Руководство оператора AgentOps

Практическое руководство по **ежедневной работе** с платформой AgentOps: вход, интерфейс, агенты, сценарии, стоимость, политики, аудит, аномалии и согласования (HITL).

PDF: полное руководство · отдельные главы

> **Для кого это руководство**
>
> Операторы, аналитики SOC, разработчики агентов и администраторы тенанта, которым нужно **пользоваться** уже развёрнутой системой.
> Развёртывание и инфраструктура — в Admin Guide и Deployment.
> Англоязычный User Guide: User Guide.

Что такое AgentOps

AgentOps — операционная панель управления AI-агентами: оркестрация, контроль затрат, маркетплейс и безопасность (ядро AegisAI).

Возможность	Зачем оператору
Рабочие процессы (Workflows)	Цепочки агентов, расписание cron, webhook, пауза на согласование
Стоимость (Cost)	Токены, задержки, бюджеты и остановка при превышении
Агенты и Discovery	Реестр агентов, «теневой» AI без регистрации
Политики	GDPR / HIPAA / ФЗ-152, OPA/Rego, fail-closed
Аудит и SIEM	Подписанный журнал действий, экспорт
Аномалии	ML-детектор отклонений поведения
HITL	Ручное одобрение рискованных действий
Маркетплейс	Публикация и установка пакетов агентов

Продакшен: agentops.fun · Документация: agentops.fun/documentation

Быстрый старт для оператора

1. Откройте веб-интерфейс (локально: `http://localhost:5173`, прод: `https://agentops.fun`).
2. Войдите по email/паролю или API-ключу (см. Вход в систему).
3. На **Dashboard** проверьте статус платформы (Online / Degraded) и KPI: Pending HITL, Anomalies, Agents.

4. Обработайте очередь **HITL** и новые **Anomalies**.
5. При необходимости запустите или проверьте **Workflows**, бюджеты в **Cost**.

> **Интерфейс на английском**
>
> `Подписи кнопок и пунктов меню в продукте сейчас на **английском**. В руководстве рядом с русскими названиями указаны английские лейблы UI, например: **Одобрить** (*Approve*).`

Оглавление

Схема работы оператора

[Диаграмма опущена в PDF — см. HTML-документацию.]

Вход в систему

Адреса

Среда	Frontend	API / Swagger
Локальная разработка	http://localhost:5173	http://localhost:8001/docs
Продакшен	https://agentops.fun	https://agentops.fun (API за тем же доменом)

Страница входа: `/login` (на демо-хосте — `/demo/login`).

Способы входа

1. Email и пароль

1. Откройте вкладку **Email / Password**.
2. Введите email и пароль.
3. Нажмите **Sign In**.
4. После успеха откроется **Dashboard** (`/dashboard`).

Система выдаёт:

- **JWT** — заголовок `Authorization: Bearer …` для пользовательских запросов;
- **Session API key** — заголовок `X-API-Key` для операций в рамках тенанта.

Оба сохраняются в сессии браузера до выхода.

> **Демо / локальный супер-админ**
>
> ```
> В `docker compose` по умолчанию: `admin@aegisai.com` / `admin123`.
> В продакшене эти учётные данные **обязательно** сменить.
> ```

Forgot password? в UI сообщает, что самостоятельный сброс недоступен — обратитесь к администратору (сброс пароля через Admin API / Users).

2. API-ключ

1. Вкладка **API Key**.
2. Вставьте ключ (обычно вида `aegis_…`).
3. Нажмите **Sign In**.

Подходит для программного доступа и ситуаций, когда JWT не используется. Тип сессии: `api_key`.

3. SSO (единый вход)

Если администратор включил IdP (Keycloak, Okta, Google, Microsoft Entra ID):

1. Нажмите кнопку **Sign in with …** в блоке SSO.
2. Пройдите аутентификацию у провайдера.
3. Вернётесь в AgentOps с кодом в URL (`?sso_code=`).

При ошибке смотрите `?sso_error=` и обратитесь к администратору SSO.

Выход

Внизу бокового меню нажмите **Logout**. Сессия и ключи очищаются, открывается `/login` .

