (CentOS, Alt Linux, ROSA Linux, РЕД ОС и др.):

```
curl --header "PRIVATE-TOKEN:<your_token>" \
"https://git.proto.group/api/v4/projects/125/packages/generic/protoobp-agent/7.80.2/
protoobp-agent-7.80.2-1.x86_64.rpm" \
--output protoobp-agent-7.80.2-1.x86_64.rpm
rpm -ivh protoobp-agent-7.80.2-1.x86_64.rpm
```

При возникновении на Alt Linux или ROSA Linux ошибки вида `Неудовлетворенные зависимости: glibc-common нужен для protoobp-agent...` установка выполняется командой:

```
rpm -ivh --nodeps protoobp-agent-7.80.2-1.x86_64.rpm
```

Проверка установки:

```
protoobp-agent version
```

5.2.2. Конфигурация

Основной файл конфигурации — `/etc/protoobp-agent/protoobp.yaml`. Минимальная конфигурация:

```
api_key: my_api_key
pobp_url: http://protoobp-backend
apm_config:
  apm_non_local_traffic: true
  apm_pobp_url: http://protoobp-backend
  telemetry:
    enabled: false
agent_telemetry:
  enabled: false
process_config:
  enabled: false
  process_collection:
    enabled: false
  container_collection:
    enabled: false
  process_discovery:
    enabled: false
```

где `api_key` — токен из лицензионного сертификата, `pobp_url` и `apm_pobp_url` — адрес сервера серверной части.

> Схему `https` используйте только в том случае, если на сервере серверной части подключён доверенный сертификат. Полный пример конфигурации приведён в файле `protoobp.yaml.example`.

При наличии на узле Docker-контейнеров добавьте служебного пользователя агента в группу `docker`:

```
usermod -a -G docker pobp-agent
```

При инструментации приложений, работающих в контейнерах, в конфигурации трейсера укажите `POBP_AGENT_HOST=host.docker.internal`, а в `docker-compose.yaml` контейнера приложения добавьте:

```
extra_hosts:
  - "host.docker.internal:host-gateway"
```

5.2.2А. Обновление агента

Обновление выполняется установкой пакета новой версии поверх прежней — удалять предыдущую версию не требуется. Файл `/etc/protoobp-agent/protoobp.yaml` и содержимое каталога `conf.d/` при обновлении сохраняются, служба перезапускается автоматически.

> **Перед обновлением** сохраните копию конфигурации — это позволит быстро вернуть настройки при откате:
>
> ```
> sudo cp -a /etc/protoobp-agent /etc/protoobp-agent.bak
> ```

DEB-дистрибутивы:

```
curl --header "PRIVATE-TOKEN:<your_token>" \
"https://git.proto.group/api/v4/projects/125/packages/generic/protoobp-agent/7.80.2/
protoobp-agent_7.80.2-1_amd64.deb" \
--output protoobp-agent_7.80.2-1_amd64.deb
apt install ./protoobp-agent_7.80.2-1_amd64.deb
```

RPM-дистрибутивы:

```
curl --header "PRIVATE-TOKEN:<your_token>" \
"https://git.proto.group/api/v4/projects/125/packages/generic/protoobp-agent/7.80.2/
protoobp-agent-7.80.2-1.x86_64.rpm" \
--output protoobp-agent-7.80.2-1.x86_64.rpm
rpm -Uvh protoobp-agent-7.80.2-1.x86_64.rpm
```

> **Внимание.** Для RPM используется именно `rpm -Uvh` (upgrade), а не `rpm -ivh` (install): команда `-ivh` на уже установленном агенте завершится ошибкой «package is already installed». При ошибке о неудовлетворённых зависимостях `glibc-common` на Alt Linux или ROSA Linux добавьте ключ `--nodeps`.

Проверка результата обновления:

```
protoobp-agent version
systemctl status protoobp-agent
sudo protoobp-agent status
```

5.2.2Б. Удаление агента

Возможны два сценария: удаление с сохранением конфигурации (например, перед переустановкой) и полное удаление вместе с настройками и журналами.

DEB-дистрибутивы:

```
# удалить агента, сохранив /etc/protoobp-agent (конфигурацию и conf.d)
sudo apt remove protoobp-agent
```

```
# полное удаление: пакет, конфигурация, журналы, служебный пользователь pobp-agent
sudo apt purge protoobp-agent
```

RPM-дистрибутивы:

```
sudo rpm -e protoobp-agent
```

Что удаляется в каждом случае:

Объект	`apt remove` / `rpm -e`	`apt purge`
Пакет и файлы в `/opt/protoobp-agent`	удаляются	удаляются
Юниты `systemd` `protoobp-agent*.service`	удаляются	удаляются
Конфигурация `/etc/protoobp-agent/`	**сохраняется**	удаляется
Журналы `/var/log/protoobp/`	**сохраняются**	удаляются
Служебный пользователь `pobp-agent`	**сохраняется**	удаляется

Для полной очистки системы после `rpm -e` или `apt remove` удалите оставшиеся каталоги и учётную запись вручную:

```
sudo rm -rf /etc/protoobp-agent /opt/protoobp-agent /var/log/protoobp
sudo userdel pobp-agent
```

5.2.3. Управление службой

Агент выполняется как служба `systemd`:

Действие	Команда
Запуск	`systemctl start protoobp-agent`
Остановка	`systemctl stop protoobp-agent`
Перезапуск	`systemctl restart protoobp-agent`
Состояние	`systemctl status protoobp-agent`

Системы без systemd (SysV-init). На дистрибутивах без `systemd` (RHEL 6.x, CentOS 6, ранние линейки Alt Linux, ROSA Linux, РЕД ОС) пакет агента не разворачивает SysV-скрипты: команды `systemctl` недоступны, а `service protoobp-agent start` возвращает `unrecognized service`. В этом случае применяются init-скрипты, приведённые в документации вендора, — для основного агента (`/etc/init.d/protoobp-agent`) и, при использовании APM, для trace-агента (`/etc/init.d/protoobp-agent-trace`). Пути и аргументы в скриптах должны дословно соответствовать unit-файлам из пакета.

5.2.4. Проверка

После запуска агента новый хост появляется в разделе `Инфраструктура` → `Хосты` в течение 1–2 минут.

> Для отображения нового хоста необходимо **обновить страницу браузера целиком**. Кнопка «Обновить» в интерфейсе обновляет только значения метрик, но не список объектов.

5.3. Агент для Windows

5.3.1. Установка

Загрузите MSI-пакет установщика:

```
$headers = @{ "PRIVATE-TOKEN" = "<YOUR-TOKEN>" }
Invoke-WebRequest `
    -Uri "https://git.proto.group/api/v4/projects/125/packages/generic/protoobp-agent/
7.80.2/protoobp-agent-7.80.2-x64.msi" `
    -Headers $headers `
    -OutFile "protoobp-agent-7.80.2-x64.msi" `
    -ErrorAction Stop
```

Выполните установку MSI-пакета. Для сценариев с доменной учётной записью службы поддерживается использование **GMSA** (Group Managed Service Account).

5.3.2. Конфигурация и управление

- Основной файл конфигурации: `%ProgramData%\Protoobp\protoobp.yaml`
- Каталог конфигураций модулей: `%ProgramData%\Protoobp\conf.d`
- Управление службой выполняется штатными средствами Windows (оснастка «Службы», `sc`, PowerShell).

Типовые проблемы:

Проблема	Действие
Служба не запускается	Проверить корректность `protoobp.yaml` и права учётной записи службы; изучить журнал установки `%TEMP%\MSI*.LOG` и рабочие журналы `C:\ProgramData\Protoobp\logs`
Ошибка «Сервис помечен для удаления»	Закрыть оснастку «Службы» и все консоли, обращающиеся к службе; при сохранении ошибки — перезагрузить узел и повторить установку

5.4. Агент в Docker

Агент может быть запущен как отдельный контейнер, в составе `docker compose` либо как служба **Docker Swarm** (глобальный режим — по одному экземпляру агента на каждый узел кластера). Конфигурация задаётся переменными окружения контейнера и соответствует параметрам файла `protoobp.yaml`. Порядок запуска для каждого варианта приведён в документации вендора.

5.5. Агент в Kubernetes

5.5.1. Установка

После установки в кластере появляются поды `protoobp` и `protoobp-cluster-agent`.

1. Добавьте helm-репозиторий:

```
helm repo add --username <my_login> --password <my_password> \
  protoobp https://git.proto.group/api/v4/projects/125/packages/helm/stable
```

2. Создайте namespace и secret для загрузки образов:

```
login="my_login"
password="my_password"
registry="registry.git.proto.group"
namespace="protoobp"

set -o errexit
kubectl create namespace ${namespace}
kubectl create secret docker-registry protoobp-registry \
  --docker-username=${login} --docker-password=${password} \
  --docker-server=${registry} --namespace=${namespace}
```

3. Подготовьте файл `values.yaml`:

```
  targetSystem: "linux"
  commonLabels: {}
  registry: registry.git.proto.group/protoobp/protoobp-artifacts
  protoobp:
    apiKey: <ВАШ_API_KEY>
    clusterName: <ИМЯ_КЛАСТЕРА>
    pobp_url: http://<адрес_сервера_protoobp>
    pobp_apm_url: http://<адрес_сервера_protoobp>
    logLevel: INFO
    processAgent:
      enabled: false
      processCollection: false
    kubelet:
      tlsVerify: false
  clusterAgent:
    image:
      name: cluster-agent
      tag: 7.40.3
      pullSecrets:
        - name: "protoobp-registry"
  agents:
    image:
      name: protoobp-agent
      tag: 7.40.3
      pullSecrets:
        - name: "protoobp-registry"
```

4. Установите чарт средствами `helm` в подготовленный namespace.

> Значение `clusterName` задаётся вручную и должно соответствовать фактическому имени кластера — оно используется для разделения данных разных кластеров в интерфейсе.
>
> При использовании локального реестра образов измените параметр `registry`, а для автоматической инструментации дополнительно задайте переменную окружения cluster-агента `POBP_ADMISSION_CONTROLLER_AUTO_INSTRUMENTATION_CONTAINER_REGISTRY` с адресом зеркала.

5.5.2. Особые режимы развёртывания

Поддерживаются: использование `hostPath` ноды, работа в непривилегированном режиме, дополнительная конфигурация для OpenShift. Порядок изменения параметров и удаления агента описан в документации вендора.

5.5.3. Автоматическая инструментация приложений

Cluster-агент реализует webhook автоматического подключения трейсеров к подам приложений. Для его работы требуется сетевой доступ от control plane Kubernetes к нодам кластера по порту 8000/tcp.

5.6. Работа агента через прокси-сервер

Агент поддерживает передачу данных через HTTP-прокси. Параметры прокси задаются в файле конфигурации агента.

5.7. Расположение файлов конфигурации и журналов агента

Назначение	Linux	Windows
Основной файл конфигурации	`/etc/protoobp-agent/protoobp.yaml`	`%ProgramData%\Protoobp\protoobp.yaml`
Конфигурации модулей технологий	`/etc/protoobp-agent/conf.d/<технология>.d/conf.yaml`	`%ProgramData%\Protoobp\conf.d\<технология>.d\conf.yaml`

Назначение	Linux	Windows
Журналы работы	`/var/log/protoobp/`	`C:\ProgramData\Protoobp\logs`
Журнал установки	`$(pwd)/protobop-agent-install.log`	`%TEMP%\MSI*.LOG`

Пример: конфигурация модуля мониторинга MySQL на Linux — `/etc/protoobp-agent/conf.d/mysql.d/conf.yaml`.

5.8. Проверка состояния агента

Платформа	Команда
Linux	`sudo protoobp-agent status`
Windows	`"C:\Program Files\Protoobp\Protoobp Agent\bin\agent.exe" status` (из привилегированной командной строки)
Docker	`sudo docker exec -it protoobp-agent agent status`
Kubernetes	`kubectl exec -it <AGENT_POD_NAME> -- agent status`

5.9. Подключение источников данных

Источник	Порядок подключения
Трейсинг приложений	Подключение трейсера соответствующего языка (Java, PHP, Node.js, Python, .NET, Go); альтернативно — OpenTelemetry, в том числе zero-code инструментация eBPF
Инфраструктурные компоненты	Активация соответствующего модуля агента в каталоге `conf.d` (свыше 300 поддерживаемых технологий)
Пользовательские метрики Prometheus	Настройка сбора метрик с эндпоинтов `/metrics`
Логи	Установка агента, либо передача логов от внешних источников по REST/HTTP (форматы OTLP, Elasticsearch, JSON, Loki, DataDog)
Мониторинг цифрового опыта (RUM)	Подключение браузерного агента к веб-приложению, SDK для Android и iOS
Сетевое оборудование и сетевые проверки	Настройка соответствующих модулей агента (HTTP, DNS, TCP, TLS, SSH, NTP, SNMP)

Конкретные параметры подключения по каждой технологии приведены в документации вендора.

6. Настройка Системы

6.1. Управление лицензией

6.1.1. Изменение лицензионного ключа

Выполняется при переходе с ознакомительной лицензии на промышленную либо при продлении.

