Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

AIOps своими руками: предиктивный мониторинг и self-healing

Строим контур предсказания инцидентов и авто-починки на Zabbix 7.0 LTS, Ansible и LLM-ассистен...

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Классический мониторинг сообщает об аварии, когда база 1С уже лежит: пороги статичны, алертов сотни, о переполненном диске узнают по звонку клиента. Мы решаем задачу перехода от реакции к предсказанию: инциденты видим за часы и сутки до аварии, а типовые чиним автоматикой без участия человека. Цена бездействия — многочасовые простои и выгорание дежурных в шуме алертов.

Почему это важно бизнесу

- Час простоя 1С в закрытие месяца — остановленная бухгалтерия, срыв платежей и отчётности в ФНС
- Предсказанный инцидент — плановые работы вечером; непредсказанный — авария в разгар рабочего дня
- Автоматика закрывает ночные типовые инциденты без платных выездов и подъёма дежурного
- Шум из сотен алертов приучает игнорировать мониторинг — так пропускают настоящую аварию

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

7.0 LTS

Zabbix — ядро контура: trend-функции аномалий и прогнозов доступны из коробки, без внешнего ML

по докам zabbix.com

3 MAD

deviations=3 в trendstl() (devalg по умолчанию — mad): алерт только при выходе за 3 отклонения...

по докам zabbix.com, trendstl

72 ч

горизонт timeleft()/forecast() для дисков и баз 1С — алерт за трое суток до исчерпания

наш стандарт

5 мин

шаг эскалации: step 1 — remediation-плейбук, инженер подключается только если авто-фикс не пом...

наш стандарт

3.x

Prometheus + Alertmanager в docker-стеках: group_wait 30s, group_interval 5m против шторма

по докам prometheus.io

90 дн

минимум истории трендов для trendstl с недельной сезонностью — на короткой истории baseline вр...

наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Предиктивный контур на Zabbix 7.0 LTS

Что настраиваем

Серверы 1С, MS SQL/PostgreSQL, терминальные фермы, гипервизоры; Zabbix-сервер в отдельном сегменте, на хостах — Zabbix agent 2

Как мы это делаем

1. Разворачиваем Zabbix server 7.0 LTS + TimescaleDB: партиционирование, retention history 31d / trends 1y — trend-функциям нужна длинная история
2. На ключевые метрики (CPU, IOPS, длительность фоновых заданий 1С) вешаем baselinewma()/baselinedev(): сравнение с тем же часом прошлых недель, а не с константой
3. Аномалии: trendstl() с eval-периодом 21d, окном детекции 24h, сезоном 7d и deviations=3 — ловим нетипичные очереди и логин-штормы без ручных порогов
4. Прогноз: timeleft(/host/vfs.fs.size[C:,pfree],30d,0)<72h и forecast() по приросту баз SQL — заявка на расширение диска уходит до аварии
5. Наверх — Grafana 12.x с плагином Zabbix: дашборд «прогноз vs факт» по каждому клиенту, чтобы видеть дрейф baseline глазами

РЕЗУЛЬТАТ

Триггеры перестают зависеть от ручных порогов: что для одной бухгалтерии норма, для другой — авария, и baseline учитывает это сам. Дисковые и ёмкостные инциденты уходят в плановые работы: ресурсы расширяем по прогнозу, а не после падения базы 1С.

КЛЮЧЕВОЙ НЮАНС

Trend-функции считаются по таблице trends, а не history: пока не накоплено 2–3 полных сезона, держим триггеры в режиме «только регистрация». TrendFunctionCacheSize (кэш trend-функций, по умолчанию 4M) и ValueC...

РЕАЛИЗАЦИЯ 2 ИЗ 3

Корреляция событий и авто-remediation

Что настраиваем

Контур реакции: Zabbix actions + эскалации, плейбуки ansible-core 2.19 на бастионе, алерты в Telegram-канал дежурных

Как мы это делаем

1. Режем шторм: зависимости триггеров (канал → всё, что за ним), event correlation по тегам, maintenance-окна на регламентные ребуты
2. Action step 1 (0–5 мин): webhook дёргает плейбук из белого списка — рестарт зависшего rphost, службы печати, очистка temp; step 2 — уже инженер
3. Каждый плейбук идемпотентный и с гвардом: не больше 2 запусков за 30 мин на хост, дальше стоп и принудительная эскалация человеку (анти-flapping)
4. Все авто-действия помечаются тегом remediation=auto в событии Zabbix — по нему же считаем долю инцидентов, закрытых без людей

РЕЗУЛЬТАТ

Ночные типовые инциденты (зависший rphost, забитый спулер, переполненный temp) закрываются за минуты без подъёма дежурного. Человек разбирает только то, что не починилось с первого прогона, — уже с собранным контекстом и историей попыток.

КЛЮЧЕВОЙ НЮАНС

Авто-remediation без ограничителей опаснее его отсутствия: цикл «упал — рестартовали — упал» маскирует деградацию железа. Поэтому счётчик попыток и эскалация зашиты в каждый плейбук, а список действий — только…

РЕАЛИЗАЦИЯ 3 ИЗ 3

LLM-ассистент дежурного: RCA-дайджест

Что настраиваем

Надстройка над алертом: python-коллектор на бастионе + LLM API; доступ строго read-only, без права на действия

Как мы это делаем

1. По срабатыванию триггера webhook собирает пакет: метрики узла, хвост журналов (Windows event log / syslog), связанные события за последние 30 минут
2. Коллектор до отправки маскирует пароли, ключи и ПДн регэкспами — наружу уходит только технический контекст
3. LLM получает пакет плюс карту зависимостей узла и возвращает в Telegram гипотезу первопричины, три шага проверки и ссылку на runbook
4. Ответ всегда помечен как «гипотеза»: решения и действия — только за инженером; качество разборов еженедельно сверяем с фактическим RCA

РЕЗУЛЬТАТ

Дежурный получает не голый алерт, а выжимку: что упало, что менялось вокруг, с чего начать. Вход в контекст по чужой инфраструктуре сокращается с десятков минут до двух-трёх — критично, когда клиентов много, а дежурный один.