Заголовки аутентификации (справка)

Способ	Заголовок	Кто использует
JWT	`Authorization: Bearer <token>`	Пользователи UI
User / Tenant API Key	`X-API-Key: <key>`	UI-сессия, SDK, агенты
Admin Key	`X-Admin-Key: <key>`	Только создание тенантов (Super Admin / ops)

WebSocket уведомлений: `ws://…/api/v1/ws?token=<jwt>` (в проде — `wss://`).

Роли и права доступа

AgentOps использует RBAC с пятью ролями. Права проверяются на API; меню UI скрывает недоступные разделы. Если страница открыта, но действие запрещено — будет **403 Forbidden**.

Иерархия

```
super_admin (5) → admin (4) → security_analyst (3) → developer (2) → viewer (1)
```

Роль	Кто обычно	Зона ответственности
Viewer	Аудитор, наблюдатель	Только обзор и чтение аудита
Developer	Инженер ML / агентов	Агенты, сценарии, чтение политик
Security Analyst	Аналитик SOC	Аудит (экспорт), аномалии, HITL, политики (чтение)
Admin	Админ тенанта	Пользователи, политики CRUD, агенты, HITL
Super Admin	Оператор платформы	Все тенанты, provisioning

Матрица возможностей (смысл для оператора)

Раздел	Viewer	Developer	Security Analyst	Admin	Super Admin
Dashboard	✔	✔	✔	✔	✔
Agents / Workflows / Cost / Marketplace	ограничено*	✔	по UI*	✔	✔
Policies (чтение)	—	✔	✔	✔	✔
Policies (создание/правка)	—	—	—	✔	✔
Audit (чтение)	✔	✔	✔	✔	✔
Audit (экспорт)	—	—	✔	✔	✔
Anomalies / Discovery / Guard ML / HITL	—	—	✔	✔	✔
Users	—	—	—	✔	✔
Settings	✔	✔	✔	✔	✔

*В интерфейсе часть пунктов меню может быть видна шире, чем разрешает API. Источник истины — ответ API (403). При сомнении запросите у Admin повышение роли.

Изоляция тенантов

- Пользователь видит только ресурсы своего `tenant_id`.
- **Super Admin** может работать кросс-тенантно; для UI часто нужно вручную указать tenant API key в **Settings**.
- Назначить роль выше своей нельзя; `super_admin` назначает только другой Super Admin.

Смена роли

После изменения роли администратором **выйдите и войдите снова**, чтобы обновился JWT.

Интерфейс и панель Dashboard

Боковое меню

Типичный порядок пунктов (видимость зависит от роли):

Меню (EN)	Путь	Русское название
Dashboard	`/dashboard`	Панель управления
Agents	`/agents`	Агенты
Workflows	`/workflows`	Рабочие процессы
Cost	`/cost`	Стоимость и бюджеты
Marketplace	`/marketplace`	Маркетплейс
Policies	`/policies`	Политики
Audit	`/audit`	Журнал аудита
Anomalies	`/anomalies`	Аномалии
Prompt Guard ML	`/guard-ml`	Prompt Guard
Discovery	`/discovery`	Обнаружение Shadow AI
HITL	`/hitl`	Согласования
Users	`/users`	Пользователи
Settings	`/settings`	Настройки
Documentation	внешняя ссылка	Документация

В подвале меню: профиль (имя, роль) и **Logout**.

Глобальный поиск в шапке (Search agents, policies, actions…) ведёт к поиску агентов (`/agents?q=…`).

Dashboard — с чего начинать смену

Откройте **Dashboard**. Это командный центр смены.

KPI-карточки

Карточка (EN)	Смысл	Куда кликнуть
Total Agents	Число зарегистрированных агентов	Agents
Active Policies	Активные политики	Policies
Anomalies Detected	Найденные аномалии	Anomalies
Pending HITL	Ожидают согласования	HITL
Prompts Blocked / Guard Latency	Блокировки Prompt Guard	Guard ML / метрики

График Activity (7 days)

Объём audit-событий за неделю. Резкие пики — повод заглянуть в **Audit** и **Anomalies**.