1. В файле `/opt/protoobp/.env` обновите:
   - `POBP_LICENSE_KEY` — значение из поля «лицензионный ключ» нового сертификата;
   - `POBP_LICENSE_COMPANY_NAME` — значение из поля «имя компании для лицензии» (при изменении).
2. Примените изменения:

```
docker compose -f docker-compose-202.yaml up -d
```

3. Проверьте журнал компонента `proto-trace-processor` на отсутствие сообщений об ошибке активации лицензии:

```
docker compose -f docker-compose-202.yaml logs proto-trace-processor
```

Альтернативно — раздел интерфейса `Диагностика` → `Логи бэкенда` → `proto-trace-processor`.

6.1.2. Изменение токена доступа к артефактам

1. Обновите учётные данные Docker-реестра:

```
docker logout registry.git.proto.group
docker login --username <USERNAME> registry.git.proto.group
```

2. При использовании собственного proxy-репозитория (Nexus, Artifactory) обновите в нём учётные данные подключения к `registry.git.proto.group`.
3. При загрузке артефактов напрямую из репозитория вендора в конвейерах CI/CD обновите значение токена в этих конвейерах.

6.2. Настройка сроков хранения данных (retention)

Сроки хранения задаются переменными в файле `.env`.

Переменная	Значение по умолчанию	Область данных
`POBP_DATA_RETENTION_METRICS`	`2w`	Инфраструктурные метрики и метрики сервисов в формате Prometheus
`POBP_DATA_RETENTION_DB_RAW_RUM_DAYS`	`7`	Данные цифрового опыта без агрегации
`POBP_DATA_RETENTION_DB_RAW_TRACES_DAYS`	`1`	Трейсы без агрегации, включая трейсы сессий
`POBP_DATA_RETENTION_DB_RUM_DAYS`	`7`	Данные цифрового опыта, минутная агрегация
`POBP_DATA_RETENTION_DB_TRACES_DAYS`	`7`	Трейсы, базовая агрегация
`POBP_DATA_RETENTION_DB_TRACES_AGG_DAYS`	`30`	Агрегированные трейсы
`POBP_DATA_RETENTION_DB_TRACES_AGG_MINUTE_DAYS`	`7`	Метрики трейсов, минутная агрегация

Переменная	Значение по умолчанию	Область данных
`POBP_DATA_RETENTION_DB_TRACES_AGG_HOUR_DAYS`	`30`	Метрики трейсов, часовая агрегация
`POBP_DATA_RETENTION_DB_TRACES_AGG_DAY_DAYS`	`60`	Метрики трейсов, суточная агрегация
`POBP_DATA_RETENTION_DB_DBMMETRICS_DAYS`	`7`	Метрики баз данных
`POBP_DATA_RETENTION_DB_PROCESS_DAYS`	`15`	Метрики бизнес-процессов
`POBP_DATA_RETENTION_EVENTS_DAYS`	`30`	События, собираемые агентами
`POBP_DATA_RETENTION_ALERTS_DAYS`	`90`	Сработавшие алерты
`POBP_CORE_RECORD_DATA_TTL`	`2`	Срок хранения записей в OpenSearch, дней
`POBP_CORE_METRICS_DATA_TTL`	`2`	Срок хранения метрик в OpenSearch, дней

Применение изменений:

```
docker compose -f docker-compose-202.yaml up -d
```

> **Планирование ёмкости.** Увеличение сроков хранения прямо пропорционально увеличивает потребление дискового пространства. Перед изменением значений оцените текущий темп прироста данных в каталоге `POBP_DATA_DIRECTORY` и предусмотрите запас не менее 20 % свободного места. При остатке свободного места менее 10 % запись трейсов прекращается.
>
> **Требование соответствия.** Сроки хранения должны быть согласованы с внутренними требованиями организации к срокам хранения технологических журналов и, при обработке персональных данных, — с требованиями законодательства.

6.3. Настройка порогов APDEX

APDEX — интегральный индекс удовлетворённости пользователей скоростью работы системы, принимает значения от 0 до 1.

Транзакции классифицируются как:

- **Satisfied** — завершённые без ошибок за время меньше `APDEX T`;
- **Tolerating** — завершённые без ошибок за время меньше `4 × APDEX T`;
- **Frustrated** — завершённые с ошибкой либо за время больше `4 × APDEX T`.

Начиная с версии 199 пороги задаются через веб-интерфейс: **APM → Настройки → APDEX**.

Доступные действия: создание порога для сервиса, редактирование существующего порога.

Поля формы:

- **Имя сервиса** — выбирается из списка или вводится вручную;
- **APDEX T (мс)** — порог времени отклика; значение по умолчанию при создании — 500 мс.

> **Рекомендация.** Значение `APDEX T` должно отражать согласованное с владельцем сервиса ожидание пользователей, а не текущее фактическое время отклика. Установка порога «под факт» лишает показатель диагностической ценности.

6.4. Настройка SMTP

Требуется для отправки приглашений новым пользователям. На этапе пилотного тестирования шаг является опциональным — учётные записи можно создать вручную с назначением пароля.

В файле `.env` задайте:

```
SMTP_HOST=smtp.my_company.net
SMTP_PORT=587
SMTP_AUTH=true
SMTP_USER=user
SMTP_PASSWORD=password
SMTP_FROM=protoobp@proto.group
SMTP_SSL=false
SMTP_STARTTLS=false
```

Параметр	Описание
`SMTP_HOST`	Адрес SMTP-сервера
`SMTP_PORT`	Порт SMTP-сервера
`SMTP_AUTH`	Использовать логин и пароль для авторизации (`true` / `false`)
`SMTP_USER`	Имя пользователя SMTP
`SMTP_PASSWORD`	Пароль пользователя SMTP
`SMTP_FROM`	Адрес отправителя уведомлений
`SMTP_SSL`	Использовать SSL (`true` / `false`)
`SMTP_STARTTLS`	Использовать STARTTLS (`true` / `false`)

> **Три независимых набора параметров почты.** Не путайте их при настройке:
>
> — `SMTP_*` в файле `.env` — отправка приглашений новым пользователям (настоящий пункт); — параметры почтового сервера в **канале оповещения** типа «Электронная почта» — задаются в веб-интерфейсе для каждого канала отдельно (см. «Руководство пользователя», п. 4.8.5); — `POBP_REPORT_RUNNER_SMTP_*` в файле `.env` — рассылка отчётов сервисом `proto-report-runner` (начиная с версии 202, см. п. 4.3, шаг 2).
>
> Настройка одного набора не включает остальные.

6.5. Настройка единого входа (SSO)

Поддерживаемые провайдеры:

- **LDAP-каталоги**: Active Directory, OpenLDAP, Red Hat Directory Server / FreeIPA, Novell eDirectory, IBM Tivoli и другие;
- **OIDC**: Keycloak, IdentityServer4.

Поддержка иных OIDC- или SAML 2.0-совместимых систем добавляется по запросу.

> **Обязательное условие.** Перед изменением типа аутентификации (например, со встроенной на LDAP) выполните:
>
> ```
> docker compose -f docker-compose-<version>.yaml stop proto-auth proto-postgresql
> proto-nginx
> docker compose -f docker-compose-<version>.yaml rm proto-auth proto-postgresql proto-
> nginx
> docker volume rm protoobp_postgres_data
> ```
>
> Данная последовательность удаляет том с данными сервиса аутентификации. Локальные учётные записи будут утрачены — предварительно зафиксируйте перечень пользователей и их роли.

6.5.1. Подготовка провайдера

LDAP. Создайте в каталоге группы **строго с указанными именами**:

- `proto_obp_admin` — пользователи с правами «Администратор»;
- `proto_obp_user` — пользователи с правами «Пользователь».

Расположение групп в каталоге произвольное. Добавьте пользователей в соответствующие группы.

OIDC через Keycloak. Создайте группы, членство в которых определяет отнесение пользователя к администраторам или пользователям, и добавьте в них пользователей.

OIDC через IdentityServer4. Создайте роли и назначьте их пользователям.

6.5.2. Настройка типа аутентификации

В файле `.env` задайте:

Провайдер	Значение
LDAP	`AUTH_TYPE=LDAP`
OIDC через Keycloak	`AUTH_TYPE=OIDC`
OIDC через IdentityServer4	`AUTH_TYPE=OIDC-IS4`

6.5.3. Файл настроек провайдера

LDAP — файл `.env_ldap`:

```
LDAP_VENDOR="ad"
LDAP_CONNECTION_URL="ldap://ldap.my_company.net"
LDAP_START_TLS="false"
LDAP_AUTH_TYPE="simple"
LDAP_BIND_DN="uid=admin,cn=users,cn=accounts,dc=my,dc=company"
LDAP_BIND_PASSWORD="secret"
LDAP_USERS_DN="cn=users,cn=accounts,dc=my,dc=company"
LDAP_SEARCH_SCOPE="2"
LDAP_USER_SEARCH_FILTER=(|
(memberOf=cn=proto_obp_user,cn=groups,cn=accounts,dc=my,dc=company)
(memberOf=cn=proto_obp_admin,cn=groups,cn=accounts,dc=my,dc=company))
LDAP_UUID_ATTRIBUTE="ipauniqueid"
LDAP_USERNAME_ATTRIBUTE="sAMAccountName"
LDAP_USER_OBJECT="person,organizationalPerson,user"
LDAP_RDN_ATTRIBUTE="cn"
LDAP_GROUPS_DN="cn=groups,cn=accounts,dc=my,dc=company"
LDAP_GROUPS_FILTER="(&(objectclass=group)(|(cn=proto_obp_user)(cn=proto_obp_admin)))"
LDAP_GROUP_OBJECT="group"
LDAP_MEMBER_ATTRIBUTE="member"
LDAP_GROUPS_STRATEGY="LOAD_GROUPS_BY_MEMBER_ATTRIBUTE"
```

Типовые значения атрибутов по типам каталогов:

Параметр	Active Directory	OpenLDAP / FreeIPA
`LDAP_UUID_ATTRIBUTE`	`objectGUID`	`entryUUID` / `ipauniqueid`
`LDAP_USERNAME_ATTRIBUTE`	`sAMAccountName`	`uid`
`LDAP_USER_OBJECT`	`person,organizationalPerson,user`	`inetOrgPerson,organizationalPerson`
`LDAP_RDN_ATTRIBUTE`	`cn`	`uid`
`LDAP_GROUP_OBJECT`	`group`	`groupOfNames` / `groupOfUniqueNames`
`LDAP_MEMBER_ATTRIBUTE`	`member`	`member` / `memberUid` (для

Параметр	Active Directory	OpenLDAP / FreeIPA
		`posixGroup`)

Стратегии получения групп:

- `LOAD_GROUPS_BY_MEMBER_ATTRIBUTE` — через атрибут группы; универсальный вариант;
- `GET_GROUPS_FROM_USER_MEMBEROF_ATTRIBUTE` — через атрибут пользователя `memberOf`; быстрее, применим для AD и серверов с поддержкой overlay `memberOf`.

> **Взаимоисключающие настройки.** `ldaps://` и `LDAP_START_TLS="true"` одновременно не применяются. `ldaps://` открывает TLS-туннель сразу на порту 636; `LDAP_START_TLS="true"` инициирует STARTTLS поверх соединения без шифрования (обычно порт 389). Одновременное указание приводит к попытке выполнить STARTTLS внутри уже установленного туннеля и к ошибке подключения.
>
> **Требование безопасности.** Для промышленной эксплуатации используйте защищённое соединение с каталогом (`ldaps://` либо STARTTLS). Соединение без шифрования допустимо только в изолированном сегменте на этапе отладки.

OIDC через Keycloak — файл `.env_oidc`:

```
OIDC_NAME=my_oidc
KEYCLOAK_URL=http://oidc_url:86
KEYCLOAK_REALM=my_realm
# KEYCLOAK_AUTH_PATH=/auth
KEYCLOAK_CLIENT_ID=my_client
KEYCLOAK_CLIENT_SECRET=client_secret
KEYCLOAK_GROUP_USER_KEY=userGroups
KEYCLOAK_GROUP_USER_VALUE=user
KEYCLOAK_GROUP_ADMIN_KEY=userGroups
KEYCLOAK_GROUP_ADMIN_VALUE=admin
```

OIDC через IdentityServer4 — файл `.env_oidc_is4`:

```
OIDC_NAME=is4-dev
IS4_BASE_URL=https://identity-server-is4.test.local
IS4_CLIENT_ID=proto_obp
IS4_CLIENT_SECRET=some-strong-secret
IS4_GROUP_USER_KEY=role
IS4_GROUP_USER_VALUE=proto_obp_user
IS4_GROUP_ADMIN_KEY=role
IS4_GROUP_ADMIN_VALUE=proto_obp_admin
```

6.5.4. Подключение файла настроек к сервису аутентификации

В `docker-compose-<version>.yaml` у сервиса `proto-auth` добавьте том с файлом настроек:

```
proto-auth:
  volumes:
    - ./.env_ldap:/usr/share/java/keycloak/.env_ldap        # для LDAP
    # - ./.env_oidc:/usr/share/java/keycloak/.env_oidc      # для Keycloak
    # - ./.env_oidc_is4:/usr/share/java/keycloak/.env_oidc_is4  # для IdentityServer4
```

Примените изменения:

```
docker compose -f docker-compose-<version>.yaml up -d
```

6.5.5. Доверенные TLS-сертификаты внешних систем

Настройка требуется во всех случаях, когда `proto-auth` обращается к внешним системам по TLS: OIDC (`https://...`), LDAPS или STARTTLS, SMTP с TLS/SSL.

