Шаблоны и автоматизация

- Шаблоны
- Конструктор шаблонов узлов
- Шаблоны здоровья
- Воркфлоу уведомлений
- Обслуживание узла
- Входящие trap / syslog

Шаблоны

Шаблоны

Маршрут: `/templates` — вкладки **Узлы** | **Каналы** | **Доступность услуги** | **Веб-сценарий (HTTP)** | **Браузер** сразу под заголовком (underline, как на Агентах мониторинга).

Шаблоны узлов

- Каталог; в topbar — **Создать** / **Импорт**; при выборе строк — **Редактор** / **Экспорт** / **Удалить** у таблицы.
- **Клик по строке** — модалка с карточкой шаблона: RU-название, техническое имя, блок **Описание** (зачем шаблон, что измеряет, как настроить), сводка сущностей, источники сбора (Agent / SNMP / JMX / IPMI / Server), связи, теги; из неё можно открыть редактор. В списке — RU-имя и превью первой строки описания.
- Как пользоваться редактором шаблонов узлов — пошагово для операторов.
- Привязка к узлам инвентаря; источники совпадают с шагом «источники сбора» в мастере узла.
- Что унаследовалось на узел: вкладка **Конфигурация** на карточке — эффективная конфигурация. Правка шаблона применяется фоном, не мгновенно на всех узлах.

Шаблоны каналов

- Профили **качества пути** между двумя зондами. DNS/HTTP/HTTPS/SIP — в доступности услуги, не в канале.
- Пороги warn/crit правятся в форме шаблона и на обзоре канала (вкладка **Пороги**) — это **шаблон**, не один канал. Сводка: Пороги.
- В topbar — **Создать** / **Импорт**; при выборе строк — **Изменить** / **Удалить** у таблицы.
- **Клик по строке** — карточка с **развёрнутым описанием** (зачем, когда ставить, что измеряет, требования). В списке — только первая строка; откройте карточку, чтобы прочитать полностью.
- Используются при создании channel-сервисов.

Шаблон	Зачем (кратко)
Качество ICMP (P-Test)	Жива ли линия: потери, задержка, джиттер. Базовый SLA.
Качество UDP (U-Test)	Голос/видео и MOS, не только ping.
Задержка TCP (L4)	Как быстро поднимается TCP — тормозят приложения.
Traceroute / RFC / MTU	Маршрут, ёмкость, Path MTU. TWAMP и прочие stub-типы скрыты из каталога оператора (legacy / будущий hardware-зонд) — см. Профили каналов.

Шаблоны доступности услуги

- Маршрут: `/templates/availability`
- Профили DNS / HTTP / HTTPS / SIP для сервисов **Доступность услуги** (один зонд → цель)
- Доступность услуги

Шаблоны веб-сценария (HTTP)

- Маршрут: `/templates/web-scenario`
- Движок шагов: **HTTP** (open, request, login, assert_status, ssl_cert и др.)
- Seeds: `ws-tpl-public-keyword`, `ws-tpl-login-assert` (HTML-form POST), `ws-tpl-login-api` (REST JSON POST — для SPA/REST; макросы `{$WS.LOGIN.URL}`, `{$WS.LOGIN.BODY}`, `{$WS.JSON.PATH}`); пароль без default
- Веб-сценарий

Шаблоны браузерных сценариев

- Маршрут: `/templates/web-scenario-browser`
- Движок шагов: **browser** (Headless Chromium на зонде: goto, click, fill, assert_text)
- Отдельный каталог — не смешивается с HTTP-шаблонами; к сервису подключается только при `web_scenario_engine=browser`
- Seeds: `ws-tpl-browser-page`, `ws-tpl-browser-login`, `ws-tpl-browser-blank`
- Браузерный веб-сценарий

Редиректы: `/node-templates` → узлы, `/channel-templates` и `/sla` → каналы, `/web-scenario-templates` → веб-сценарий (HTTP), `/web-scenario-browser-templates` → браузер.

Технические детали: Шаблоны узлов, Шаблоны каналов.