Recent Events

Последние записи аудита: агент, действие, ресурс, риск, время. Обновляются в реальном времени по WebSocket.

Статус платформы

Индикатор **Online** / **Degraded** (из `GET /api/v1/status`): здоровье БД, Redis, OPA, маска API-ключа, uptime.

> **Degraded**
>
> `Не запускайте критичные сценарии «вслепую». См. Типовые проблемы и эскалацию админу инфраструктуры.`

Живые уведомления (toast)

В правом нижнем углу появляются события:

Событие	Что делать
`policy.violated`	Открыть Audit / Policies
`anomaly.detected`	Открыть Anomalies
`hitl.requested`	Открыть HITL
`hitl.resolved`	Информационно
`cost.budget.alert`	Открыть Cost
`system.status`	Проверить статус платформы

Тема оформления

В **Settings** можно переключить Light / Dark. Язык UI сейчас только английский.

Агенты и Discovery

Реестр агентов (`/agents`)

Таблица показывает зарегистрированных AI-агентов тенанта.

Колонка (EN)	Смысл
Name	Имя агента
Type	Тип (`llm`, `tool`, `workflow`, `custom` и др.)
Predicted	Тип, предсказанный ML
Risk	Оценка риска
Status	`active` / `inactive` / `error`
Version	Версия
Last Seen	Последний heartbeat
Actions	View / Delete

Типовые действия оператора

1. Откройте **Agents**.
2. Отфильтруйте или найдите агента (semantic search).
3. Нажмите на строку → **Agent Details** (`/agents/:id`): метаданные, heartbeats, связанные audit и anomalies.
4. Удаление (**Delete**) — обычно Admin+; подтвердите в диалоге.

Регистрация агента

Чаще выполняется через API или SDK (не обязательно из UI):

```
curl -X POST http://localhost:8001/api/v1/agents/register \
  -H "X-API-Key: $TENANT_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "prod-billing-llm",
    "type": "llm",
    "version": "1.0.0",
    "capabilities": ["chat", "summarize"]
  }'
```

Сохраните возвращённый `id` (UUID). Агент должен периодически слать heartbeat:

```
curl -X POST "http://localhost:8001/api/v1/agents/$AGENT_ID/heartbeat" \
  -H "X-API-Key: $TENANT_API_KEY"
```

Рекомендуемый интервал — около **60 секунд**. Если **Last Seen** устарел, статус станет `inactive` / `error` — проверьте ключ и процесс агента.

Практические советы

- Регистрируйте агента **до** включения в workflow.
- Одно стабильное имя на сервис: `prod-billing-llm`, не «test1».
- Указывайте честные capabilities — от них зависят политики и поиск.

Discovery — Shadow AI (`/discovery`)

Раздел для **Security Analyst+**. Сюда попадают кандидаты незарегистрированной AI-активности.

Действие UI	Результат
Confirm	Кандидат подтверждается → создаётся запись агента + fingerprint
Dismiss	Кандидат закрывается как нерелевантный

Регламент

1. Ежедневно просматривайте backlog кандидатов.
2. По evidence решайте: Confirm (легитимный сервис без регистрации) или Dismiss (шум / ложное срабатывание).
3. После Confirm проверьте агента в **Agents** и при необходимости добавьте политики / бюджет.

> **Не игнорируйте backlog**
>
> `Незарегистрированные агенты обходят реестр и усложняют аудит. Рост кандидатов — сигнал для разработки и безопасности.`

Рабочие процессы (Workflows)

Раздел **Workflows** (`/workflows`) — создание и запуск цепочек агентов.

Создание сценария

1. Откройте **Workflows**.
2. Выберите шаблон (Templates: data analysis, documents, monitoring, support) или нажмите **New workflow**.
3. В конструкторе добавьте шаги:
   - выберите агента;
   - задайте input (JSON);
   - при необходимости включите **Require approval** (пауза HITL на шаге).
4. Используйте подстановки:
   - `{{today}}` — текущая дата;
   - `{{step1.output}}` , `{{step2.output}}` — выход предыдущих шагов;
   - `{{input.field}}` — поля входных данных запуска.
5. При необходимости укажите **schedule** (cron), например `0 9 * * *` — ежедневно в 09:00.
6. Нажмите **Save**, затем **Run** / **Execute**.