Признак проблемы. В журнале `proto-auth` появляются ошибки `SSLHandshakeException`, `PKIX path building failed`, `unable to find valid certification path to requested target` — контейнер не доверяет сертификату удалённого сервера или его цепочке.

Порядок устранения:

1. Создайте каталог `truststores` рядом с `docker-compose-<version>.yaml`.
2. Поместите в него сертификаты удостоверяющих центров (корневые и промежуточные) в формате `.pem` (либо `.p12` / `.pfx` / `.pkcs12`).
3. Добавьте у сервиса `proto-auth`:

```
proto-auth:
  environment:
    KC_TRUSTSTORE_PATHS: /opt/keycloak/conf/truststores
    KC_TLS_HOSTNAME_VERIFIER: DEFAULT
  volumes:
    - ./truststores:/opt/keycloak/conf/truststores:ro
```

Рекомендации:

- Добавляйте сертификаты удостоверяющих центров (корневой и промежуточные), а не только конечный сертификат сервера.
- Для промышленной эксплуатации используйте `KC_TLS_HOSTNAME_VERIFIER=DEFAULT`. Значение `ANY` допустимо только для кратковременной диагностики.
- Для защищённых LDAP-соединений проверка имени хоста должна корректно отрабатывать на стороне сертификата LDAP-сервера.

6.6. Настройка модуля инцидентов

Названия и порядок этапов жизненного цикла инцидента настраиваются в подразделе `Инциденты и Алерты` → `Настройки Инцидентов`. По умолчанию применяется последовательность в соответствии с практиками ITIL: Обнаружен → Диагностика → Устранение → Закрыт.

Условия доступности ИИ-функций и связанные требования безопасности приведены в п. 4.3 (шаг 2) и п. 9.5.

6.7. Окна обслуживания

Начиная с версии 202 в подразделе `Инциденты и Алерты` → `Окна обслуживания` объявляются периоды регламентных работ. На время окна срабатывания по заданным объектам не поднимают инциденты и не рассылаются по каналам оповещения, но остаются в истории алертов с пометкой «Обслуживание». Правила обнаружения пропадания данных (No-Data) в окне также не срабатывают.

Порядок работы с окнами описан в «Руководстве пользователя», п. 4.9А. Задачи администратора:

- выдать право на ресурс `maintenance_window` подразделениям, выполняющим регламентные работы (см. п. 7.3.4);
- закрепить в регламенте эксплуатации требование объявлять окно **до** начала работ, а не после первых оповещений;
- при планировании работ на самой платформе (обновление, резервное копирование с остановом) объявлять окно на затронутые объекты — см. п. 8.3 и п. 8.4;
- контролировать по журналу аудита создание и изменение окон: слишком широкая область действия может скрыть реальные отказы.

6.8. Отчёты по электронной почте

Начиная с версии 202 доступна регулярная рассылка дашбордов в виде PDF-документов (вкладка `Дашборды и отчёты` → `Отчёты`). Задачи администратора:

- задать параметры почтового сервера отчётности в файле `.env` (см. п. 4.3, шаг 2);
- перевыпустить пароль служебной учётной записи отчётности (`POBP_REPORT_RUNNER_KEYCLOAK_CLIENT_SECRET`);
- выдать право `admin` на ресурс `reports` пользователям, ответственным за отчётность;
- контролировать историю запусков: систематические ошибки отправки указывают на проблемы с почтовым сервером, ошибки отрисовки — на недоступность источников данных дашборда.

Порядок настройки отчётов пользователем описан в «Руководстве пользователя», п. 4.13.

---

7. Администрирование пользователей и доступа

7.1. Модель доступа

Система реализует ролевую модель управления доступом (RBAC). Начиная с версии 200 ролевая модель включена по умолчанию и распространяется в том числе на программные интерфейсы.

Базовая модель прав (раздел «Настройки»):

Действие	Администратор	Пользователь
Добавление пользователей	+	–
Удаление пользователей	+	–
Изменение роли пользователя	+	–
Изменение (сброс) пароля пользователя	+	–
Просмотр и чтение основных данных	+	+
Создание кастомных дашбордов	+	–
Редактирование кастомных дашбордов	+	–
Удаление кастомных дашбордов	+	–

Расширенная модель доступна в разделе **«Администрирование RBAC»** (с версии 200) и включает пользовательские роли, матрицу прав, ресурсно-ограниченные права, привязку внешних групп и журнал аудита.

7.2. Базовое управление пользователями (раздел «Настройки»)

Добавление пользователя:

1. Войдите в интерфейс под учётной записью администратора.
2. Перейдите в раздел «Настройки».
3. Нажмите **«+ Пользователь»**.
4. Укажите данные пользователя и нажмите **«Добавить»**. На указанный адрес электронной почты будет направлено приглашение (требуется настроенный SMTP, п. 6.4).

Изменение роли: раздел «Настройки» → столбец «Операции» → значок «карандаш» в строке пользователя.

Сброс пароля: раздел «Настройки» → столбец «Операции» → значок «…» → задайте временный пароль. Пароль подлежит смене при первом входе пользователя.

7.3. Раздел «Администрирование RBAC»

Раздел доступен только пользователям с правами администратора и содержит пять вкладок.

7.3.1. Вкладка «Пользователи»

Таблица зарегистрированных пользователей со столбцами: Имя, Email, Роли, Последний вход.

Возможности: поиск по имени или email; фильтр по роли; приглашение нового пользователя.

Операции над пользователем: управление ролями, редактирование данных, сброс пароля, включение/отключение учётной записи.

7.3.2. Вкладка «Роли»

Роли делятся на два типа:

- **системные** — предустановленные («Администратор», «Пользователь»); отмечены индикатором, не подлежат удалению, их разрешения не изменяются;
- **пользовательские** — создаются администратором под конкретные сценарии доступа.

Операции: создание роли (имя и описание), редактирование, просмотр деталей.

7.3.3. Вкладка «Права доступа»

Матрица разрешений: строки — типы ресурсов платформы (дашборды, алерты, сервисы, настройки и др.), столбцы — роли, ячейки — разрешённые действия (просмотр, создание, редактирование, удаление, администрирование).

Разрешения системных ролей не изменяются — для специфических прав создайте пользовательскую роль.

7.3.4. Ресурсно-ограниченные права

Права выдаются не только на уровне типа ресурса, но и на конкретные экземпляры.

Тип ресурса	Пример ограничения	Доступно с версии
`service`	Доступ только к сервису `payment_bank`	200
`host`	Доступ только к хосту `db-01.internal`	201
`service_group`	Доступ только к группе сервисов `credit_conveyor`	201
`bt` (бизнес-транзакция)	Просмотр только выделенных ключевых бизнес-транзакций	200
`slo`	Редактирование только SLO определённой команды	200
`dashboard`	Редактирование только выделенных дашбордов	200
`maintenance_window`	Управление окнами обслуживания (создание, изменение, отмена, удаление)	202
`reports`	Доступ к вкладке «Отчёты» и управление рассылками; требуется право `admin`	202

Порядок настройки:

1. Откройте редактирование роли на вкладке «Роли».
2. В секции прав выберите тип ресурса и действия.
3. Вместо подстановочного значения `*` укажите идентификаторы или имена конкретных ресурсов.
4. Сохраните изменения.

Если для типа ресурса не задано ни одного ограничения, применяется правило «нет доступа». Пользователям с ролью администратора (`{admin, *, *}`) доступны все ресурсы.

7.3.5. Ограничение области видимости данных

Права на типы `service`, `host` и `service_group` автоматически применяются ко **всем запросам данных**: виджетам дашбордов, Traces Explorer, Logs Explorer, браузеру метрик, разделам «Инфраструктура → Хосты», «Бизнес-приложения», «SLO».

Три измерения области видимости:

Измерение	Тип ресурса	Метка в данных
Сервисы	`service`	`service`
Хосты	`host`	`host` (метрики, трейсы), `hostname` (логи)
Группы сервисов	`service_group`	`service_group`

Для каждого измерения возможны три состояния: доступ ко всем (ключ `*`), список конкретных значений, не настроено (измерение не участвует в фильтрации).

Правило комбинирования — объединение (ИЛИ). Пользователь видит данные, попадающие хотя бы под одно настроенное измерение.

> **Критично для корректной настройки.**
>
> — Если **любое** из трёх измерений выставлено в «Доступ ко всем» (*), фильтр по данным снимается **целиком**, а не только по этому измерению. Роль с `host = *` и списком сервисов покажет данные всех сервисов и всех хостов. — Комбинация «список сервисов + список хостов» из-за объединения по ИЛИ **расширяет**, а не сужает выборку. Чтобы ограничить пользователя одним хостом, оставьте в роли только ограничение по `host`. — Права **аддитивны**: эффективный доступ — объединение всех ролей пользователя, назначенных напрямую и через внешние группы. Ограничивающая роль не может «вычесть» доступ, выданный другой ролью. Если пользователю дополнительно назначена системная роль «Пользователь» с правилом `{view, *, *}`, ограничение снимается. — При расследовании ситуации «пользователь с ограниченной ролью видит все данные» проверьте членство во внешних группах: роль может назначаться автоматически. Членство задаётся в провайдере учётных записей, а не в разделе «Администрирование RBAC».

Разворачивание групп сервисов. Грант на конкретную группу сервисов при загрузке прав разворачивается в набор её сервисов-участников по данным за последние 30 дней. Открытый грант (`service_group = *`) не разворачивается.

Применение фильтра к источникам данных:

Источник	Механизм
PromQL (метрики)	На каждое ограниченное измерение добавляется отдельный `extra_filters[]`: `{service=~"…"}`, `{host=~"…"}`, `{service_group=~"…"}`; значения объединяются по ИЛИ
LogsQL (логи)	Добавляется `extra_filters` вида `(service:in("…") or hostname:in("…"))`. Отдельного поля `service_group` в логах нет — доступ к группе доходит через разворачивание в сервисы
CubeJS (OLAP)	В JWT-токен добавляются claim'ы `allowed_services`, `allowed_hosts`, `allowed_service_groups`, преобразуемые в условия фильтрации

При полном отсутствии прав на ресурс пользователю отображается локализованная панель «Доступ запрещён».

7.3.6. Вкладка «Внешние группы»

Обеспечивает автоматическое назначение ролей платформы по группам внешнего провайдера (LDAP, OIDC).

При входе через SSO платформа определяет группы пользователя во внешнем провайдере; при наличии настроенного сопоставления пользователю автоматически назначаются соответствующие роли.

Источники сопоставления (`source_type`):

Источник	Когда применяется
LDAP	Пользователи, аутентифицируемые через LDAP-каталог
OIDC	Пользователи, входящие через внешнего OIDC-провайдера
Встроенный (built-in)	Предустановленные сопоставления для исторических имён групп; не удаляются, могут быть отключены

По умолчанию платформа ожидает список групп в claim `groups`. Если провайдер использует другое имя (`roles`, `ldap_groups`, `cognito:groups`), оно указывается в настройках Keycloak.

При обновлении с предыдущих версий исторические имена LDAP-групп сохраняются как встроенные сопоставления, что исключает потерю доступа существующими пользователями после миграции.

7.3.7. Вкладка «Журнал аудита»

Журнал содержит хронологический перечень действий пользователей в платформе и отвечает на вопросы службы информационной безопасности и аудиторской проверки: кто, когда, с какого адреса, над каким объектом выполнил действие и чем оно завершилось.

> **Изменение в версии 202.** Журнал переработан: добавлены раздельные потоки событий, состояние объекта «до» и «после», расширенные фильтры и экспорт. До версии 202 журнал охватывал только операции управления доступом.

Потоки событий

Поток	Что попадает
Изменения	Создание, изменение и удаление объектов: роли и права, правила алертинга, окна обслуживания, отчёты, дашборды, ресурсы PCM и другие
Доступ	Обращения на чтение данных: просмотр, выгрузка, запуск операций
Все	Оба потока вместе

Разделение введено потому, что у потоков разная плотность и разный смысл: изменений немного и каждое значимо, обращений на чтение — на порядки больше.

Фильтры

Фильтр	Назначение
Период времени	Интервал, за который отображаются события
Действие	Создал, изменил, удалил, просмотрел, выгрузил, обновил, запустил, отменил, запустил расследование
Тип ресурса	Класс объекта: сервисы, хосты, дашборды, инциденты, SLO, окна обслуживания, отчёты, управление доступом и другие
Результат	«Разрешено», «Отказано», «Ошибка»
Объект	Поиск по объекту: точное совпадение по идентификатору либо начало имени
IP-адрес	Адрес, с которого выполнено действие

Отдельно доступны быстрые срезы **«Только отказы»** и **«Изменения прав доступа»** — два наиболее частых запроса службы безопасности. Из строки события можно развернуть подробности и перейти к срезам **«Показать всю активность этого пользователя»** и **«Показать всю активность с этого адреса»** — типовой путь разбора инцидента безопасности.