Конструктор шаблонов узлов

Редактор шаблонов узлов

Маршрут: **Инфраструктура → Шаблоны узлов** (`/templates/nodes`, редирект с `/node-templates`).

Главное

Шаблон узла — это набор метрик (как шаблон wiSLA). Он **не мониторит сам по себе**: его нужно **привязать к узлу** в карточке сервиса.

В каталоге уже есть **готовые шаблоны** (Linux, Windows, ICMP Ping, PostgreSQL ODBC, Docker и др.). В большинстве случаев **редактор не нужен** — достаточно выбрать шаблон при настройке узла.

Когда открывать редактор

Ситуация	Что делать
Мониторинг типового Linux/Windows/БД	Взять готовый шаблон из каталога, привязать к узлу
Нужна одна своя ODBC/SNMP/HTTP метрика	**Создать** свой шаблон или **Импорт** YAML из export-файла
Доработать импортированный шаблон	Выделить строку → **Редактор**

Три кнопки на странице

1. **Создать** / **Импорт** — слева в topbar рядом с крошками, как на других списках.
2. **Редактор** / **Экспорт** / **Удалить** — появляются у заголовка таблицы после выбора строк чекбоксами (**Редактор** — только для одного шаблона).
3. Альтернатива: клик по строке → карточка → **Редактор**.

Вкладки редактора (как в wiSLA)

Вкладка	Содержимое
Шаблон	Имя, vendor, version, description, группы, теги
Связанные	Выбор linked templates из каталога (chips), не CSV
Элементы данных	Обычные items (без discovery / prototypes) — split list + form
Обнаружение	Discovery rules и item prototypes
Триггеры	name, expression, recovery, priority, status, item key
Макросы	`{$MACRO}`, value, description, config JSON для «Настроить» на узле

Graphs / Dashboards / Web — пока только счётчики на списке и import/export.

Пошагово: свой шаблон с ICMP

1. **Создать** → имя, например `My server (Custom)` → группы `Templates` → **Создать**.
2. Вкладка **Связанные** → выберите `Template Module ICMP Ping` — ping без ручного добавления метрик.
3. **Элементы данных** — только если нужны **дополнительные** метрики:
   - **Добавить метрику** → справа форма
   - **Key** — уникальный идентификатор, например `my.custom.metric`
   - **Тип сбора** — откуда берётся значение (Agent, SNMP, HTTP, ODBC…)
   - **Сохранить метрику** — строка появится в списке слева
4. При необходимости заполните **Триггеры** / **Макросы**.
5. **Проверить** — красные пункты (**error**) нужно исправить: в strict mode сохранение блокируется.
6. **Сохранить** (внизу окна) — шаблон в каталоге.

Редактор позволяет типы сбора, которые уже есть в runtime (Agent, SNMP, HTTP, ODBC…). Это **не** означает, что соответствующий золотой шаблон в каталоге помечен `Supported` — см. golden-templates-dod.md.

Элементы / обнаружение: список слева, форма справа

- Слева — метрики вкладки; клик по строке открывает правку справа.
- Справа — секции: **Основное**, **Сбор данных** (OID, URL, SQL…), **Связи и LLD**, **Preprocessing**.
- Discovery rule — галочка LLD; прототипы привязываются через поле «Discovery rule key».

После сохранения

1. Откройте **Сервисы** → карточку узла.
2. В блоке шаблонов добавьте ваш шаблон (или шаблон с linked ICMP).
3. Метрики начнут собираться по правилам collector (agent / server:snmp / server:odbc …).

Подробнее для админов: каталог seed-шаблонов. API: templates-api.md.

Шаблоны здоровья

Шаблоны здоровья

Маршрут: **SLA/SLI → Шаблоны здоровья** (`/sla-sli/health-templates`). Это **не SLA**: индекс здоровья (ИЗ) — мгновенная оценка 0–100 по **открытым проблемам триггеров**, а не по выбранным метрикам и не по доле времени доступности.