Редактирование: `/workflows/:id` .

Статусы выполнения

Статус	Значение	Действие оператора
`running`	Идёт выполнение	Наблюдать / при необходимости **Cancel**
`completed`	Успех	Готово
`failed`	Ошибка	Смотреть лог/причину → **Retry** или исправить шаги
`awaiting_approval`	Ждёт HITL	Перейти в **HITL** → **Approve** / **Reject**

Отмена и повтор

- **Cancel** — остановить текущий run.
- **Retry** — повторить после сбоя (если доступно для run).
- История: блок **Recent runs** / API `GET /api/v1/workflows/{id}/runs` .

Триггеры

Тип	Как работает
Ручной запуск	Кнопка Execute в UI или `POST .../execute`
Cron	Поле `schedule` в определении workflow
Webhook	`POST /api/v1/workflows/{id}/webhook` (с проверками SSRF/секрета)

Безопасность и бюджеты

- Политики применяются **fail-closed**: при недоступности OPA/критичном отказе сценарий не «проскакивает» без проверки.
- Мягкий/жёсткий бюджет в **Cost** может остановить следующие шаги.
- Шаг с **Require approval** переводит run в `awaiting_approval` до решения в HITL.

Пример через API

```
curl -X POST http://localhost:8001/api/v1/workflows \
  -H "Authorization: Bearer $JWT" \
  -H "X-API-Key: $TENANT_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Daily Sales Report",
    "steps": [
      {"agent": "sales-fetcher", "input": {"date": "{{today}}"}},
      {"agent": "report-generator", "input": {"data": "{{step1.output}}"}},
      {"agent": "email-sender", "input": {"report": "{{step2.output}}", "to":
"ceo@company.com"}}
    ],
    "schedule": "0 9 * * *"
  }'
```

Подробнее (EN): Workflows.

Стоимость и бюджеты (Cost)

Раздел **Cost** (`/cost`) показывает расход по агентам и позволяет задать лимиты.

Что учитывается

На шагах workflow и вызовах агентов:

- токены входа / выхода (LLM);
- задержка (latency, ms);
- внешние API-вызовы;
- оценка в USD (по настроенным тарифам окружения).

Работа в UI

1. Откройте **Cost**.
2. Просмотрите таблицу **Cost by agent** (spend, requests, latency).
3. Если данных нет — сначала выполните workflow (No cost data yet — run a workflow first).
4. Установите бюджет:
   - выберите агента;
   - укажите лимит в USD;
   - нажмите **Set budget**.

Баннер предупредит, если расход близко к лимиту или лимит **EXCEEDED**.

Алерты

- WebSocket: `cost.budget.alert`.
- При жёстком превышении следующие шаги сценария могут остановиться.

Через API

```
# Расход агента
curl "http://localhost:8001/api/v1/cost/agent/$AGENT_ID?period=last_7_days" \
  -H "X-API-Key: $TENANT_API_KEY"

# Бюджет
curl -X POST http://localhost:8001/api/v1/cost/budget \
  -H "Authorization: Bearer $JWT" \
  -H "X-API-Key: $TENANT_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "agent_id": "'"$AGENT_ID"'",
    "limit_usd": 100,
    "period": "monthly",
    "alert_threshold": 0.8
  }'
```

`alert_threshold: 0.8` — предупреждение при 80% бюджета.

Регламент оператора

- Ежедневно: агенты near limit / EXCEEDED.
- Еженедельно: пересмотр лимитов с владельцами агентов.
- При внезапном росте spend — корреляция с **Anomalies** и **Audit** (возможный цикл или атака).

Подробнее (EN): Cost.

Маркетплейс агентов

Раздел **Marketplace** (`/marketplace`) — каталог пакетов агентов для публикации и установки внутри (и при публичных пакетах — между) тенантами.

