Агенты

Агенты мониторинга

Меню: **Инфраструктура → Агенты мониторинга**.
Маршруты: `/probes` (вкладки **Агенты** | **Обновление ПО**), установка — `/probes/install`.
Legacy `/agents` → редирект на **Обновление ПО** (`/probes/updates`); `/agents/install` → `/probes/install`.
Для новой установки открывайте **Скачать агент**, не `/agents`.

wiSLA vs wiProbe

	wiSLA	**wiProbe**
Что это	Текущий Go-агент (`wisla-wiprobe-agent`)	Программный агент предыдущей системы (SLAMON autoreg)
Установка	Визард `/probes/install`	Авторег с менеджера; не из каталога пакетов wiSLA
Remote update	Да (каталог пакетов)	**Недоступно**
Каналы / availability jobs	Да	Нет

Версию пакета берите из `/agents/index.json` на вашем менеджере (не из устаревших примеров в docs).

Токен (auth-required)

На native-стенде обычно **`wisla.agent.auth-required=true`**. Без Bearer-токена в `/etc/wisla/agent.yaml` агент может зарегистрироваться, но heartbeat даёт **401** → в UI **«Не в сети»**.

- Same-host после `.run`: локальный агент
- Remote: UI **Обслуживание → Связь с менеджером** (агент ≥ 0.2.11) или `sudo wisla-wiprobe-agent config`
- За proxy задайте `wisla.agent.manager-public-url` — reverse-proxy

Агенты (список)

- **Связь** — в сети / не в сети по TTL heartbeat (время последнего контакта). Heartbeat только отмечает «жив»; тяжёлая работа (LLD, каталог items) не должна блокировать этот запрос.

- **В менеджере** — включён / выключен (флаг `enabled`: участвует ли агент в мониторинге). Это не remote start/stop ОС-сервиса.
- При выборе агентов: **Остановить мониторинг** / **Восстановить мониторинг** — отдельные кнопки; **Массовые действия** — автообновление и протокол HTTP/HTTPS; **Удалить** открывает модалку с именем агента (или списком имён). То же подтверждение — кнопка **Удалить агент** на карточке. Запись пропадает из списка, привязки у сервисов сбрасываются; это необратимо.
- Привязка к узлам, версии агента; колонка **ОС** (Linux / Windows / macOS) отдельно от **Тип агента** (**wiSLA** — текущий Go-агент, **wiProbe** — программный агент предыдущей системы).
- Программный **wiProbe** появляется в том же списке после авторега. В SLAMON в `wisla-url` / `autoreg-url` достаточно `http://<менеджер>:8088`. Первый контакт часто **пустой POST** с заголовком `pb_r`; менеджер отвечает 204 и ключом `pb_k`, затем агент шлёт паспорт. После смены кода **перезапустить backend**.
- Полный авторег создаёт строку; повтор с UUID обновляет паспорт; пустые POST с того же IP только «жив». На карточке агента интерфейсы, маршруты и диски wiProbe — во вкладках **Инвентаризация хоста** (как у wiSLA), не в боковой колонке.
- **Теги** — ваши метки для фильтрации и группировки (`prod`, `dc1`, площадка и т.д.). Редактируются на карточке агента (карандаш у поля «Теги») и видны отдельным столбцом в списке. Поиск по списку учитывает теги.
- **Роли** (`host`, `channel`) — **только чтение**: что агент умеет на этом хосте (метрики узла / канальные пробы). Задаётся в `agent.yaml` (`roles:`), в UI не редактируется напрямую. Имена `host` и `channel` нельзя использовать как пользовательские теги. Пакет ≥ **0.2.17** ставит по умолчанию **`host` + `channel`** (echo ленивый — без channel jobs нагрузка как у host-only). Для старых агентов без `channel`: `POST /api/v1/agents/{id}/maintenance/identity` с `roles: ["host","channel"]` или скрипт alpha-channel-enable-roles.py.
- На диске **один** yaml: `/etc/wisla/agent.yaml` (+ `agent.meta.json` для групп). Что снимать (CPU, Docker, ICMP канала) приходит с менеджера в heartbeat (`host_items` / `channel_jobs`). Старый `/etc/wisla-wiprobe-agent/agent.yaml` — leftover до 0.2: агент при старте/config переносит в `/etc/wisla` и **удаляет** дубликат (разные `server:` / порты в двух файлах больше не нужны).
- В блоке **Состояние хоста** — **Синхронизация NTP** (статус, смещение, источник). Важно для агентов с ролью `channel`; данные приходят с метрикой `system.ntp.get` после обновления агента.
- **Расположение** на карточке зонда — одно поле с автодополнением (город / улица / дом), для карты. Этот адрес используется на карте сервисов и на виджете **Геокарта площадок** дашборда. Без адреса агент на карту не попадает.
- **Название** по умолчанию совпадает с **hostname** хоста. Переименование — карандаш → **Сохранить**. Агент ≥ **0.2.11** сам пишет новое имя в `agent.yaml` на heartbeat; иначе — вкладка **Обслуживание → Связь с менеджером** или `wisla-wiprobe-agent config`. Менеджер помнит старое имя (`previous_agent_name`), дубликат не плодится.
- На **Обслуживание**: remote-правка Manager URL / групп / token / advertise IP (тот же канал, что логи и reboot).
- **IP-адрес** в списке — адрес, который агент сообщил менеджеру (сеть хоста), а не «откуда ставили» пакет. Для **wiProbe** берётся из интерфейсов авторега (не loopback): у Windows часто `ip=127.0.0.1` в заголовке, в списке будет Ethernet. В инвентаре интерфейсов wiSLA-агента IP берётся из `net.if.*` и может временно быть пустым («Не назначен»).