Состояние «до» и «после»

Для изменений прав доступа и окон обслуживания в подробностях события отображается таблица изменённых полей: **Поле**, **Было**, **Стало**. Сравнивать две версии объекта вручную не требуется.

Дополнительно в подробностях доступны технические данные: идентификатор запроса, User-Agent, HTTP-метод, код ответа, параметры запроса (только имена — значения не сохраняются) и исходный JSON события с копированием в буфер обмена.

> **Ограничения размера.** Тело запроса сохраняется, если оно в формате JSON и не превышает 64 КБ; иначе событие содержит соответствующую пометку. Крупные подробности могут быть сохранены частично — такие записи помечаются явно, усечение без уведомления не выполняется.

Экспорт

Журнал выгружается в форматах **CSV** и **JSON**. Выгружается **вся выборка по текущим фильтрам**, а не только текущая страница. Если выборка не поместилась в файл целиком, выводится предупреждение с числом событий — период следует сузить и повторить выгрузку.

Для непрерывной передачи событий во внешние системы используется экспорт в SIEM (порядок настройки приведён в документации вендора).

Сроки хранения

Поток	Срок хранения
Изменения	365 дней
Доступ	90 дней

Обновление платформы до версии 202 выполняется без перезаписи данных: записи, накопленные в предыдущих версиях, сохраняются и хранятся 365 дней, но не содержат новых полей (состояние «до/после», категория события).

Права доступа и регламент

Просмотр журнала требует права администратора. Если схема журнала на стенде ещё не обновлена, интерфейс сообщает об этом отдельно — часть возможностей в таком режиме недоступна.

> **Требование эксплуатации.** Журнал аудита подлежит регулярному контролю (рекомендуемая периодичность — не реже одного раза в неделю с обязательным просмотром среза «Только отказы») и передаче в систему централизованного сбора событий безопасности организации (SIEM). Журнал входит в состав данных, подлежащих резервному копированию (таблица `rbac_audit_log`, см. п. 8.4).

7.3.8. Вкладка «Проверка на ПДн»

Доступна начиная с версии 203. Выполняет **выборочную проверку телеметрии на наличие персональных данных**.

Инструмент отвечает на практический вопрос эксплуатации: не попадают ли в трейсы, логи и параметры URI сведения, которые организация обязана защищать. До версии 203 такая проверка выполнялась вручную — выборочным просмотром трейсов и логов.

Порядок применения:

1. Выполните проверку до ввода в промышленную эксплуатацию — после подключения каждого нового приложения или источника данных.

2. По результатам проверки настройте дополнительные правила обфускации и игнорирования ресурсов (п. 9.3).
3. Повторите проверку и убедитесь, что выявленные категории данных больше не собираются.
4. Результаты проверки приобщите к материалам, подтверждающим выполнение требований по обработке персональных данных (п. 9.7).

> **Ограничение метода.** Проверка является **выборочной**: отсутствие находок не является доказательством того, что персональные данные не собираются. Она не заменяет анализ состава собираемых данных совместно с подразделением информационной безопасности и владельцами приложений, а дополняет его.

7.4. Учётные записи для интеграций

Для каждой внешней интеграции (скрипт, BI-инструмент, внешний дашборд) создаётся отдельная учётная запись платформы с минимально необходимым набором прав — правом просмотра на требуемые сервисы.

Порядок получения токена и вызова API приведён в «Руководстве пользователя», п. 4.14.2.

> **Требования.** Не используйте персональные учётные записи сотрудников в интеграциях. Реквизиты интеграционных учётных записей размещайте в системе управления секретами организации. Время жизни access-токена — около 5 минут; длительно работающие интеграции обновляют токен по `refresh_token`.

7.5. Регламент управления доступом

Рекомендуемый порядок:

1. **Предоставление доступа** — по заявке руководителя подразделения с указанием требуемой роли и перечня сервисов/хостов.
2. **Принцип минимальных привилегий** — по умолчанию выдаётся роль с правом просмотра и ограниченной областью видимости.
3. **Пересмотр прав** — не реже одного раза в полугодие; проверка соответствия назначенных ролей фактическим обязанностям.
4. **Отзыв доступа** — в день прекращения трудовых отношений или перевода сотрудника; при использовании SSO — исключение из соответствующих групп каталога.
5. **Контроль административных учётных записей** — количество учётных записей с ролью «Администратор» минимизируется; их действия контролируются по журналу аудита.

8. Эксплуатация и регламентное обслуживание

8.1. Управление компонентами серверной части

Все операции выполняются из каталога `/opt/protoobp`.

Операция	Команда
Запуск всех компонентов	`docker compose -f docker-compose-202.yaml up -d`
Остановка всех компонентов	`docker compose -f docker-compose-202.yaml stop`
Перезапуск отдельного компонента	`docker compose -f docker-compose-202.yaml restart <имя_сервиса>`
Состояние компонентов	`docker ps -a --format "table {{.Names}}\t{{.Status}}" \
Журнал компонента	`docker compose -f docker-compose-202.yaml logs <имя_сервиса>`
Применение изменений конфигурации	`docker compose -f docker-compose-202.yaml up -d`

Просмотр журналов компонентов также доступен через веб-интерфейс: раздел `Диагностика` → `Логи бэкенда`.

8.1.1. Раздел «Диагностика»

Раздел содержит сведения о работе самой платформы.

Пункт	Содержание	Доступ
Лицензия	Параметры и срок действия лицензии	—
Объём данных	Сколько телеметрии принимает и хранит платформа, в разрезе типов данных и источников (начиная с версии 203)	Только администраторам
Processor	Метрики работы компонентов обработки телеметрии	—
Реестр конфигурации	Просмотр параметров конфигурации сервисов платформы с описаниями и фильтрами (начиная с версии 203). Только чтение	Только администраторам
Логи бэкенда	Журналы работы компонентов платформы	—

Как это применять в эксплуатации:

- **«Объём данных»** — основной инструмент планирования ёмкости. Он показывает фактическое распределение принимаемой и хранимой телеметрии по типам и источникам, поэтому позволяет принимать решения об изменении сроков хранения (п. 6.2) и о расширении дисковой подсистемы на основании данных, а не оценок. Регулярный просмотр также выявляет источники с аномальным ростом объёма — как правило, это следствие избыточного логирования или высокой кардинальности метрик.
- **«Реестр конфигурации»** — просмотр фактических значений параметров сервисов платформы. Применяется при разборе нештатных ситуаций (проверить, какое значение параметра фактически применено) и после обновления версии (сверить состав и значения параметров с примечаниями к выпуску). Изменение параметров через реестр не выполняется — конфигурация задаётся в файле `.env` и в файле композиции.

8.2. Регламент контроля состояния

Периодичность	Контролируемый параметр	Критерий нормы	Действие при отклонении
Ежедневно	Состояние контейнеров	Все `Up`; контейнеры с	Изучить журнал

Периодичность	Контролируемый параметр	Критерий нормы	Действие при отклонении
		проверками — `healthy`; отсутствие циклических перезапусков	компонента, п. 10
Ежедневно	Свободное место в `POBP_DATA_DIRECTORY`	Не менее 20 %	Расширить том либо сократить сроки хранения (п. 6.2). **При остатке менее 10 % запись трейсов прекращается**
Ежедневно	Метрика `consumer lag` с тегом `protoobp-consumer` (`Инфраструктура` → `Kafka` → хост серверной части)	Значение не возрастает монотонно	Постоянный рост означает нехватку ресурсов сервера — обратиться в техническую поддержку
Ежедневно	Актуальность данных на дашбордах сервисов	Данные не старше нескольких минут	Проверка по цепочке приложение → трейсер → агент → серверная часть (п. 10.1)
Еженедельно	Журнал аудита: поток «Изменения» и срез «Только отказы»	Отсутствие неавторизованных изменений и аномальных отказов в доступе	Уведомить подразделение информационной безопасности
Еженедельно	История запусков отчётов	Отсутствие систематических ошибок отправки и отрисовки	Проверить параметры почтового сервера отчётности и доступность источников данных дашбордов
Еженедельно	Перечень окон обслуживания	Отсутствие бессрочных регулярных окон со слишком широкой областью действия	Скорректировать область действия либо задать дату окончания серии
Еженедельно	Перечень контролируемых узлов	Соответствие фактическому составу инфраструктуры	Установить агент на неохваченные узлы
Ежемесячно	Загрузка CPU и памяти сервера серверной части	Наличие запаса не менее 20 %	Планирование расширения ресурсов
Ежемесячно	`Диагностика` → `Объём данных`	Отсутствие источников с аномальным ростом объёма телеметрии	Разобрать источник: избыточное логирование либо высокая кардинальность метрик; при необходимости скорректировать сроки хранения (п. 6.2)
Ежеквартально	`Роли и доступ` → `Проверка на ПДн`	Отсутствие персональных данных в выборке телеметрии	Настроить дополнительные правила обфускации и игнорирования ресурсов (п. 9.3), повторить проверку
Ежеквартально	Перечень правил алертинга	Отсутствие правил, срабатывания которых систематически не приводят к действиям	Отключить или скорректировать правила
Ежеквартально	Права доступа	Соответствие фактическим обязанностям	Пересмотр ролей (п. 7.5)
По событию	Срок действия TLS-сертификата	Не менее 30 суток до истечения	Плановая замена (п. 4.6)
По событию	Срок действия лицензии	Не менее 30 суток до истечения	Продление лицензии (п. 6.1)

8.3. Обновление версии

Обновление выполняется в согласованное окно обслуживания.

Рекомендуемый порядок:

1. Изучить release notes целевой версии; выявить изменения, затрагивающие конфигурацию и совместимость (например, включение RBAC по умолчанию в версии 200 нарушает работу интеграций, обращавшихся к OLAP API без аутентификации).
2. Выполнить резервное копирование конфигурации и данных (п. 8.4).
3. Загрузить дистрибутив и образы целевой версии.
4. Перенести значения из действующего файла `.env` в шаблон новой версии; сверить перечень параметров с release notes.
5. Применить новую конфигурацию `docker compose` целевой версии.
6. Проверить состояние всех компонентов и работоспособность по перечню п. 8.5.
7. Проверить актуальность данных на дашбордах и доставку тестового оповещения.
8. При необходимости обновить агенты на контролируемых узлах.

> Обновление в промышленной среде рекомендуется предварительно отработать на тестовом контуре. Порядок обновления для конкретного перехода между версиями уточняется в release notes и у службы технической поддержки вендора.

8.3.1. Обновление с версии 201 до версии 202 (установка на одном сервере)

Перед началом работ объявите окно обслуживания на затронутые объекты (п. 6.7) и выполните полное резервное копирование (п. 8.4).

1. Загрузите архив дистрибутива:

```
   curl --header "PRIVATE-TOKEN:<your_token>" \
   "https://git.proto.group/api/v4/projects/125/packages/generic/protoobp-backend-dist/
202/protoobp-backend-dist-singlehost-202.tar.gz" \
   --output protoobp-backend-dist-singlehost-202.tar.gz
```

2. Распакуйте архив в каталог действующей установки:

```
   tar -xzvf protoobp-backend-dist-singlehost-202.tar.gz -C /opt/protoobp
```

3. Переключитесь на привилегированную учётную запись (`sudo su`) и загрузите новые образы:

```
   cd /opt/protoobp && docker compose -f docker-compose-202.yaml pull
```

4. **Обязательно.** Удалите временные данные Kafka:

```
   rm -rf /opt/protoobp/data/kafka/*
```

5. **Обязательно.** Задайте в файле `.env` параметры почтового сервера отчётности (см. п. 4.3, шаг 2) — без них платформа запустится штатно, но отчёты отправляться не будут:

```
   POBP_REPORT_RUNNER_SMTP_HOST=
   POBP_REPORT_RUNNER_SMTP_USER=
   POBP_REPORT_RUNNER_SMTP_PASSWORD=
```

Там же перевыпустите и укажите пароль служебной учётной записи отчётности в переменной `POBP_REPORT_RUNNER_KEYCLOAK_CLIENT_SECRET`.

6. **Обязательно.** При использовании собственных TLS-сертификатов раскомментируйте соответствующие строки для сервисов `proto-nginx` и `proto-auth` в файле `docker-compose-202.yaml` (см. п. 4.6) — новый файл композиции поставляется с настройками по умолчанию.
7. **Обязательно.** При использовании единого входа (SSO) раскомментируйте либо перенесите соответствующие строки для сервиса `proto-auth` (см. п. 6.5.4).

8. Остановите предыдущую версию:

```
docker compose -f docker-compose-201.yaml down
```

9. Запустите новую версию:

```
docker compose -f docker-compose-202.yaml up -d
```

Запуск всех компонентов занимает 2–4 минуты. При первом запуске автоматически применяются миграции хранилища версии 202 (окна обслуживания, отчётность, журнал аудита) — дополнительных действий не требуется.