Список

- Поиск по имени и связанному шаблону мониторинга
- Фильтр по активной привязке
- Колонки: имя, шаблон мониторинга, привязка, число правил, время обновления
- Пагинация в footer карточки таблицы (25 строк по умолчанию)
- **ADMIN / OPERATOR** создают, правят и удаляют; **VIEWER / INFOSEC** только смотрят
- Для каждого **шаблона мониторинга узла с включёнными триггерами** система сама создаёт шаблон здоровья `Health: …` и делает его активной привязкой (импорт YAML и seed — тот же путь). Шаблоны без таких триггеров (например ICMP-модули) **не** получают шаблон здоровья. При seed marker пишется только после фиксации строки `health_templates` в БД (иначе на пустой базе каталог мог остаться почти пустым).
- На один шаблон мониторинга можно завести **несколько** шаблонов здоровья (ручное создание). Активная привязка всё равно одна. Сид при старте синхронизирует все такие шаблоны и не падает, если их больше одного.
- Создание вручную: имя + шаблон мониторинга; в списке только источники с триггерами. Пустой каталог триггеров — ошибка.
- **Сохранить** в редакторе возвращает в список шаблонов здоровья

Редактор

Шаблон здоровья привязан к **одному** шаблону мониторинга узла. Активная привязка — не больше одной на шаблон мониторинга.

- В шапке карточки **Шаблон** справа — статус привязки (**Активен** / **Не активен**). Поле **Привязка** справа от имени: кнопка «Снять» / «Сделать активной» в стиле остальных контролов темы.
- На **обзоре узла** клик по блоку индекса (gauge) и на списке **Мониторинг** клик по чипу **ИЗ** открывают одну модалку. Справа от процента — переключатель **Потоки** / **Радар**. Потоки: Sankey от узла **Здоровье** (цвет как у текущего индекса; дальше остаток по осям, штраф от открытых проблем, потолок критичности). Если на оси больше 12 спокойных правил, они сворачиваются в узел с числом («14 правил»); клик открывает полный список. Радар: вершины — теги `scope:*` выбранных правил шаблона (локализованные имена: доступность, производительность и т.д.), центр 0%, край 100% по категории. Данные грузятся при открытии.

- **Веса важности** — radar-диаграмма (многоугольная сетка по осям, цвета из темы). Подписи осей остаются на диаграмме и не перекрывают список весов справа. Показаны только уровни с **включёнными правилами**; вес уровня без правил **перераспределяется** на остальные, ось скрывается. У важности с хотя бы одним активным правилом вес **не меньше 1%**. Задать вес: перетащить вершину, стрелки клавиатуры, кнопки **–** / **+** или поле процента в легенде. **Рассчитать автоматически** возвращает веса к продуктовым долям (disaster 30 / high 25 / average 20 / warning 15 / info 5 / not classified 5) и масштабирует их на активные оси. Сумма **всегда 100%**. При расчёте score вес важности делится поровну на включённые правила этой важности.
- Правила — таблица триггеров шаблона мониторинга (как список шаблонов): сортировка по столбцу (включая **Активно**), «Столбцы» (порядок и видимость), изменение ширины; в строке — toggle включения, имя и описание, важность, теги; **Параметры** (критичность, потолок, LLD). Фильтры: поиск, важность, включение, LLD и **панель тегов** (OR по выбранным чипам). Без группировки по категории. В шапке **Активно** — общий переключатель только по **показанным** строкам (текущие фильтры и поиск); состояние шапки не сохраняется.
- По умолчанию правила важности **Information** и **Not classified** выключены (не влияют на индекс, пока оператор их не включит). Warning и выше включены. Веса осей без активных правил сохраняются как 0% (масса уходит на остальные).
- **Критичность:** обычное / ключевая функция (потолок score, по умолчанию 60) / отказ услуги (потолок 0). Поле потолка видно, когда его можно править (не «обычное») или когда значение уже задано.
- Триггеры недоступности **системы** по умолчанию — **отказ услуги** (ИЗ = 0), даже если в шаблоне мониторинга важность Average/Warning/High: агент («wiSLA Agent is not available»), SNMP («No SNMP data collection», ICMP ping), ODBC («Service is down / unreachable / unavailable», нет данных, порт БД), JMX/HTTP без агента (нет данных, status/stats page, connection unavailable). Проблема `agent.offline` (нет heartbeat) тоже обнуляет индекс. LLD-экземпляры не считаются отказом всей системы.
- **LLD:** одно правило на прототип. Режимы: любой экземпляр, доля затронутых, каждый экземпляр с потолком.
- Источник триггера может отсутствовать или быть выключен в шаблоне мониторинга — правило в шаблоне здоровья сохраняется.