Публикация

1. Убедитесь, что агент зарегистрирован в **Agents**.
2. Откройте **Marketplace → Publish agent**.
3. Укажите метаданные, версию, visibility: **public** или **private**.
4. Сохраните. Private-пакеты других тенантов вам не видны.

Право публикации ограничено ролью/функцией `canPublish()` в UI.

Установка

1. Найдите пакет в каталоге.
2. Нажмите **Install**.
3. В тенанте появится локальный агент по манифесту пакета (с санитизацией URL / SSRF-защитой).

Отзывы

После установки можно оставить review/rating через UI или API `POST /api/v1/marketplace/packages/{id}/reviews`.

Когда использовать

Задача	Действие
Переиспользовать готовый агент команды	Install из каталога
Отдать агент другим командам	Publish (public/private)
Быстрый старт без сборки с нуля	Templates workflows + marketplace package

Подробнее (EN): Marketplace.

Политики безопасности

Раздел **Policies** (`/policies`) управляет правилами доступа агентов (OPA/Rego + шаблоны соответствия).

> **Минимальные роли**
>
> `Чтение — Developer / Security Analyst+. Создание и изменение — **Admin+**.`

Список политик

Колонки: Name, Severity (`low` … `critical`), Status (Active), Default, Actions (Edit / Delete).

Создание политики

1. Нажмите **Create Policy**.
2. Выберите шаблон из каталога (`GET /api/v1/policies/templates`):

Шаблон	Типичная severity	Назначение
`gdpr`	Critical	Персональные данные (EU)
`hipaa`	Critical	Медицинские данные (US)
`fz152`	Critical	Персональные данные (РФ, ФЗ-152)
`hitl`	High	Требование ручного согласования
`custom`	Medium	Свои правила

1. Заполните Name, Description, Severity, Rules (JSON/Rego в редакторе).
2. Включите **Active** (и при необходимости **Default**).
3. **Save**. Редактирование: `/policies/:id` .

> **Юридическая оговорка**
>
> `Шаблоны GDPR / HIPAA / ФЗ-152 — стартовая база для инженерии контроля, а не юридическая сертификация. Финальную валидацию делает комплаенс вашей организации.`

Проверка действия (Policy Check)

Перед опасной операцией агент/интеграция вызывает:

```
curl -X POST http://localhost:8001/api/v1/policies/check \
  -H "X-API-Key: $TENANT_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "agent_id": "'"$AGENT_ID"'",
    "action": "export_data",
    "resource": "customers",
    "context": {}
  }'
```

В ответе смотрите: `allowed`, `reason`, `severity`, `required_approval`, `risk_score`, признаки anomaly.

- `allowed=false` → событие `policy.violated`, запись в Audit.
- `required_approval=true` → путь через **HITL**.

Операторский цикл

1. Получили toast / запись о нарушении.
2. Найдите событие в **Audit**.
3. Откройте соответствующую политику в **Policies**.
4. Решите: ошибка контекста агента, нужна HITL, или политика слишком жёсткая/мягкая (меняет Admin).
5. Зафиксируйте вывод для команды.

Подробнее (EN): Policies.

Журнал аудита (Audit)

Раздел **Audit** (`/audit`) — неизменяемый журнал действий агентов (при включённой подписи — HMAC).

Просмотр

Колонки: Time (UTC), Agent, Action, Resource, Risk, Approved, Signature (если signing включён).

Фильтры

- Agent ID
- Action
- Resource
- Диапазон дат (From / To)
- Порог риска

Нажмите **Apply Filters**.

Экспорт

Кнопка / способ	Кто	Когда
Export CSV / **JSON** / **JSONL**	Security Analyst+	Быстрый выгруз до лимита
Async Export	Security Analyst+	Большие объёмы: job → статус → скачивание

Viewer / Developer обычно только читают журнал без экспорта.

Зачем оператору

- Расследование нарушений политик и аномалий.
- Доказательная база для комплаенса.
- Корреляция с HITL (Approved) и risk score.