10. Выполните проверки по перечню п. 8.5, дополнительно проверив: доступность вкладки `Дашборды и отчёты` → `Отчёты`, доставку тестового отчёта действием «Отправить сейчас» и работу подраздела `Окна обслуживания`.

> **Журнал аудита при обновлении.** Журнал переведён на новую схему со сроками хранения 365 дней для потока «Изменения» и 90 дней для потока «Доступ». Обновление выполняется без перезаписи данных: записи, накопленные в предыдущих версиях, сохраняются и хранятся 365 дней, но не содержат новых полей (состояние «до/после», категория события).
>
> **Установка на нескольких серверах.** Порядок обновления предоставляется контактными лицами со стороны технической поддержки партнёра или вендора.

8.4. Резервное копирование и восстановление

Регламент описывает резервное копирование серверной части в варианте установки **на одном сервере**. Для конфигураций из двух и более серверов порядок копирования предоставляется по запросу в службу технической поддержки.

Далее используются значения путей по умолчанию: каталог установки — `/opt/protoobp`, каталог данных (`POBP_DATA_DIRECTORY`) — `/opt/protoobp/data`. Если при установке значения изменялись, скорректируйте команды.

Агенты резервному копированию не подлежат: их конфигурация задаётся при установке и восстанавливается переустановкой агента.

8.4.1. Что подлежит копированию

Состояние платформы делится на четыре группы с разной ценностью и разной стоимостью копирования.

Группа	Состав	Объём	Критичность	Периодичность
Конфигурация в файлах	`/opt/protoobp/.env`, `docker-compose-202.yaml`, каталог `config/`, файлы `.env_ldap*`, `.env_oidc*`, файлы TLS-сертификата и ключа, каталог `truststores/`	единицы МБ	высокая	после каждого изменения и еженедельно
Учётные записи и права	БД сервиса `proto-auth` (том `postgres_data`): пользователи, роли, группы, параметры подключения LDAP/OIDC	десятки МБ	высокая	ежесуточно
Настройки и журналы в БД	Таблицы настроек и журналов в `data/database`: правила алертинга, каналы и маршруты оповещений, дашборды и папки	единицы–сотни МБ	**высокая**	ежесуточно

Группа	Состав	Объём	Критичность	Периодичность
	дашбордов, бизнес-процессы, ключевые бизнес-транзакции, SLO, окна обслуживания, правила и расписания отчётов, настройки рабочего процесса инцидентов, пороги APDEX, правила извлечения данных, правила нормализации URL, маркеры изменений, настройки модели ресурсов, журнал аудита, история изменений конфигурационных единиц, история изменений рабочего процесса, история запусков отчётов			
Данные телеметрии	Остальное содержимое `data/database` (метрики сервисов, трейсы, инциденты, состояние модели ресурсов, история исполнения бизнес-процессов), `data/metrics`, `data/opensearch`, `data/logs`	десятки–сотни ГБ	средняя	еженедельно либо по решению заказчика

> **Критично для правильной организации копирования.** Настройки, которые администратор и пользователи создают в веб-интерфейсе, хранятся **не в файлах, а в БД** `proto-database` — в том же каталоге `data/database`, что и телеметрия. Резервное копирование только файлов конфигурации из `/opt/protoobp` их **не сохраняет**: при потере каталога данных правила алертинга, каналы оповещений, дашборды, бизнес-процессы, ключевые транзакции, SLO, окна обслуживания и расписания отчётов будут утрачены, а платформа стартует с поставляемыми по умолчанию шаблонами.
>
> Поэтому в регламент обязательно включается **либо** ежесуточная выгрузка таблиц настроек и журналов (п. 8.4.5), **либо** полное копирование каталога данных (п. 8.4.6).
>
> Файлы `alertrules.yml` и `alertpolicies.yml` резервной копией настроек алертинга **не являются**: это генерируемый слепок, который платформа перезаписывает из БД при каждом изменении правил и каналов. Восстановление этих файлов без БД настройки не вернёт — при следующей синхронизации они будут перезаписаны текущим содержимым БД.

Не копируются (восстанавливаются автоматически при запуске платформы): `data/kafka` и `data/zookeeper` — транзитная очередь телеметрии; `data/analyzer` — кеш аналитического слоя; `debug/` — отладочные дампы.

> **О целесообразности копирования телеметрии.** Данные телеметрии хранятся ограниченное время (см. п. 6.2): сырые трейсы — сутки, метрики — недели, агрегаты — до 60 дней. Восстановление недельной копии возвращает данные, значительная часть которых уже утратила актуальность. Минимально достаточный регламент — копирование **конфигурации, учётных записей, а также настроек и журналов из БД**: он позволяет восстановить работоспособную и полностью настроенную платформу за минуты, потеряв только историю телеметрии. Полное копирование данных оправдано, если история метрик и трейсов используется для разбора инцидентов задним числом или для отчётности.

8.4.2. Рекомендуемый регламент

Периодичность	Объект	Останов платформы	Ориентировочная длительность
После каждого изменения `.env` или параметров запуска	конфигурация в файлах	не требуется	секунды
После массовых изменений правил алертинга, каналов оповещения или дашбордов	настройки и журналы из БД	не требуется	секунды
Ежесуточно, ночью	конфигурация + учётные записи + настройки и журналы из БД	не требуется	минуты
Еженедельно, в окно обслуживания	полная копия, включая данные телеметрии	**требуется**	зависит от объёма данных, обычно 20–90 минут
Перед обновлением версии платформы	полная копия	**требуется**	—

Глубина хранения по умолчанию: ежесуточные копии — 14 дней, еженедельные — 8 недель, предобновленческие — до успешной приёмки следующего обновления.

8.4.3. Копирование конфигурации

Выполняется на работающей платформе.

```
BACKUP_DIR=/backup/protoobp/$(date +%F)
mkdir -p "$BACKUP_DIR"

tar -czf "$BACKUP_DIR/protoobp-config.tar.gz" \
  -C /opt/protoobp \
  --exclude=./data --exclude=./debug \
  .
```

Архив содержит `.env` с лицензионным ключом, паролями SMTP и ключом доступа к внешней языковой модели, поэтому подлежит хранению с ограниченным доступом (п. 8.4.9).

8.4.4. Копирование учётных записей и прав доступа

Учётные записи, роли, группы и параметры подключения к каталогу пользователей хранятся в БД сервиса `proto-auth`. Копия снимается на работающей платформе:

```
docker exec proto-postgresql \
  pg_dump -U protoobp -d proto-auth --format=custom \
  > "$BACKUP_DIR/proto-auth.dump"
```

При использовании внешнего каталога пользователей (LDAP/AD или OIDC) сами учётные записи находятся во внешней системе; в копии сохраняются только параметры подключения и локальные назначения ролей.

8.4.5. Копирование настроек и журналов из БД

Выгрузка выполняется на работающей платформе и занимает секунды: перечисленные таблицы малы по сравнению с телеметрией. Такую выгрузку целесообразно выполнять ежесуточно, даже если полное копирование данных не производится.

```
CH="docker exec proto-database clickhouse-client -u protoobp --password proto-clickhouse"
mkdir -p "$BACKUP_DIR/settings"

for T in \
  alert_rules alert_receivers alert_routes \
```

```
  dashboards dashboard_folders dashboard_versions \
  business_processes business_process_steps bp_step_map \
  key_business_transactions slo_definitions \
  maintenance_windows maintenance_window_revisions \
  report_definitions report_runs \
  incident_workflow_config incident_workflow_config_history \
  service_apdex_threshold data_extraction_rules annotations \
  normalized_segments \
  resource_types relationship_types resource_type_ci_mapping \
  ci_classes ci_subclasses resource_retention_policies \
  ci_change_history ci_mapping_quarantine rbac_audit_log
do
  $CH -q "SELECT * FROM proto.$T FORMAT Native" > "$BACKUP_DIR/settings/$T.native"
done

tar -czf "$BACKUP_DIR/protoobp-settings.tar.gz" -C "$BACKUP_DIR" settings && rm -rf
"$BACKUP_DIR/settings"
```

В выгрузку входят как сами настройки, так и связанные с ними журналы, которые невозможно восстановить из других источников: журнал аудита действий пользователей (`rbac_audit_log`), история изменений конфигурационных единиц модели ресурсов (`ci_change_history`), карантин сопоставления типов ресурсов (`ci_mapping_quarantine`), история изменений рабочего процесса инцидентов (`incident_workflow_config_history`), история запусков отчётов (`report_runs`).

Состав таблиц соответствует версии 202; при обновлении платформы перечень следует сверять с примечаниями к выпуску.

В выгрузку **не входит** история исполнения бизнес-процессов (`business_process_execution_steps`) — это телеметрия, привязанная к трейсам, с собственным коротким сроком хранения; она копируется только вместе с каталогом данных.

Восстановление выгрузки выполняется на запущенной платформе после её первичной инициализации (таблицы уже созданы):

```
tar -xzf protoobp-settings.tar.gz

for F in ./settings/*.native; do
  T=$(basename "$F" .native)
  docker exec -i proto-database clickhouse-client -u protoobp --password proto-clickhouse
\
    -q "INSERT INTO proto.$T FORMAT Native" < "$F"
done
```

> **Восстановление поверх существующих данных.** Загрузка выполняется добавлением строк. Настройки следует восстанавливать **в чистую инсталляцию**: при загрузке в платформу, где настройки уже создавались заново, возможны дубликаты объектов, которые придётся удалять вручную через веб-интерфейс. Загрузку выполняйте только один раз.

8.4.6. Копирование данных телеметрии

Хранилища платформы держат часть состояния в памяти, поэтому копирование каталога данных на работающей платформе даёт **неконсистентную и непригодную для восстановления копию**. Полная копия данных снимается только при остановленной платформе.

1. Остановите платформу:

```
cd /opt/protoobp && docker compose -f docker-compose-202.yaml down
```

2. Скопируйте каталог данных, исключив транзитные и кешируемые подкаталоги:

```
tar -czf "$BACKUP_DIR/protoobp-data.tar.gz" \
  -C /opt/protoobp/data \
  --exclude=./kafka --exclude=./zookeeper --exclude=./analyzer \
  .
```

3. Запустите платформу:

```
   cd /opt/protoobp && docker compose -f docker-compose-202.yaml up -d
```

> **Влияние останова.** На время останова телеметрия от агентов не принимается и накапливается на стороне агентов ограниченное время; продолжительный останов приводит к разрыву в данных. Окно следует выбирать в период минимальной нагрузки и заранее объявлять окном обслуживания (п. 6.7), чтобы не порождать ложные оповещения.

8.4.7. Проверка резервной копии

Копия, которую ни разу не разворачивали, резервной копией не является. Не реже одного раза в квартал следует выполнять контрольное восстановление на отдельном сервере и проверять:

- платформа поднимается, веб-интерфейс доступен по адресу из `UI_URL`;
- вход выполняется существующей учётной записью, роли и права сохранены;
- дашборды и правила алертинга присутствуют;
- при восстановлении данных телеметрии — исторические метрики отображаются на дашбордах.

Для быстрой проверки целостности архивов достаточно:

```
tar -tzf "$BACKUP_DIR/protoobp-config.tar.gz" >/dev/null && echo OK
tar -tzf "$BACKUP_DIR/protoobp-data.tar.gz" >/dev/null && echo OK
```

8.4.8. Восстановление

Восстановление выполняется на сервере, подготовленном согласно п. 3.1 и п. 4.1, с установленным Docker. Версия образов платформы должна совпадать с версией, из которой снята копия (указана в `.env` и в имени файла композиции). Если на сервере ещё нет образов платформы, потребуется доступ к репозиторию образов (`docker login`, п. 4.3) либо образы, перенесённые вручную по процедуре offline-установки (п. 4.4).

1. Восстановите конфигурацию:

```
   mkdir -p /opt/protoobp && tar -xzf protoobp-config.tar.gz -C /opt/protoobp
```

2. При восстановлении на другом сервере проверьте в `.env` значения `POBP_COMPUTE` и `UI_URL` и при необходимости скорректируйте их и записи DNS.
3. Восстановите данные телеметрии — только если они копировались:

```
   mkdir -p /opt/protoobp/data && tar -xzf protoobp-data.tar.gz -C /opt/protoobp/data
```

Затем **в любом случае** создайте недостающие каталоги данных и восстановите права доступа:

```
   mkdir -p /opt/protoobp/data/database   && chown -R 101:101
/opt/protoobp/data/database
   mkdir -p /opt/protoobp/data/opensearch && chown -R 1000:1000
/opt/protoobp/data/opensearch
   mkdir -p /opt/protoobp/data/logs       && chown -R 1000:1000   /opt/protoobp/data/logs
   mkdir -p /opt/protoobp/data/metrics    && chown -R 1000:1000
/opt/protoobp/data/metrics
   mkdir -p /opt/protoobp/data/kafka      && chown -R 65532:65532
/opt/protoobp/data/kafka
   mkdir -p /opt/protoobp/data/zookeeper  && chown -R 1001:1001
/opt/protoobp/data/zookeeper
   chown 65532:65532 /opt/protoobp/alertrules.yml /opt/protoobp/alertpolicies.yml
```

4. Запустите платформу и дождитесь инициализации:

```
   cd /opt/protoobp && docker compose -f docker-compose-202.yaml up -d
```

При первом запуске часть контейнеров инициализируется и перезапускается — как и при установке, появление ошибки в выводе `docker compose` на этом этапе является штатным; повторите команду.