- В блоке **Данные, полученные агентом**: вкладки **Система**, **CPU**, **ОЗУ**, интерфейсы, маршруты, диски. У **wiProbe** те же вкладки (без CPU/ОЗУ — их нет в автореге), данные из паспорта авторега, не `net.if.get`. У wiSLA CPU/ОЗУ опрашиваются каждые **5 мин**; интерфейсы — раз в **час**; диски — раз в **12 ч**. На вкладке **CPU** длинное имя модели обрезается в таблице: наведите курсор для подсказки или нажмите на модель — откроется окно с полным названием.
- **Эталон**: после первого полного набора система запоминает паспорт. Кнопка **Сравнить с эталоном** показывает, что изменилось; раз в сутки (настраивается) при отличии появляется событие «Изменение конфигурации хоста».
- Если есть устаревшие или ожидающие обновления **wiSLA**-агенты — на вкладке **Обновление ПО** показывается счётчик. Программные **wiProbe** там со статусом **Недоступно**: каталог пакетов их не обновляет.

Матрица SLAMon / wiProbe / каналы / availability: SLAMon ≥ 1.14.10. Каналы и доступность услуги с агента исполняет только **wiSLA** (`wisla-wiprobe-agent`). Classic SLAMon ≥ 1.14.10 — в списке и в паспорте; jobs на него не выдаются. Железо WPE/Berkut — отдельная волна.

Автоонбординг при регистрации агента настраивается в **Обнаружение** (`/discovery`, тип **Агенты мониторинга**; deep-link с списка агентов: **Настроить автоонбординг**). Описание: Discovery — агенты (UX). Backend API `/api/v1/agents/registration-rules`.

Обслуживание агента (карточка зонда)

Рядом с заголовком **Паспорт агента** — вкладки **Паспорт** | **Обслуживание** (Material-style underline, как у инвентаризации). На **Паспорт** — две карточки (описание и состояние хоста). На **Обслуживание** — автообновление, протокол загрузки, уровень логов, архив, перезапуск. Для логов и перезапуска нужны агент ≥ **0.2.0** и статус **в сети**:

- **Автообновление** — переключатель на стороне менеджера: при выключении агент не получает директиву на установку новой версии (даже если на вкладке **Обновление ПО** задана целевая версия).
- **Протокол обновления** — **HTTP** или **HTTPS** для скачивания пакета с менеджера (по умолчанию HTTPS). Базовые URL — **Настройки → Мониторинг → Обновления агентов**; при первом запуске backend подставляет адрес сервера автоматически. Переопределение: `wisla.agent.manager-public-url` в `wisla.properties` (см. Настройки).
- **Уровень логирования** — ERROR … TRACE; команда уходит на heartbeat, уровень сохраняется в `agent.yaml`.
- **Скачать логи** — агент собирает zip с хоста и загружает на менеджер; скачивание идёт через API с авторизацией (без ухода на белый экран).
- **Перезагрузить агент** — рестарт **сервиса агента**, не ОС (краткий offline до heartbeat).

Логи на хосте: `/var/log/wisla-agent/` (Windows: `ProgramData\wisla-agent\log`). Файлы `{level}-{дата}.log`, раз в сутки архив `logs-{дата}.zip`, общий лимит **300 МБ**. На Windows нужен агент ≥ **0.2.2** (исправлены пустые log-файлы и ACL каталога).

Установка

Визард пакетов DEB / RPM / Windows MSI / **macOS** с понятными подписями (**x86-64 / ARM64**, **64-bit / 32-bit**, **Intel / Apple Silicon**) и командами установки под выбранную платформу. Кнопка **Копировать** работает и при открытии UI по HTTP (не только HTTPS).

На шаге выбора пакета есть блок **«Браузер для веб-сценариев»** — addon `wisla-browser-runtime`. Для **первого выпуска** не считайте Chromium web-scenario прод-готовым (stub / следующая волна) — ограничения · режим Браузер.

В шаге «Адрес сервера» нужен URL менеджера, обычно `http://MANAGER:8088`. Если указать только IP, установщик и агент сами добавят `http://` и порт **8088**. Без схемы регистрация не уходит (`unsupported protocol scheme`). Проверка на хосте: `grep '^server:' /etc/wisla/agent.yaml` и `journalctl -u wisla-wiprobe-agent -e`.