КЛЮЧЕВОЙ НЮАНС

LLM в контуре эксплуатации — только советник: никаких кредов в промпте, никакого выполнения команд из ответа модели. Пайплайн строим так, чтобы даже утечка промпта не дала злоумышленнику ничего полезного.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

✕ Статические пороги на всём

CPU>90% для сервера 1С в закрытие месяца — норма. Ключевые метрики переводим на baselinewma/baselinedev, статику оставляем дискам и доступности.

✕ Baseline учится на грязной истории

Инцидентные недели попадают в тренд и становятся «нормой». Держим новые триггеры месяц в тестовом режиме и пересматриваем eval-период после аварий.

✕ Рестарт-цикл авто-фикса

Плейбук без счётчика попыток бесконечно рестартует умирающую службу и прячет деградацию железа. Лимит 2 запуска за 30 мин, дальше — эскалация.

✕ Шторм без корреляции

Падение одного канала рождает сотни алертов с хостов за ним. Зависимости триггеров и группировка в Alertmanager обязательны до любого «AI».

✕ LLM с правом действий

Модель, исполняющая команды по собственному выводу, — источник новых аварий. У ассистента read-only доступ и роль советника, действия — за инженером.

✕ Прогноз на короткой истории

forecast() по трём дням данных даёт мусор и ложные закупки дисков. Минимум 30 дней истории, для сезонных метрик — 90.

✕ Мониторинг без самоконтроля

Умерший Zabbix — тишина, неотличимая от «всё хорошо». Внешний heartbeat-чек сервера мониторинга с независимого узла обязателен.

✕ Алерты без runbook

Дежурный видит anomaly rate 0.7 и не знает, что делать. К каждому ML-триггеру прикладываем ссылку на runbook прямо в описании триггера.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Zabbix 7.0 LTS, пороги + зависимости триггеров, алерты в Telegram
- timeleft()/forecast() на все диски и приросты баз, горизонт 72 ч
- Heartbeat-контроль самого сервера мониторинга с внешнего узла

НОРМАЛЬНО

- baselinewma/baselinedev на ключевые метрики 1С, СУБД и терминалов
- Event correlation и maintenance-окна: один инцидент — один алерт
- Белый список remediation-плейбуков с лимитом запусков и эскалацией
- Grafana-дашборды «прогноз vs факт» по каждому клиенту

ХОРОШО

- trendstl-аномалии с deviations=3 и сезоном 7d на бизнес-метрики
- LLM-дайджест RCA в Telegram: read-only коллектор, маскирование секретов
- Ежемесячный отчёт по доле инцидентов, закрытых автоматикой
- Учения game-day: гасим службу и проверяем весь контур end-to-end

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Есть ли прогнозные триггеры (timeleft/forecast) на всех дисках — или о переполнении узнаёте по факту?
- [ ] Сколько алертов приходит при падении одного канала связи — один или лавина?
- [ ] Может ли автоматика рестартовать зависшую службу, и есть ли у неё ограничитель числа попыток?
- [ ] Учитывают ли пороги CPU и очередей сезонность: закрытие месяца, утренний логин-шторм?
- [ ] Кто и за сколько минут узнает, что упал сам сервер мониторинга?
- [ ] Прикреплён ли к каждому триггеру runbook: что проверить и что дёрнуть?
- [ ] Логируются ли авто-действия так, чтобы через месяц восстановить, кто и что чинил?
- [ ] Проверяли ли контур учениями: алерт реально доходит, плейбук реально отрабатывает?
- [ ] Замаскированы ли секреты в данных, которые уходят во внешние AI-сервисы?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Разворачиваем Zabbix 7.0 LTS с TimescaleDB и предиктивными триггерами под вашу инфраструктуру
- Пишем и обкатываем remediation-плейбуки Ansible с анти-flapping гвардами под типовые инциденты 1С и Windows
- Настраиваем корреляцию, эскалации и Telegram-алертинг: один инцидент — один алерт дежурному
- Подключаем LLM-ассистента дежурного: read-only коллектор, маскирование секретов, RCA-дайджест
- Ведём ежемесячный отчёт: доля авто-закрытых инцидентов, точность прогнозов, дрейф baseline

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Zabbix Manual — Trend functions (trendstl, baselinewma) (zabbix.com — 7.0 LTS)

02 Zabbix Manual — Event correlation (zabbix.com — 7.0 LTS)

03 Zabbix Manual — Escalations, remote commands (zabbix.com — 7.0 LTS)

04 Prometheus docs — Alerting rules (prometheus.io — 3.x)

05 Alertmanager docs — Grouping and inhibition (prometheus.io — 0.33)

06 Grafana docs — Zabbix data source plugin (grafana.com — 12.x)

07 Ansible docs — Playbooks and idempotency (docs.ansible.com — 2.19)

08 Наш шаблон remediation-плейбука с анти-flapping гвардом (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.