5. Восстановите учётные записи:

```
docker compose -f docker-compose-202.yaml stop proto-auth
docker exec -i proto-postgresql \
  pg_restore -U protoobp -d proto-auth --clean --if-exists < proto-auth.dump
docker compose -f docker-compose-202.yaml start proto-auth
```

6. Если каталог данных не восстанавливался, загрузите выгрузку настроек и журналов из БД (п. 8.4.5). Загрузку выполняйте после полной инициализации платформы и **только один раз**, чтобы не создать дубликаты объектов.
7. Проверьте состояние платформы:

```
cd /opt/protoobp && docker compose -f docker-compose-202.yaml ps
```

Все контейнеры должны находиться в состоянии `running`. Затем откройте веб-интерфейс и выполните проверки из п. 8.4.7 и п. 8.5.

Если данные телеметрии не восстанавливались, платформа стартует с пустыми хранилищами и начинает накапливать данные заново. Все пользовательские настройки — правила алертинга, каналы и маршруты оповещений, дашборды, бизнес-процессы, ключевые транзакции, SLO, окна обслуживания, расписания отчётов, настройки модели ресурсов, журнал аудита и история изменений — восстанавливаются из копии `data/database` либо из выгрузки таблиц настроек; файлы `alertrules.yml` и `alertpolicies.yml` платформа перегенерирует из БД самостоятельно. **При отсутствии обеих копий эти объекты будут созданы заново из поставляемых по умолчанию шаблонов, а пользовательские настройки утрачены.**

8.4.9. Хранение копий

- Копии размещаются **вне сервера платформы** — на выделенном хранилище или в системе резервного копирования организации.
- Архив конфигурации и дамп учётных записей содержат секреты (лицензионный ключ, пароли SMTP и служебных учётных записей, ключ доступа к языковой модели) — доступ к ним ограничивается администраторами платформы, при передаче по сети применяется шифрование.
- Права на файлы копий — `0600`, владелец — учётная запись, от имени которой выполняется копирование.
- Факт и результат каждого копирования фиксируются в системе резервного копирования организации; отсутствие успешной копии за сутки должно порождать оповещение.

8.5. Проверка работоспособности после регламентных работ

№	Проверка	Ожидаемый результат
1	Состояние контейнеров	Все `Up`, контейнеры с проверками — `healthy`
2	Журналы компонентов за период запуска	Отсутствие сообщений об ошибке активации лицензии и критических ошибок
3	Открытие веб-интерфейса по адресу `UI_URL`	Отображается страница входа с действующим сертификатом
4	Вход пользователя (в том числе через SSO)	Успешная аутентификация, назначенные роли применены
5	Раздел `Инфраструктура` → `Хосты`	Список хостов полон, данные актуальны
6	Раздел `Приложения` → `Сервисы`	Данные сервисов актуальны, трейсы поступают
7	Раздел `Логи`	Запрос за последние 15 минут возвращает записи
8	Раздел `Алерты` → `Правила`	Перечень правил сохранён, статусы соответствуют ожидаемым
9	Доставка тестового оповещения	Уведомление получено в канале (для канала электронной почты — кнопкой «Отправить тестовое письмо»)

№	Проверка	Ожидаемый результат
10	Метрика `consumer lag`	Не возрастает монотонно
11	Вкладка `Дашборды и отчёты` → `Отчёты`	Перечень отчётов сохранён; действие «Отправить сейчас» завершается успехом
12	Подраздел `Инциденты и Алерты` → `Окна обслуживания`	Перечень окон сохранён, статусы соответствуют ожидаемым
13	`Роли и доступ` → `Журнал аудита`	Журнал доступен, события фиксируются, экспорт работает

9. Информационная безопасность

9.1. Модель обработки данных

Схема обработки: **Приложение → Трейсер ProtoOBP → Агент ProtoOBP → Серверная часть ProtoOBP (on-premise)**.

Собираемая информация хранится исключительно на серверах Заказчика и не покидает его периметр. Передача данных во внешние облачные системы не производится.

Исключение составляет подключение внешней языковой модели для ИИ-функций модуля «Инциденты» (п. 4.3, шаг 2) — при её использовании фрагменты телеметрии передаются за пределы контура. Функция является опциональной и по умолчанию отключена.

9.2. Категории данных, потенциально содержащих конфиденциальные сведения

Категория	Описание
Запросы к базам данных	Литералы, последовательности литералов или связующие переменные в операторах СУБД
Параметры URI	Значения параметров в переменной части пути или строки запроса URI
Данные о пользователе в URI	Фрагменты URI, потенциально содержащие имя пользователя

Обфускация — замена строковых параметров на символ `?`.

9.3. Механизмы обфускации

Механизмы предусмотрены на двух уровнях и в большинстве своём включены по умолчанию:

Уровень трейсера (библиотеки инструментации). Поведение по умолчанию зависит от языка:

Язык	Запросы к БД	Параметры URI	Примечание
Java	Собираются, обфусцируются	Собираются, обфусцируются	—
PHP	Собираются, обфусцируются	Собираются, обфусцируются	Данные о пользователе в URI собираются и обфусцируются
Python	Собираются, обфусцируются	Собираются, обфусцируются	Параметры URI по умолчанию **выключены**, включаются явно
Ruby	Собираются, обфусцируются	Собираются, обфусцируются	—
.NET	Собираются, обфусцируются	Собираются, **не обфусцируются**	Требует внимания при обработке чувствительных данных в URI
Node.js	Собираются, **не обфусцируются**	Собираются, обфусцируются	Параметры URI по умолчанию выключены
Go	Собираются, обфусцируются на уровне агента	Серверные путь и строка запроса обфусцируются; клиентские — нет	Тело HTTP и cookies обфусцируются

Дополнительно на уровне трейсера настраиваются: изменение query strings, состав собираемых HTTP-заголовков.

Уровень агента. Доступны: обфускация данных в именах ресурсов, обфускация содержимого трейсов, замена тегов, игнорирование заданных ресурсов.

Инструмент контроля (начиная с версии 203). Проверить фактический состав собираемых данных позволяет вкладка `Роли и доступ` → `Проверка на ПДн` (п. 7.3.8) — выборочная проверка телеметрии на наличие персональных данных. Проверка выполняется после подключения каждого нового приложения или источника и после изменения правил обфускации. Ввиду выборочного характера она дополняет, но не заменяет анализ состава данных, описанный ниже.

Обязательное действие при вводе в эксплуатацию. До подключения к Системе приложений, обрабатывающих персональные данные, сведения, составляющие банковскую или коммерческую тайну, или иные сведения ограниченного доступа, проведите совместно с подразделением информационной безопасности анализ фактического состава собираемых трейсов и настройте дополнительные правила обфускации и игнорирования ресурсов. Отдельного внимания требуют комбинации «язык — категория», отмеченные в таблице как необфусцируемые по умолчанию.

9.4. Организационно-технические меры

Мера	Порядок реализации
Смена реквизитов по умолчанию	Пароль встроенной учётной записи `admin` изменяется немедленно после установки
Защита канала доступа	Подключение доверенного TLS-сертификата (п. 4.6); доступ по HTTP допустим только в изолированном сегменте на этапе отладки
Централизованная аутентификация	Подключение SSO (LDAP / OIDC) с применением парольной политики организации (п. 6.5)
Разграничение доступа	Применение ролевой модели с ограничением области видимости данных (п. 7.3)
Контроль действий	Ведение и регулярный анализ журнала аудита (п. 7.3.7): поток «Изменения» (срок хранения 365 дней) и поток «Доступ» (90 дней), быстрые срезы «Только отказы» и «Изменения прав доступа», состояние объекта «до» и «после»; экспорт в CSV/JSON и передача событий в SIEM
Планирование регламентных работ	Объявление окон обслуживания (п. 6.7) до начала работ; контроль области действия окон по журналу аудита — чрезмерно широкая область может скрыть реальные отказы
Защита рассылок	Ограничение перечня получателей отчётов (п. 6.8); учёт того, что PDF-отчёт содержит сведения о состоянии информационных систем и направляется по внешним каналам электронной почты
Защита реквизитов	Ограничение прав доступа к файлам `.env`, `.env_ldap`, `.env_oidc*` на уровне файловой системы; исключение их из систем контроля версий
Сегментация сети	Ограничение сетевых доступов перечнем из п. 3.3; закрытие портов агентов (8125, 8126) на внешнем периметре
Защита резервных копий	Хранение копий конфигурационных файлов в зашифрованном виде
Контроль целостности дистрибутивов	Проверка источника и целостности загружаемых артефактов до установки
Управление уязвимостями	Своевременное обновление версий платформы и агентов; контроль release notes

9.5. Особенности применения ИИ-функций

При использовании ИИ-сводки, ИИ-расследования, автоматического черновика постмортема, а также раздела «AI-аналитик» (начиная с версии 203):

- за пределы контура организации передаются наименования сервисов и ресурсов, тексты алертов и хронология инцидента;
- ключ доступа `POBP_AI_LLM_API_KEY` является секретом и в журналы не выводится, однако подлежит защите наравне с иными реквизитами;
- решение о включении функций принимается совместно с подразделением информационной безопасности;
- при обработке сведений ограниченного доступа рекомендуется применять языковую модель, размещённую в контуре организации, с OpenAI-совместимым интерфейсом;
- результаты работы модели носят рекомендательный характер и подлежат проверке специалистом.

9.6. Сведения о включении в реестр российского программного обеспечения

Программное обеспечение **Proto Observability Platform** включено в **единый реестр российских программ для электронных вычислительных машин и баз данных**, ведение которого осуществляется Министерством цифрового развития, связи и массовых коммуникаций Российской Федерации в соответствии с постановлением Правительства Российской Федерации от 16.11.2015 № 1236.

Реквизит	Значение
Номер реестровой записи	14369
Дата реестровой записи	03.08.2022
Основание включения	Поручение Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 03.08.2022 по протоколу заседания экспертного совета от 25.07.2022 № 1041пр

Формулировка для включения в проектную и закупочную документацию:

> Запись в реестре № 14369 от 03.08.2022 произведена на основании поручения Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 03.08.2022 по протоколу заседания экспертного совета от 25.07.2022 № 1041пр.

Практические следствия для эксплуатанта:

- ПО относится к программному обеспечению российского происхождения и может применяться при закупках, на которые распространяется установленный запрет на допуск иностранного программного обеспечения (постановление Правительства Российской Федерации от 16.11.2015 № 1236);
- ПО применимо в проектах перехода на преимущественное использование российского программного обеспечения, в том числе на объектах критической информационной инфраструктуры, в рамках соответствующих ведомственных планов;
- при подготовке обоснования закупки и при формировании комплекта документации на информационную систему в состав приложений включается выписка из реестра с указанными выше реквизитами;
- актуальность реестровой записи подлежит проверке на официальном ресурсе реестра на дату подготовки документации: состав сведений реестра может изменяться.

9.7. Соответствие требованиям в области защиты информации

При эксплуатации Системы в составе информационных систем, к которым предъявляются требования законодательства Российской Федерации в области защиты информации, администратору необходимо:

- определить категорию обрабатываемой информации и применимые требования (в том числе Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных» и нормативных актов ФСТЭК России, если Система обрабатывает сведения соответствующих категорий);
- учитывать, что телеметрия (трейсы, логи, параметры URI) может содержать персональные данные — в этом случае состав собираемых данных подлежит минимизации средствами обфускации и игнорирования ресурсов (п. 9.3);
- согласовать сроки хранения данных (п. 6.2) с требованиями к срокам хранения соответствующих категорий информации;
- обеспечить регистрацию событий безопасности и их передачу в систему централизованного сбора (п. 7.3.7);
- выполнять периодическую проверку телеметрии на наличие персональных данных средствами вкладки «Проверка на ПДн» (п. 7.3.8, начиная с версии 203) и хранить результаты проверок как подтверждение принятых мер по минимизации состава обрабатываемых данных;
- приобщить к комплекту документации на информационную систему выписку из реестра российского программного обеспечения (п. 9.6);
- уточнить у вендора актуальные сведения о наличии сертификатов соответствия требованиям по безопасности информации и приобщить подтверждающие документы, если такие требования предъявляются к информационной системе.

> **Разграничение понятий.** Включение ПО в единый реестр российских программ (п. 9.6) подтверждает **российское происхождение** программного обеспечения и является основанием для его применения при закупках с ограничением допуска иностранного ПО. Реестровая запись **не является** сертификатом соответствия требованиям по безопасности информации и не заменяет аттестацию информационной системы. Сведения о наличии сертификатов соответствия ФСТЭК России уточняются у вендора отдельно.

Настоящий документ не является заключением о соответствии Системы требованиям по безопасности информации; оценка соответствия проводится в установленном порядке.