На объекте мониторинга

На обзоре узла блок ИЗ вторичен относительно SLA. Он показывает текущий score; шаблоны здоровья, даты правки/привязки, переопределения и hash спрятаны под **Подробная информация** (свёрнуто по умолчанию, чтобы не растягивать карточку узла слева). Предупреждение «глобальный шаблон менялся после переопределений» тоже в этой секции. Ленту истории изменений правил на блоке не показывают.

Меню ⋯ на карточке узла открывает **разреженные переопределения только этого узла** (включение правила, критичность, потолок, LLD) — без правки глобального шаблона. Таблица в диалоге совпадает с редактором шаблона: колонки **Участие**, триггер, важность, теги, **Параметры**; «Как в шаблоне» — наследовать глобальное правило. Поиск по имени триггера и шаблону. Кнопка **Открыть шаблон** ведёт в редактор связанного шаблона здоровья; если активного шаблона нет — в список шаблонов здоровья.

Подробнее о формуле: SLA и индекс здоровья.

Воркфлоу уведомлений

Воркфлоу уведомлений

Страница `/workflows`.

Как открыть

- Сайдбар **Администрирование → Воркфлоу** (`/workflows`) — все роли, у которых виден пункт (включая ADMIN).
- Из **Настройки → Уведомления** — ссылка «Открыть воркфлоу».

Список: сводка (все / активны / выключены), поиск, таблица. Один объект: холст + публикация + тумблер «активен» + история запусков.

Вкладки редактора — **Холст** | **Запуски**, underline сразу под крошками (как шаблоны и обнаружение). Имя, описание, включение и удаление — кнопка **Редактировать**. В диалоге **Удалить** слева внизу (не рядом с **Сохранить**); перед удалением нужно подтверждение. **Сохранить** пишет черновик графа; запуски идут по **опубликованному** графу, и только если воркфлоу включён.

На холсте блоки — цветные карточки. Слева внизу — масштаб (по центру / + / −). По нижнему краю холста — полоса с подсказками: связь, **Если** Да/Нет, **Пауза с ожиданием** Дождались/Время вышло, удаление/изгиб линии, колесо. **Связь:** точка справа (у **Если** — Да / Нет; у паузы с событием — **Дождались** / **Время вышло**), затем **левый вход** следующего блока. Пока тянете связь, левые входы пульсируют цветом блока; наведите на блок — пунктир «прилипает». Клик или отпускание в этом состоянии ставят связь (не обязательно попадать в точку). Клик по пустому месту без прилипания — отмена.

- Клик по блоку или линии — выделить, **Delete** / **Backspace** удаляет. У блока также корзина и замок в инспекторе справа; у линии — кнопка «Удалить связь».
- Рамка мышью по пустому холсту выделяет блоки и группы; перетаскивание одного выделенного сдвигает все. **Shift** добавляет к выделению. Сдвиг холста — правая кнопка, средняя кнопка мыши, **Alt** или пробел.
- Замок слева внизу холста блокирует редактирование графа (панорама и масштаб остаются). Замок на блоке или группе не даёт их сдвинуть или удалить.
- **Группа** в палитре — цветная рамка с названием, значком и необязательной картинкой из файла. Цвет заливки — из палитры или свой; края — цвет, толщина и скругление. На запуск воркфлоу не влияет. Перетаскивание группы двигает блоки внутри неё.
- Новый блок из палитры появляется в **видимой** области холста, рядом с центром кадра, а не внизу графа.
- Точка на середине линии — потянуть, чтобы изогнуть.
- Колёсико — масштаб, перетаскивание пустого места — сдвиг, кнопка **По центру** — вписать граф в экран.