Windows: команду нужно выполнять **от имени администратора**. В визарде три строки: скачать MSI ⟶ `msiexec /i … SERVER=… GROUPS=…` — пакет сам вызывает `setup` (конфиг + служба). Удаление — `msiexec /x`. Повторная установка не падает, если служба уже есть или `agent.meta.json` с BOM.

Linux: скопированная команда пишет `/etc/wisla/install.env` и ставит выбранный DEB или RPM (три строки, как Windows). Так `WISLA_MANAGER` доходит до postinstall даже при жёстком sudo. Удаление: `apt-get remove` или `rpm -e`. Нужны `curl` или `wget`.

Апгрейд с **очень старого** пакета: unit мог остаться в `/etc/systemd/system/wisla-wiprobe-agent.service` с `-config /etc/wisla-wiprobe-agent/agent.yaml`. Бинарь обновляется, а `server:` — нет (self-update не задаёт `WISLA_MANAGER`). В журнале будет `registered with server`, но `server=http://СТАРЫЙ:8088` — на новом менеджере агента нет. Проверка:

```
systemctl cat wisla-wiprobe-agent
grep -n '^server:' /etc/wisla/agent.yaml
# если ещё есть — удалить дубликат после сверки:
# sudo rm -f /etc/wisla-wiprobe-agent/agent.yaml
```

Перенацелить на текущий менеджер — **только** канонический файл (подставьте свой URL, не lab IP):

```
sudo sed -i 's|http://OLD_MANAGER:8088|http://NEW_MANAGER:8088|' /etc/wisla/agent.yaml
sudo rm -f /etc/wisla-wiprobe-agent/agent.yaml /etc/wisla-wiprobe-agent/agent.meta.json
sudo systemctl daemon-reload
sudo systemctl restart wisla-wiprobe-agent
```

Или без правки yaml: `sudo wisla-wiprobe-agent config` / UI **Обслуживание ⟶ Связь с менеджером**.

Если `preferred port 10050 busy` — агент слушает **10051** (часто занят старым wiSLA Agent). После регистрации на нужном менеджере `access_point` должен быть `IP:10051`. Удаление строки в UI не останавливает сервис на хосте: сначала `systemctl stop/disable`, иначе агент снова зарегистрируется.

macOS: удаление — `sudo bash /usr/local/wisla/wisla-wiprobe-agent/uninstall.sh` (скрипт ставится вместе с агентом).

После установки на хосте:

```
wisla-wiprobe-agent help        # команды
wisla-wiprobe-agent -v          # версия
```

```
wisla-wiprobe-agent status       # локальный статус
sudo wisla-wiprobe-agent config  # перенастройка без правки yaml (или UI Обслуживание ≥ 0.2.11)
```

На macOS бинарь **без подписи Apple** (lab): при блокировке Gatekeeper разрешите запуск в **Privacy & Security**. Windows MSI тоже unsigned — SmartScreen может спросить «Run anyway». Prod-подпись — ещё не в CI (политика).

Обновление ПО

- Загрузка пакетов администратором на вкладке **Обновление ПО**.
- В таблице: имя (клик ⟶ карточка агента), **IP**, ОС, текущая/целевая версия, статус (на узких экранах вторичные колонки скрываются). Порядок строк — **по имени**, не по последнему heartbeat.
- **Одновременно** — сколько агентов качают пакет параллельно (очередь / concurrency); значение сохраняется и для следующих волн.
- **Только в окне** — опциональный суточный интервал (HH:mm, часовой пояс инстанса): вне окна обновления ставятся в очередь и стартуют, когда окно откроется.
- Фильтр статусов: **Все** / **Устарели** / **Ожидают** / **Актуальны**.
- Кнопка «Обновить устаревшие» назначает целевую версию (статус **Ожидает** до heartbeat; вне окна — отложенный старт). **Обновить список** только перезагружает таблицу.
- Реальная установка: агент на **следующем heartbeat** (и при открытом окне, если оно включено) получает `update` (url + sha256), скачивает пакет, проверяет хеш, ставит и перезапускает сервис. После успешного апгрейда версия совпадает ⟶ статус снова **Ок**.
- Транспорт пакета:
  - manager `https://...` ⟶ пакет тоже только HTTPS;
  - manager `http://...` (лаба) ⟶ HTTP с того же manager **разрешён** (с 0.1.7), SHA-256 обязателен;
  - принудительно: `WISLA_UPDATE_ALLOW_INSECURE_HTTP=1`.
- Агент должен быть **онлайн** (heartbeat). Офлайн так и останется в «Ожидает».
- Нужен пакет в каталоге под OS/arch агента (`deb` / `rpm` / `msi` / `tar`) с sha256 в `index.json`.
- С агента **0.1.8+**: если установка сорвалась (или нет пакета/sha256 в каталоге), рядом со статусом **Ожидает** показывается текст ошибки (`last_update_error`). После успешного обновления ошибка очищается.

Старые URL `/agents` и `/agents/install` редиректят на новые.

Технические детали: Зонды / агенты UI, Agents API, Установка, Агент.