---

10. Аварийные ситуации и восстановление

10.1. Отсутствие трейсов и метрик от приложений

Диагностика выполняется последовательно по уровням.

Уровень 1. Приложение

Убедиться, что приложение запущено, обрабатывает запросы и на нём фактически создаётся нагрузка. При отсутствии вызовов данные не формируются.

Уровень 2. Трейсер

- Проверить, что трейсер подключён к приложению корректным для данного языка способом и что версия трейсера совместима с версией среды исполнения.
- Проверить параметры подключения трейсера к агенту: адрес агента (`POBP_AGENT_HOST`), порты 8125/8126.
- Изучить журнал приложения на наличие сообщений трейсера об ошибках инициализации и передачи данных.
- Для приложений в контейнерах — убедиться в наличии `extra_hosts: - "host.docker.internal:host-gateway"` и доступности агента с контейнера.

Уровень 3. Агент

- Проверить состояние службы: `sudo protoobp-agent status` (либо соответствующая команда платформы, п. 5.8).
- Проверить корректность `api_key`, `pobp_url`, `apm_pobp_url` в файле конфигурации.
- Проверить сетевую доступность серверной части с контролируемого узла по портам 80/443.
- Изучить журналы агента в `/var/log/protoobp/` (Linux) или `C:\ProgramData\Protoobp\logs` (Windows).
- Убедиться, что хост отображается в разделе `Инфраструктура` → `Хосты` (при отсутствии — обновить страницу браузера целиком).

Уровень 4. Серверная часть

- Проверить состояние всех контейнеров (`healthy`, отсутствие циклических перезапусков).
- **Проверить свободное место на томе данных.** При остатке менее 10 % свободного места **запись трейсов прекращается**.
- Проверить метрику `consumer lag` с тегом `protoobp-consumer` на дашборде `Инфраструктура` → `Kafka` → хост серверной части. Монотонный рост значения без снижения указывает на нехватку системных ресурсов сервера — требуется обращение в техническую поддержку.
- Изучить журналы `proto-trace-processor`, `proto-trace-receiver`, `proto-metric-receiver`.

10.2. Типовые нештатные ситуации и порядок их устранения

№	Признак	Причина	Действия
1	Ошибка контейнера `proto-storage` при первом запуске	Не выполнено создание каталогов данных с требуемыми правами	Выполнить шаг 3 п. 4.3 и перезапустить компоненты
2	Постоянный перезапуск `proto-data-analyzer-router`, `proto-data-analyzer-`	Процессор сервера не поддерживает инструкции AVX	В `docker-compose-202.yaml` заменить тег образов с `202` на `202-non-`

№	Признак	Причина	Действия
	`worker-1`, `proto-data-analyzer-worker-2`		`avx`, выполнить `docker compose -f docker-compose-202.yaml up -d`
3	Ошибка при первом запуске сразу после `up -d`	Штатная инициализация и перезапуск отдельных компонентов	Прервать команду (`Ctrl+C`) и повторить `docker compose ... up -d`
4	В журнале `proto-auth`: `SSLHandshakeException`, `PKIX path building failed`, `unable to find valid certification path`	Контейнер не доверяет сертификату внешней системы (OIDC, LDAPS, SMTP)	Настроить truststore по п. 6.5.5
5	Ошибка активации лицензии в журнале `proto-trace-processor`	Некорректный лицензионный ключ, наличие переносов строк в значении, несовпадение имени компании	Проверить `POBP_LICENSE_KEY` (одна строка) и `POBP_LICENSE_COMPANY_NAME`, применить изменения (п. 6.1)
6	Агент не устанавливается на Alt Linux / ROSA Linux: `Неудовлетворенные зависимости: glibc-common`	Особенность состава пакетов дистрибутива	Установить с ключом `--nodeps` (п. 5.2.1)
7	На дистрибутиве без systemd: `service protoobp-agent start` возвращает `unrecognized service`	Пакет не разворачивает SysV-скрипты	Установить init-скрипты для основного и trace-агента (п. 5.2.3)
8	Служба агента Windows не запускается	Некорректная конфигурация или недостаточные права учётной записи службы	Проверить `protoobp.yaml`, права учётной записи, журнал `%TEMP%\MSI*.LOG`
9	Ошибка Windows «Сервис помечен для удаления»	Открыта оснастка, удерживающая дескриптор службы	Закрыть оснастку «Службы» и консоли; при сохранении — перезагрузить узел
10	Внешние интеграции возвращают `401 TOKEN_MISSING`	С версии 200 RBAC включён по умолчанию, требуется Bearer-токен	Создать интеграционную учётную запись и перевести интеграцию на авторизацию по токену (п. 7.4)
11	Пользователь с ограниченной ролью видит все данные	Аддитивность прав: назначена дополнительная роль с открытым измерением, в том числе через внешнюю группу	Проверить полный перечень ролей пользователя и членство во внешних группах (п. 7.3.5)
12	Прекратилась запись трейсов	Свободное место на томе данных менее 10 %	Освободить место, расширить том, сократить сроки хранения (п. 6.2)
13	Уведомления об алертах не доставляются	Не нажата кнопка «Применить»; правило отключено; несовпадение лейблов с матчерами; неверные реквизиты канала; действует окно обслуживания, охватывающее объект	Проверка по перечню «Руководства пользователя», п. 5.2; проверить перечень активных окон обслуживания
14	Отчёты не доставляются получателям, в истории запусков — ошибка отправки	Не заданы параметры `POBP_REPORT_RUNNER_SMTP_*` либо они неверны	Задать параметры в `.env` (п. 4.3, шаг 2) и применить конфигурацию; изучить текст ошибки в истории запусков отчёта
15	Отчёты приходят с предупреждением: часть виджетов не отрисована	Источник данных виджета не ответил в отведённое время	Проверить работоспособность источников данных дашборда; при систематическом повторении

№	Признак	Причина	Действия
			— сократить число виджетов или период дашборда
16	После обновления до версии 202 компоненты не стартуют или наблюдаются ошибки обработки телеметрии	Не выполнена очистка временных данных Kafka	Выполнить `rm -rf /opt/protoobp/data/kafka/*` и повторно запустить платформу (п. 8.3.1, шаг 4)
17	После обновления перестал работать вход через SSO либо используется самоподписанный сертификат вместо собственного	В новом файле композиции не восстановлены строки для сервисов `proto-auth` и `proto-nginx`	Раскомментировать либо перенести соответствующие строки (п. 4.6, п. 6.5.4) и применить конфигурацию
18	После восстановления из резервной копии отсутствуют правила алертинга, дашборды, SLO, окна обслуживания	Копировались только файлы конфигурации; настройки хранятся в БД	Восстановить каталог данных либо выгрузку таблиц настроек (п. 8.4.1, п. 8.4.5)
19	После восстановления настроек появились дубликаты дашбордов и правил	Выгрузка настроек загружена в инсталляцию, где объекты уже создавались заново, либо загружена повторно	Удалить дубликаты через веб-интерфейс; загружать выгрузку только в чистую инсталляцию и однократно (п. 8.4.5)

10.3. Повреждение данных хранилища трейсов

Признаки. В журнале компонента `proto-database` появляются сообщения вида `Part ... is broken and needs manual correction`; компонент `proto-trace-processor` циклически перезапускается с сообщениями:

```
DB::Exception: Suspiciously big size (870 parts, 3.14 GiB in total) of all broken parts
to remove
while maximum allowed broken parts size is 1.00 GiB.
```

или

```
DB::Exception: Suspiciously many (870 parts, 3.14 GiB in total) broken parts to remove
while maximum allowed broken parts count is 100.
```

Типовая причина. Внеплановое отключение работающего сервера либо ошибки файловой системы.

Порядок устранения:

1. На сервере с ролью `proto-database` в каталоге продукта создайте файл `merge_tree_settings.xml`:

```
   <clickhouse>
       <merge_tree>
           <max_suspicious_broken_parts>2000</max_suspicious_broken_parts>

<max_suspicious_broken_parts_bytes>5000000000</max_suspicious_broken_parts_bytes>
       </merge_tree>
   </clickhouse>
```

2. В `docker-compose-<version>.yaml` для сервиса `proto-database` добавьте том с этим файлом:

```
   proto-database:
     container_name: proto-database
     volumes:
       - ${POBP_DATA_DIRECTORY}/database:/var/lib/clickhouse
       -
./merge_tree_settings.xml:/etc/clickhouse-server/config.d/merge_tree_settings.xml:ro
```

3. Примените изменения и проконтролируйте запуск компонентов.

> **Внимание.** Указанные параметры повышают допустимый объём удаляемых повреждённых частей данных. Повреждённые данные при этом **удаляются** — часть исторической телеметрии будет утрачена. После восстановления работоспособности верните параметры к значениям, соответствующим политике эксплуатации, и устраните первопричину (некорректное завершение работы сервера, неисправность дисковой подсистемы).

10.4. Сбор технических журналов для службы поддержки

Создайте на каждом сервере, где размещены компоненты Системы, скрипт `collect-logs.sh`:

```
#!/usr/bin/env bash
set -euo pipefail

COMPOSE_FILE="docker-compose-202.yaml"
LOG_OPTS="${*:-}"

STAMP="$(date +%Y%m%d_%H%M%S)"
HOST="$(hostname -s)"
WORKDIR="logs_${HOST}_${STAMP}"
ARCHIVE="${WORKDIR}.tar.gz"

mkdir -p "$WORKDIR"

containers="$(docker compose -f "$COMPOSE_FILE" ps -q || true)"
if [ -z "$containers" ]; then
  echo "No containers found for compose file: $COMPOSE_FILE"
  exit 1
fi

for cid in $containers; do
  name="$(docker inspect --format '{{.Name}}' "$cid" | sed 's|/||')"
  docker logs $LOG_OPTS "$cid" >"${WORKDIR}/${name}.log" 2>&1 || true
done

docker compose -f "$COMPOSE_FILE" config > "${WORKDIR}/compose_effective.yaml"
2>/dev/null || true

tar -czf "$ARCHIVE" "$WORKDIR"
echo "Логи упакованы: $ARCHIVE"
rm -rf "$WORKDIR"
```

Запуск:

```
chmod +x collect-logs.sh
./collect-logs.sh              # все журналы
./collect-logs.sh --since 24h  # журналы за последние сутки
```

Полученные архивы передайте в службу технической поддержки.

> **Перед передачей.** Журналы могут содержать имена хостов, сервисов, адреса и фрагменты запросов. Оцените допустимость их передачи в соответствии с регламентом организации; при необходимости согласуйте передачу с подразделением информационной безопасности.

10.5. Сбор расширенной диагностической информации

Выполняется по запросу вендора на основном сервере с ролью `proto-compute`, из каталога продукта (обычно `/opt/protoobp`).

Создайте скрипт `collect-debug.sh`:

```
#!/bin/bash
set -e

if [ -z "$1" ]; then
  echo "Период сбора диагностики не указан."
```

```
  echo "Использование: $0 <время_в_секундах>"
  exit 1
fi

duration=$1

if [ -f "docker-compose-202.yaml" ]; then
  compose_file="docker-compose-202.yaml"
elif [ -f "docker-compose.yaml" ]; then
  compose_file="docker-compose.yaml"
else
  echo "Файл docker-compose не найден. Запустите скрипт из каталога продукта (обычно
/opt/protoobp)."
  exit 1
fi

if [ ! -d "debug/ap" ]; then
  mkdir -p debug/ap
  chmod 777 debug/ap
fi

echo "POBP_TRACE_PROCESSOR_DEBUG_SAVE_AP_TO_FILE: true" >> debug/config.yaml
docker compose -f $compose_file restart proto-trace-processor proto-nginx
sleep $duration
sed -i '$d' debug/config.yaml
docker compose -f $compose_file restart proto-trace-processor proto-nginx

apdate=$(date +'%Y.%m.%d-%H-%M-%S')
tar czf debug/incoming-agent-payloads-${apdate}.tar.gz debug/ap/incoming*
rm -rf debug/ap/*
echo "Итоговый файл: debug/incoming-agent-payloads-${apdate}.tar.gz"
```

Запуск (параметр — время сбора в секундах):

```
chmod +x collect-debug.sh
./collect-debug.sh 60
```

> **Внимание.** Режим отладки приводит к перезапуску компонентов `proto-trace-processor` и `proto-nginx` — в моменты перезапуска возможны кратковременные потери поступающей телеметрии. Собираемые файлы содержат **необработанные полезные нагрузки агентов**, то есть фактические данные телеметрии. Выполняйте сбор в согласованное окно и оценивайте состав данных перед передачей.

10.6. Общий порядок действий при аварии

1. **Зафиксировать** время начала, признаки, состав затронутых компонентов; сделать снимки экрана.
2. **Не выполнять** удаление данных, томов и конфигурационных файлов до выяснения причины.
3. **Собрать** состояние контейнеров и журналы за интервал, охватывающий момент возникновения (п. 10.4).
4. **Проверить** ресурсы сервера: свободное место, загрузку CPU и памяти, состояние дисковой подсистемы.
5. **Устранить** ситуацию по перечню п. 10.2, если она в нём описана.
6. **Обратиться** в службу технической поддержки, приложив собранные материалы (п. 11).
7. **Восстановить** конфигурацию из резервной копии, если восстановление иным способом невозможно (п. 8.4).
8. **Проверить** работоспособность по перечню п. 8.5.
9. **Оформить** разбор ситуации и внести корректирующие мероприятия в регламент эксплуатации.