Цепочка читается слева направо: **Триггер** (когда стартовать) → **Если** (что верно сейчас) → **Пауза** / **Уведомить** / **Тикет**.

Пустой новый воркфлоу сразу показывает **готовые сценарии**. Если очистить холст, панель сама не возвращается — её открывает кнопка **С чего начать** внизу палитры **Блоки**. Карточка **добавляет** сценарий рядом с уже лежащими блоками (не заменяет их): написать при открытии; напомнить, если не взяли за 15 минут; повторять, пока открыта; написать при

закрытии; взяли — исполнителю; высокая — исполнителю иначе общий канал; только в рабочие часы; сутки без закрытия — тикет; ночью тихо; **лестница эскалации** (сразу уведомить участников воркфлоу, затем ещё раз через 15 и 30 минут, пока не возьмут или не закроют). Карточка тикета недоступна, пока в Интеграциях нет HTTP-подключения. Воркфлоу **не закрывает** проблему.

Блоки

Тип	Назначение
Триггер	Старт, когда проблема открылась, её взяли в работу, сменили исполнителя или закрыли.
Если	Проверка **текущего** состояния после старта. Точка **Да** / **Нет**. Не дублирует триггер.
Пауза	Минуты (чипы 15 мин / 1 ч / 1 сутки) и по желанию **По расписанию**. Можно ждать **взяли** / **закрылась** — тогда два выхода: **Дождались** и **Время вышло**.
Уведомить	Email / Telegram / MAX / webhook. Опция **Повторять** — тот же блок шлёт снова, петлю рисовать не нужно.
Тикет	Отправляет аварию в Jira, ITCM или webhook. Куда слать — именованная система из Интеграций; секрет на холсте не хранится.

Черновик не влияет на матчинг, пока не **опубликовать**. Активен матчит **published** граф.

Триггер

- **Когда запускать** — типы событий (можно несколько). Это единственное место, где задаётся старт.
- **Мин. критичность** — порог или **Любая** (не важно).
- **Узлы** — отдельный блок: поиск по имени, IP и агенту, точка статуса (доступен / нет / обслуживание). Ничего не отмечено — **все узлы**. Выбранные узлы показываются чипами.

Если (И / ИЛИ)

Блок ставят **после** триггера. Триггер уже сработал; здесь смотрим, что с проблемой **сейчас**.

1. Добавьте одно или несколько условий: проблема ещё открыта, никто не взял, взята в работу, критичность, чем начался запуск, узел.
2. Между условиями — свои кнопки **И** / **ИЛИ**. **И** склеивает группу, **ИЛИ** начинает новую: `(A и B) или (C и D)`.
3. Соедините точку **Да** с тем, что делать при совпадении, и **Нет** — с альтернативой.

Если у условия нет своей связки, берётся запасное `combine` блока (как в старых графах: все **И** или все **ИЛИ**).

Условие «узел из списка» срабатывает, только если отмечены хосты.

Старые графы без И/ИЛИ остаются рабочими. В инспекторе их можно переключить на **Если**.

Пауза

Сначала пауза в минутах (чипы **15 мин** / **1 ч** / **1 сутки**, можно 0). Кнопка **По расписанию** открывает дни недели; у каждого включённого дня свои часы «с» и «до». После паузы шаг дождётся ближайшего такого окна. Часовой пояс — из настроек инстанса. Старые графы с одним интервалом на все дни по-прежнему работают.

Ждать событие этой же проблемы — отдельно от триггера. Триггер **стартует новый** запуск; здесь **текущий** запуск просыпается, когда проблему взяли в работу и/или закрыли, либо когда вышло время. Если событие уже случилось к моменту входа в паузу (уже взята / уже закрыта) — сразу ветка **Дождались**, без сна. Пустой список событий — как раньше: один выход, только таймер.