Типовой разбор

1. Из Dashboard / Anomalies возьмите `agent_id` и время.
2. В Audit отфильтруйте агента и окно времени.
3. Просмотрите цепочку Action → Resource → Risk → Allowed/Denied.
4. При необходимости экспортируйте выборку для тикета/SIEM.

Подробнее (EN): Audit.

Аномалии и Prompt Guard

Аномалии (`/anomalies`)

Доступ: **Security Analyst+**.

Таблица: Timestamp, Agent, Risk, Source (`auto` / `manual`), Status, Details. Клик по строке раскрывает подробности.

Разбор аномалии

1. Отсортируйте по risk (сначала высокие).
2. Откройте детали: что отклонилось от базового поведения.
3. Сверьте с **Audit** того же агента и периода.
4. Вынесите вердикт:
   - **True Positive (TP)** — реальное отклонение / угроза → подтвердите в UI, ужесточите политику, заблокируйте действие через HITL/Admin;
   - **False Positive (FP)** — ложное срабатывание → отметьте FP (feedback), чтобы модель дообучалась.

Feedback через API:

```
curl -X POST http://localhost:8001/api/v1/anomalies/detector/feedback \
  -H "Authorization: Bearer $JWT" \
  -H "Content-Type: application/json" \
  -d '{"anomaly_id": "...", "label": 0}'
```

`label: 0` — FP, `1` — TP. После накопления событий возможен auto-retrain.

Detector Status

На странице — версия модели, backend (`sklearn` / `onnx` / …), samples, contamination, last trained.
Обучение / rollback версий — **Admin+** (`POST /detector/train`, rollback).

Если агент «сирота» (нет в реестре) — смотрите **Discovery**.

Prompt Guard ML (`/guard-ml`)

При `ENABLE_GUARD=true` перед вызовами LLM проверяются injection / jailbreak / утечки.

На **Dashboard** видны:

- Prompts Blocked
- Guard Latency
- Top attack patterns

Оператор:

1. Следит за ростом блокировок и паттернами атак.
2. При всплеске — Audit + Anomalies по затронутым агентам.
3. Сообщает разработчикам, если легитимные промпты массово блокируются (нужна настройка Guard).

Подробнее (EN): Anomalies, Prompt Guard.

HITL — согласование человеком

HITL (Human-in-the-Loop) — очередь ручных решений по рискованным действиям агентов и шагам workflow.
Раздел: `/hitl`. Роль: **Security Analyst+**.

Когда появляется запрос

- Политика вернула `required_approval`.
- Высокий risk score / критичные операции (`delete_data`, `export_phi` и т.п.).
- В шаге workflow включено **Require approval**.

Статус run workflow при этом: `awaiting_approval`.

Работа с очередью

1. Toast `hitl.requested` или KPI **Pending HITL** на Dashboard.
2. Откройте **HITL**.
3. Изучите: Agent, Action, Resource, Reason, контекст.
4. Решение:
   - **Approve** — действие/шаг разрешён, агент продолжает (поллинг статуса);
   - **Reject** — отклонено; **обязательно** укажите причину (reason).

Поле UI	Смысл
Requested	Время запроса
Status	`pending` / `approved` / `rejected`
Approver	Кто принял решение (имя можно задать в Settings)

Правила хорошей практики

- Не копите pending: определите SLA (например, 15–60 минут в рабочее время).
- Reject без комментария недопустим для расследований.
- Критичные ресурсы (PII, PHI, массовое удаление) — эскалация Admin / владельцу данных.
- В **демо-режиме** согласования могут быть отключены (read-only).

API (справка)

```
# Очередь
curl http://localhost:8001/api/v1/hitl/pending -H "Authorization: Bearer $JWT"

# Одобрить
curl -X POST http://localhost:8001/api/v1/hitl/approve/$ID \
  -H "Authorization: Bearer $JWT" \
  -H "Content-Type: application/json" \
  -d '{"approver": "ivan.petrov"}'

# Отклонить
curl -X POST http://localhost:8001/api/v1/hitl/reject/$ID \
  -H "Authorization: Bearer $JWT" \
  -H "Content-Type: application/json" \
  -d '{"approver": "ivan.petrov", "reason": "Недостаточно оснований для экспорта"}'
```

Подробнее (EN): HITL.