10.7. Действия при обнаружении признаков несанкционированного вмешательства

При обнаружении неавторизованных изменений конфигурации, неизвестных учётных записей, аномальных записей в журнале аудита RBAC или иных признаков компрометации:

1. Незамедлительно уведомить подразделение информационной безопасности организации.
2. **Не удалять** журналы компонентов и журнал аудита RBAC — они являются источником сведений для расследования.
3. Зафиксировать состояние: выгрузить журнал аудита, собрать журналы компонентов (п. 10.4), сохранить копии конфигурационных файлов.
4. По решению подразделения информационной безопасности: заблокировать скомпрометированные учётные записи, отозвать интеграционные токены, сменить пароли и секреты (`POBP_LICENSE_KEY` не является секретом доступа, но пароли SMTP, LDAP, ключ языковой модели и реквизиты Docker-реестра подлежат смене).
5. Восстановить конфигурацию из заведомо доверенной резервной копии.
6. Провести анализ причин и внести изменения в порядок предоставления доступа.

11. Взаимодействие со службой технической поддержки

11.1. Состав сведений при обращении

При обращении подготовьте:

- версию платформы и версии агентов;
- описание ситуации, признаки, время возникновения с указанием часового пояса;
- перечень затронутых компонентов и объектов мониторинга;
- вывод команды состояния контейнеров;
- архив журналов, собранный по п. 10.4;
- при запросе вендора — архив расширенной диагностики по п. 10.5;
- сведения об изменениях, предшествовавших ситуации (обновление, изменение конфигурации, работы на инфраструктуре).

11.2. Каналы получения конфигураций и инструкций по запросу

По запросу в службу технической поддержки вендора предоставляются:

- конфигурация развёртывания серверной части на двух и более серверах (п. 4.7);
- поддержка дополнительных провайдеров SSO (OIDC и SAML 2.0-совместимых систем);
- порядок обновления между конкретными версиями;
- рекомендации по расширению ресурсов при росте объёма телеметрии.

12. Приложения

Приложение А. Соответствие документа требованиям нормативных документов

Требование	Нормативный документ	Раздел настоящего документа
Общие сведения о программе, назначение, функции	ГОСТ 19.503-79, п. 2.1	1.1, 2.1, 2.2
Структура программы, сведения о составных частях и связях между ними	ГОСТ 19.503-79, п. 2.2	2.2, 2.3
Настройка программы на условия конкретного применения	ГОСТ 19.503-79, п. 2.3	4, 5, 6
Проверка программы, способы проверки работоспособности	ГОСТ 19.503-79, п. 2.4	4.3 (шаг 6), 8.5
Дополнительные возможности	ГОСТ 19.503-79, п. 2.5	4.4, 4.5, 4.7, 5.5.2, 6.6
Сообщения системному программисту, тексты сообщений и действия по ним	ГОСТ 19.503-79, п. 2.6	10.1–10.3
Условия применения, требования к техническим и программным средствам	РД 50-34.698-90	3
Порядок установки и первоначальной настройки	РД 50-34.698-90	4, 5, 6
Администрирование доступа, регистрация событий	РД 50-34.698-90	7
Аварийные ситуации и восстановление	РД 50-34.698-90	10

Приложение Б. Сводная таблица параметров файла .env

Группа	Параметр	Назначение	Раздел
Лицензия	`POBP_LICENSE_COMPANY_NAME`	Имя компании-владельца лицензии	4.3
Лицензия	`POBP_LICENSE_KEY`	Лицензионный ключ (одна строка)	4.3, 6.1
Адресация	`POBP_COMPUTE`	Имя хоста для подключения агентов	4.3
Адресация	`UI_URL`	Адрес доступа к веб-интерфейсу	4.3
Хранение	`POBP_DATA_DIRECTORY`	Каталог хранения данных	4.3
TLS	`POBP_SSL_CERT`, `POBP_SSL_CERT_KEY`	Пути к сертификату и ключу	4.6
Реестр	`DOCKER_REGISTRY`	Адрес локального реестра образов	4.5
Аутентификация	`AUTH_TYPE`	Тип провайдера: `LDAP`, `OIDC`, `OIDC-IS4`	6.5.2
SMTP (приглашения)	`SMTP_HOST`, `SMTP_PORT`, `SMTP_AUTH`, `SMTP_USER`, `SMTP_PASSWORD`, `SMTP_FROM`, `SMTP_SSL`, `SMTP_STARTTLS`	Параметры почтового сервера для приглашений пользователей	6.4
SMTP (отчёты)	`POBP_REPORT_RUNNER_SMTP_HOST`, `POBP_REPORT_RUNNER_SMTP_USER`, `POBP_REPORT_RUNNER_SMTP_PASSWORD`	Параметры почтового сервера для рассылки отчётов (начиная с версии 202)	4.3, 6.8
Отчётность	`POBP_REPORT_RUNNER_KEYCLOAK_CLIENT_SECRET`	Пароль служебной учётной записи отчётности; подлежит перевыпуску в промышленной эксплуатации	4.3, 6.8

Группа	Параметр	Назначение	Раздел
ИИ	`POBP_AI_LLM_BASE_URL`, `POBP_AI_LLM_API_KEY`, `POBP_AI_LLM_MODEL`	Подключение внешней языковой модели	4.3, 9.5
Хранение данных	`POBP_DATA_RETENTION_*`, `POBP_CORE_*_DATA_TTL`	Сроки хранения по видам данных	6.2

Приложение В. Расположение конфигурационных файлов и журналов

Объект	Расположение
Каталог продукта (серверная часть)	`/opt/protoobp`
Параметры окружения	`/opt/protoobp/.env`
Файл композиции	`/opt/protoobp/docker-compose-<version>.yaml`
Настройки SSO	`/opt/protoobp/.env_ldap`, `.env_oidc`, `.env_oidc_is4`
Доверенные сертификаты	`/opt/protoobp/truststores/`
Конфигурация алертинга	`/opt/protoobp/alertrules.yml`, `/opt/protoobp/alertpolicies.yml`
Данные	`/opt/protoobp/data` (или значение `POBP_DATA_DIRECTORY`)
Журналы компонентов	`docker compose logs <сервис>`; веб-интерфейс: `Диагностика` → `Логи бэкенда`
Конфигурация агента (Linux)	`/etc/protoobp-agent/protoobp.yaml`
Конфигурация модулей агента (Linux)	`/etc/protoobp-agent/conf.d/<технология>.d/conf.yaml`
Журналы агента (Linux)	`/var/log/protoobp/`
Конфигурация агента (Windows)	`%ProgramData%\Protoobp\protoobp.yaml`
Конфигурация модулей агента (Windows)	`%ProgramData%\Protoobp\conf.d`
Журналы агента (Windows)	`C:\ProgramData\Protoobp\logs`

Приложение Г. Чек-лист ввода в промышленную эксплуатацию

№	Мероприятие	Раздел	Отметка
1	Сервер соответствует требованиям по ресурсам	3.1	☐
2	Сетевые доступы согласованы и открыты	3.3	☐
3	Лицензионный сертификат получен, ключ применён, ошибок активации нет	4.3, 6.1	☐
4	Все компоненты в состоянии `healthy`, циклических перезапусков нет	4.3	☐
5	Пароль встроенной учётной записи `admin` изменён	4.3, 9.4	☐
6	Подключён доверенный TLS-сертификат	4.6	☐
7	Настроен SMTP	6.4	☐
8	Настроен и проверен вход через SSO	6.5	☐
9	Определены роли, ограничения области видимости, назначены пользователям	7.3	☐
10	Заведены отдельные учётные записи для интеграций	7.4	☐

№	Мероприятие	Раздел	Отметка
11	Сроки хранения данных согласованы и заданы	6.2	☐
12	Пороги `APDEX T` заданы по согласованию с владельцами сервисов	6.3	☐
13	Агенты установлены на все контролируемые узлы	5	☐
14	Трейсеры подключены к целевым приложениям	5.9	☐
15	Каналы, политики и правила алертинга настроены и применены	«Руководство пользователя», 4.8	☐
16	Доставка тестового оповещения подтверждена	8.5	☐
17	Состав собираемых данных проверен, правила обфускации настроены	9.3	☐
18	Настроена передача событий аудита в SIEM	7.3.7	☐
19	Настроено и проверено резервное копирование	8.4	☐
20	Восстановление из резервной копии проверено на тестовом контуре	8.4	☐
21	Регламент контроля состояния введён в действие	8.2	☐
22	Решение о применении ИИ-функций согласовано с ИБ	9.5	☐
22а	Выписка из реестра российского ПО приобщена к документации на ИС	9.6	☐
22б	Заданы параметры почтового сервера отчётности; пароль служебной учётной записи отчётности перевыпущен	4.3, 6.8	☐
22в	Проверена доставка тестового отчёта действием «Отправить сейчас»	6.8, 8.5	☐
22г	Выданы права на ресурсы `maintenance_window` и `reports`; регламент объявления окон обслуживания введён в действие	6.7, 6.8, 7.3.4	☐
22д	Проверена ежесуточная выгрузка настроек и журналов из БД (не только файлов конфигурации)	8.4.1, 8.4.5	☐
22е	Выполнена проверка телеметрии на наличие персональных данных, результаты приобщены к материалам по защите информации	7.3.8, 9.7	☐
22ж	Проверено отображение фактического объёма принимаемой и хранимой телеметрии; сроки хранения согласованы с ёмкостью дисковой подсистемы	8.1.1, 6.2	☐
23	Персонал ознакомлен с «Руководством пользователя»	—	☐
24	Определён порядок обращения в службу технической поддержки	11	☐

Приложение Д. Термины версий и совместимость

Изменение	Версия	Влияние на эксплуатацию
Настройка порогов APDEX перенесена в веб-интерфейс	199	Конфигурационные файлы для APDEX более не используются
Ролевая модель RBAC включена по умолчанию; запросы к OLAP API требуют Bearer-токен	200	Интеграции без аутентификации получают `401 TOKEN_MISSING` — требуется миграция
Разделы «Логи», «PCM», «Администрирование RBAC»; источник данных LogsQL для виджетов	200	Новые функциональные возможности
Автоматическая генерация правил алертинга по SLO	200	Правила не редактируются напрямую

Изменение	Версия	Влияние на эксплуатацию
Модуль «Инциденты» с ИИ-функциями	201	Требуется подключение внешней языковой модели
Grafana-совместимый API маркеров событий `/api/annotations`	201	Прежний способ через `marker.sh` продолжает работать
Ограничение прав по `host` и `service_group`	201	Расширение возможностей разграничения доступа
Разделы «Алерты» и «Инциденты» объединены в «Инциденты и Алерты»; разделы «Базы данных», «Сеть», «Очереди сообщений», «Веб-серверы» перенесены в каталог «Инфраструктура»	202	Изменение расположения разделов; прежние адреса дашбордов продолжают работать
Окна обслуживания (`maintenance_window`)	202	Новый ресурс ролевой модели; подавление оповещений и инцидентов на время регламентных работ
Отчёты по электронной почте (`reports`)	202	Требуются параметры `POBP_REPORT_RUNNER_SMTP_*` и перевыпуск `POBP_REPORT_RUNNER_KEYCLOAK_CLIENT_SECRET`; добавлены сервисы `proto-report-runner` и `proto-report-renderer`
Правила алертинга группы `NO_DATA`	202	Оповещение о прекращении поступления метрик от источника
Переработанный журнал аудита	202	Потоки «Изменения» и «Доступ», состояние «до/после», экспорт CSV/JSON, сроки хранения 365 и 90 дней
Флейм-граф и выгрузка трейса, вкладка «Кардинальность», виджеты «Соты здоровья» и «KPI-карточка», инциденты как источник данных	202	Новые функциональные возможности интерфейса
Приём ошибок из открытых Sentry SDK	202	Настраивается переменной `POBP_TRACE_PROCESSOR_SENTRY_PROJECT_MAP` сервиса `proto-trace-processor`
При обновлении на 202 обязательна очистка `data/kafka`	202	Шаг 4 процедуры обновления (п. 8.3.1)
Подраздел «Надёжность изменений» (метрики класса DORA)	203	Считается по маркерам релизов и развёртываний и по инцидентам платформы; дополнительной настройки не требует
Раздел «AI-аналитик»	203	Чат по данным платформы; требует подключения внешней языковой модели (`POBP_AI_LLM_*`)
Вкладка «Проверка на ПДн»	203	Выборочная проверка телеметрии на наличие персональных данных
Пункты диагностики «Объём данных» и «Реестр конфигурации»	203	Доступны администраторам; «Реестр конфигурации» — только на чтение
Актуальная версия агента — 7.80.2	—	Обновление выполняется установкой пакета поверх прежней версии; для RPM используется `rpm -Uvh`

Конец документа