На вкладке **Запуски** шаг паузы показывается как «Дождались: …» или «Время вышло», а не сырой `resumed:true`.

Уведомить

1. Отметьте **каналы** — как отправить.
2. Выберите **кому**:
   - **Я укажу адреса** — email, Telegram chat id и MAX id отдельными полями.
   - **Личные каналы пользователей** — адреса из Аккаунт → Уведомления.
   - **Кто взял проблему** — исполнителю. Если никто не взял, шаг пропускается.
   - **У кого есть доступ к воркфлоу** — owner и ACL.
3. **Webhook** всегда идёт на URL из Интеграций, поле адреса ему не нужно.

Повторять — тот же блок отправит снова через N минут (чипы **15 мин** / **1 ч** / **1 сутки** в поле интервала, как у Паузы), пока проблему не возьмут/не закроют (или только пока не закроют) либо пока не исчерпан максимум. Стрелку назад на холсте не рисуйте: публикация по-прежнему отвергает циклы.

Тикет

Блок **Тикет** создаёт задачу во внешней системе. В инспекторе выбираете **куда отправить**:

1. **Jira** — если интеграция включена (URL, email, token, проект).
2. **Yandex Tracker** — если включены org id, OAuth и очередь.
3. Свой webhook — только если в Интеграциях уже есть «Другая система».

Учётные данные задаются в **Настройки → Интеграции**. На холсте их нет: для Jira и Tracker JSON править не нужно — уйдёт тема проблемы и описание (узел, критичность, ссылка). Для своего webhook по-прежнему шаблон JSON и кнопки **данные аварии**.

Повторный запрос на ту же аварию и ту же систему пропускается — второй тикет не создастся. С карточки Событий тоже создаётся **не больше одного** issue; flap-reopen пишет комментарий, а не новую задачу. Ошибка ITSM не откатывает открытие проблемы.

Если систем ещё нет: администратор добавляет **Jira** или **Tracker** в Интеграциях (из инспектора есть ссылки). Оператор без доступа к Настройкам видит подсказку попросить администратора.

Kill-switch уведомлений (`notifications.enabled`) глушит и тикеты.

Тихие часы

В **Редактировать**, рядом с «включить воркфлоу»: дни и часы, когда **письма и тикеты** ждут начала окна. Сам запуск по событию ночью всё равно создаётся. Если к началу окна проблема уже закрылась — утреннее письмо/тикет не уйдёт. **High и disaster сразу** (по умолчанию включено) — warning и ниже ждут утра.

Поделиться

Две оси:

- **Копия** — независимый воркфлоу, владелец = получатель. VIEWER как получатель копии — ошибка (нужен access).
- **Доступ** — тот же объект + ACL `view` / `edit`.

Каналы notify: **оставить** или **очистить**. Access + очистить ⟶ получатели `acl_members_media` (личные каналы owner и ACL).

Создают/правят ADMIN и OPERATOR. VIEWER читает доступные ему и настраивает свои каналы. INFOSEC читает все, каналы не пишет.

Личные email / Telegram: **Аккаунт ⟶ Уведомления**. Транспорт инстанса (SMTP, Telegram, MAX, webhook) — Интеграции.

См. также

- Уведомления (админ) — kill-switch, layout страницы, роли, **карта кода**
- Воркфлоу уведомлений (справочник) — domain events и матчинг
- Nav фронта — IA для разработчиков
- Роли — кто куда ходит

Обслуживание узла

Обслуживание узла (ППР)

На списке **Мониторинг** (`/services`) выберите один или несколько объектов и нажмите **ППР**. Задайте интервал (тип **ППР** или **исключение SLA**), при необходимости — причину и еженедельное повторение (8 недель). Пока now пересекает окно ППР, в таблице виден бейдж **ТО**.

Отменить запланированное или **остановить** активное окно:

- на списке — блок **Активные и ближайшие окна** (кнопка справа у строки);
- на карточке узла/канала/доступности — блок **Календарь окон** (кнопка **Отменить** / **Остановить**; подсказка по смыслу — в `?` у заголовка).