Пользователи и настройки

Пользователи (`/users`) — Admin+

Управление учётными записями тенанта:

- создание пользователя;
- смена роли;
- блокировка / разблокировка;
- сброс пароля;
- удаление.

Рекомендации

- Принцип наименьших привилегий.
- Разделяйте обязанности: Developer ведёт агентов, Analyst — HITL/аномалии.
- Блокируйте уволенных и «забытые» учётки.
- Роль `super_admin` — только платформенным операторам.

SSO-пользователи могут не иметь локального пароля (JIT-провижининг).

Подробнее (EN): Users, Roles.

Настройки (`/settings`) — все роли

Блок	Действие оператора
API Key	Вставить tenant/session key → **Save**; проверить **Test Connection**
Theme	Light / Dark
Account	Email, имя, роль, tenant, last login (`/auth/me`)
HITL Approver Name	Имя, подставляемое при Approve
Password	Смена своего пароля
Profile	Обновление профиля
User API keys	Создать / отозвать персональные ключи (**сырой ключ показывается один раз**)
Sign Out	Выход

Если Dashboard пустой / нет данных

Сообщение вроде API key is not configured. Re-login or set key in Settings.

1. Выйдите и войдите снова, **или**

2. Settings → вставьте корректный **X-API-Key** тенанта → Save → Test Connection.

Super Admin без привязанного тенанта часто должен вручную указать ключ нужного тенанта в Settings.

Ежедневная и еженедельная работа

Ежедневно — Security Analyst / дежурный оператор

1. **Вход** → проверить Dashboard: статус **Online**.
2. **Pending HITL** — разобрать очередь (Approve/Reject + reason).
3. **Anomalies** — высокий risk → Audit → FP/TP feedback.
4. **Discovery** — Confirm / Dismiss кандидатов Shadow AI.
5. **Recent Events** и пики Activity — нет ли массовых `policy.violated`.
6. **Cost** — near limit / EXCEEDED.
7. **Agents** — статусы `inactive` / `error` (пропал heartbeat).

Ежедневно — Developer

1. Статус своих агентов и регистрация новых.
2. **Workflows**: failed runs → Retry / правка шагов; согласовать HITL с Analyst.
3. Контроль spend по своим агентам в **Cost**.
4. Publish/Install в Marketplace по необходимости.
5. Перед новыми опасными actions — `policies/check`.

Еженедельно — Admin / Analyst

1. Экспорт **Audit** (CSV/JSON/async) в архив / SIEM.
2. Ревизия Active Policies (severity, default, актуальность ФЗ-152/GDPR/HIPAA).
3. Пользователи: роли, блокировки, отзыв API-ключей.
4. Detector Status / retrain при шуме аномалий.
5. Пересмотр бюджетов и `alert_threshold`.
6. Метрики Shadow AI и Guard (всплески атак).

Ежеквартально

- Аудит ролей (least privilege).
- Ограничить число `super_admin`.
- Ротация seed-паролей и критичных ключей (совместно с платформенным ops).

Реагирование на инциденты

Краткие playbook’и для типовых ситуаций. Цель — стабильный порядок действий без импровизации.

A. Нарушение политики (`policy.violated`)

1. Зафиксируйте toast / запись в **Audit** (agent, action, resource, risk).
2. Откройте **Policies** — какая политика и severity.
3. При `required_approval` — обработайте **HITL**.
4. Если в ответе check есть anomaly — откройте **Anomalies**.
5. Решение:
   - исправить контекст агента (consent, lawful basis и т.д.);
   - ужесточить/ослабить политику (**Admin**);
   - временно остановить агента / workflow.
6. Сохраните вывод для тикета (экспорт Audit при необходимости).

B. Очередь HITL / критичное согласование

1. **HITL** → сортировка pending по критичности ресурса.
2. Проверьте action, records_affected, причину запроса.
3. **Approve** или **Reject** + reason.
4. Убедитесь, что workflow вышел из `awaiting_approval`.
5. При перегрузе очереди — эскалация Admin / второй дежурный.