Эффект

	ППР (`maintenance`)	**Исключение SLA (`exception`)**
Новые проблемы	не открываются, пока now пересекает окно	открываются как обычно
Уведомления / воркфлоу	не матчатся	идут
SLA	время окна вычитается (`maintenance_excluded_ms`)	время окна вычитается (`exception_excluded_ms`)
Лента статусов	тон ППР	тон «исключено из SLA»

Источник истины — пересечение окон с now, а не только флаг `maintenance` в карточке (флаг — кэш для списка).

Уже открытые события остаются в истории — их можно закрыть вручную.

Когда включать

- Плановое обновление ОС / агента
- Сетевые работы, заведомо ломающие ICMP/SNMP
- Нагрузочные тесты, которые не должны будить дежурных
- Ретроспективное исключение из SLA (вид **исключение SLA**, интервал в прошлом) — события не переписываются

Когда выключать

Сразу после окончания работ — **Остановить** активное окно (или дождаться конца интервала), иначе мониторинг «молчит» и вы пропустите реальный инцидент.

Административный контекст retention: admin/maintenance-retention.

Входящие trap / syslog

SNMP trap и syslog (входящие)

Не отдельная страница. Приём — на **узле** (шаблон + IP); лента — вкладка **Входящие сообщения** в панели метрик (только если есть trap/syslog items); аварии по триггерам — в **Событиях** / воркфлоу (и краткий блок «Последние события» в сайдбаре).

Как настроить

1. **Админ** ⟶ Настройки ⟶ Мониторинг ⟶ **Приём trap / syslog**:
   - SNMP trap: вкл (default), порт **162**.
   - Syslog: вкл **явно** (default выкл), порт **514**.
   - **Принимать с loopback**: для Docker Desktop lab обычно вкл; в проде часто выкл.
   - После смены — **перезапуск backend**. Linux unit: `CAP_NET_BIND_SERVICE` (wisla-backend.service).
2. Узел ⟶ Шаблоны ⟶ **Принимать SNMP traps** / **Принимать syslog**.
3. IP интерфейса = peer, как его видит backend (на Docker Desktop часто `127.0.0.1`).
4. На устройстве: destination = IP wiSLA, порт из настроек.
5. Обзор узла ⟶ вкладка **Входящие сообщения**. Поиск, период (1ч–7д / свой диапазон), выпадающие фильтры типа и критичности, чипы как на «Последних событиях», **свернуть повторы**, пагинация. Проблемы — триггеры по `{$...PROBLEM.REGEXP}` или воркфлоу.

Триггеры по умолчанию

Шаблон	Макрос	Смысл
`SNMP traps`	`{$SNMPTRAP.PROBLEM.REGEXP}`	`find(...iregexp...)=1` ⟶ WARNING
`Syslog`	`{$SYSLOG.PROBLEM.REGEXP}`	то же для syslog

Макросы правятся на узле (Настроить / Параметры). Пустой или слишком широкий regexp = шум в Событиях.

Техдетали

	Trap	Syslog
Listener	`SnmpTrapListener` (SNMP4J v1/v2c/v3 + inform ack)	`SyslogListener`
Type / collector	`SNMP_TRAP` / `server:snmptrap`	`SYSLOG` / `server:syslog`
Key	`snmptrap.fallback`, `snmptrap[oid]` (linkDown/linkUp/...)	`syslog.fallback`, `syslog[err]`...
Парсинг	SNMP4J PDU ⟶ trap OID + varbinds	PRI ⟶ facility/severity labels

	Trap	Syslog
Peer	`IncomingPeerMatcher` + `monitoring.incoming_accept_loopback`	то же

- API: `GET …/incoming/capable`, `GET …/metrics/recent` (PG ring + ClickHouse fallback).
- Smoke: incoming-trap-syslog-smoke.md.
- ADR / matrix: ADR-002, zabbix-template-runtime.

Не в scope (пока)

TCP/TLS syslog, отдельная страница Incoming, auto-fallback на порт 1162, полный trap catalog на все enterprise MIB.