C. Аномалия / подозрение на атаку

1. **Anomalies** → высокий risk → детали.
2. Корреляция в **Audit** по `agent_id` и времени.
3. TP → политика / блок / HITL на действие; FP → feedback `label: 0`.
4. Orphan-агент → **Discovery** Confirm.
5. При деградации детектора — Admin (retrain / rollback), rule fallback.

D. Превышение бюджета

1. **Cost**: баннер EXCEEDED / toast `cost.budget.alert`.
2. Найдите агента с аномальным spend.
3. **Cancel** активные workflow при необходимости.
4. Поднимите `limit_usd` (осознанно) или остановите источник нагрузки.
5. Проверьте, нет ли цикла вызовов / инцидента безопасности.

E. Платформа Degraded

1. Индикатор статуса на Dashboard.
2. Сообщите админу инфраструктуры; проверьте health:

```
curl https://<host>/api/v1/health
curl https://<host>/api/v1/health/ready
```

1. OPA down → workflow fail-closed; не считайте «тишину» успехом.
2. Нет API key → Settings / повторный логин.

F. Shadow AI

1. **Discovery** → evidence кандидата.
2. **Confirm** (зарегистрировать) или **Dismiss**.
3. После Confirm — политики, бюджет, владелец сервиса.

Типовые проблемы оператора

Симптом	Что проверить / сделать
Не удаётся войти	Email/пароль; пользователь не заблокирован; SSO error в URL; обратиться к Admin
Пустой Dashboard, нет метрик	Re-login или **Settings → API Key → Save → Test Connection**; Super Admin — ключ тенанта вручную
403 Forbidden	Роль не позволяет раздел/действие — запросить Admin
Агент `inactive`	Heartbeat ~60 с; верный API key агента; процесс жив
Политика всегда deny	Контекст запроса; политика Active; здоровье OPA; Admin
Workflow `awaiting_approval`	Очередь **HITL**
Workflow failed / стоп по cost	**Cost** бюджет; **Cancel** / **Retry**; fail-closed OPA
Шумные аномалии	FP feedback; Admin retrain/rollback
Долгий экспорт Audit	**Async Export** → дождаться completed → download
Forgot password	Только Admin (сброс); в UI самообслуживания нет
Демо-режим	HITL/часть настроек read-only — не путать с продом

Быстрая проверка API

```
curl -s http://localhost:8001/api/v1/health
curl -s http://localhost:8001/api/v1/health/ready
curl -s http://localhost:8001/api/v1/status -H "X-API-Key: $TENANT_API_KEY"
```

Инфраструктурный troubleshooting: Deployment Troubleshooting.

Глоссарий EN ↔ RU

В интерфейсе (EN)	В руководстве (RU)
Dashboard	Панель управления
Agents	Агенты
Workflows	Рабочие процессы / сценарии
Cost / Set budget	Стоимость / Установить бюджет
Marketplace / Publish / Install	Маркетплейс / Опубликовать / Установить
Policies / Create Policy	Политики / Создать политику
Audit / Export	Журнал аудита / Экспорт
Anomalies	Аномалии
Prompt Guard ML	Защита промптов (ML)
Discovery / Confirm / Dismiss	Обнаружение Shadow AI / Подтвердить / Отклонить
HITL / Approve / Reject	Согласование человеком / Одобрить / Отклонить
Users	Пользователи
Settings	Настройки
Sign In / Logout	Войти / Выйти
Execute / Run / Cancel / Retry	Запустить / Отменить / Повторить
Require approval	Требовать согласование
Pending	Ожидает
Online / Degraded	В норме / Деградация
Tenant	Тенант (организация / изолированная среда)
Shadow AI	Незарегистрированная AI-активность
Fail-closed	Отказ по умолчанию при сбое проверки
Heartbeat	Сигнал «агент жив»
RBAC	Ролевая модель доступа

Полезные ссылки

- Продакшен: agentops.fun
- Документация: agentops.fun/documentation
- User Guide (EN): User Guide
- Admin Guide (EN): Admin Guide

- Репозиторий: valera7623/AgentOps