Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Netdata: диагностика сервера 1С и терминалки за 5 минут

Когда Netdata быстрее Zabbix и Prometheus: наш подход к экспресс-мониторингу

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Когда сервер 1С «встал» или терминалка тормозит у 40 пользователей, время считают минутами: бухгалтерия не закрывает смену, склад не отгружает, руководитель звонит каждые 5 минут. Разворачивать Zabbix с шаблонами и триггерами или поднимать Prometheus+Grafana ради разового инцидента — это часы настройки, которых на месте у клиента просто нет.

Почему это важно бизнесу

- Простой сервера 1С в рабочие часы — прямые убытки: бухгалтерия и склад останавливаются, отгрузки срываются
- Без объективных метрик спор «сервер тормозит» решается на глаз, а не по CPU/RAM/диску — теряем время на диагностику
- Долгая настройка мониторинга откладывает решение о докупке железа — клиент платит за деградацию месяцами
- Без хронологии инцидентов невозможно доказать поставщику оборудования или интернета SLA-нарушение
- Плохо настроенный мониторинг с ложными алертами приучает инженеров игнорировать уведомления

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

1 с

update_every=1 — секундный шаг сбора метрик Netdata из коробки, сразу видно всплеск нагрузки

Netdata docs

~2000

метрик агент Netdata собирает автоматически на типовом Linux-сервере 1С без единого шаблона

Netdata docs

15-30 с

scrape_interval Prometheus для node_exporter — баланс детализации метрик и нагрузки на TSDB

Prometheus docs

1 ГБ/тир

дефолтный лимит места на tier dbengine — первое, что мы пересчитываем на продовых серверах

Netdata dbengine

3 тира

tier0 посекундно, tier1 поминутно, tier2 почасово — структура retention под глубину истории

Netdata dbengine

60 с

типовой Update interval Zabbix agent2 для item, который мы правим под 1С-процессы

Zabbix docs

РЕАЛИЗАЦИЯ 1 ИЗ 3

Экспресс-диагностика сервера 1С за 5 минут

Что настраиваем

сервер 1С:Предприятие (кластер rphost/ragent) на Linux, разовый выезд на инцидент у клиента

Как мы это делаем

1. Ставим агент одной командой kickstart.sh --stable-channel --disable-telemetry — без БД и без шаблонов, шаг сбора 1 с
2. Открываем локальный дашборд на :19999, сразу смотрим CPU по ядрам, RAM, disk I/O и apps.plugin по процессам
3. В apps_groups.conf добавляем группу 1c: rphost* ragent* — видим долю CPU и RAM именно кластера 1С, а не всей ОС
4. Смотрим Overview за 2 минуты: iowait, swap-активность, top-процессы — понимаем, во что реально упирается сервер
5. Фиксируем срез (например disk I/O выше 80% на tier0) как основание для решения о SSD или доп. rphost-процессах

РЕЗУЛЬТАТ

Получаем объективную картину нагрузки за 5 минут вместо часов настройки Zabbix-шаблонов — понятно, упирается сервер в CPU, диск или память, и какой именно процесс кластера 1С в этом виноват, без прерывания работы пользователей.

КЛЮЧЕВОЙ НЮАНС

Netdata — не замена постоянному мониторингу: retention ограничен по умолчанию (1 ГБ на tier). Для разбора инцидента снимаем срез сразу, а для контроля SLA во времени ставим отдельно Zabbix или Prometheus.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Постоянный мониторинг терминального сервера RDS

Что настраиваем

терминальный сервер (RDS/RDSH) на 30-80 сессий, длительное наблюдение и алертинг для клиента

Как мы это делаем

1. Разворачиваем Zabbix agent2 (Go, плагинная архитектура) на терминалке, Update interval = 60 с для CPU/RAM/диска
2. Заводим LLD-дискаверинг по процессам сессий, триггер на число активных RDP-сессий выше 40
3. Настраиваем отдельные warn/crit пороги с hysteresis, чтобы алерт не дребезжал на границе значения
4. History храним 14 дней, trends — 5 лет через Housekeeping, чтобы видеть сезонность нагрузки по месяцам
5. Параллельно держим Netdata на этом же хосте как второй независимый источник для live-проверки прямо сейчас

РЕЗУЛЬТАТ

Получаем управляемые пороги и долгую историю для планирования апгрейда терминалки (когда докупать RAM/CPU), не теряя возможности за секунды увидеть live-картину через Netdata, если пользователи жалуются прямо сейчас.

КЛЮЧЕВОЙ НЮАНС

Zabbix-триггер — это реакция на опрос, а не постоянное наблюдение: слишком редкий Update interval (300 с) пропустит кратковременный всплеск CPU — для терминалок держим шаг не реже 60 с.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Единый Prometheus+Grafana контур на несколько серверов клиента

Что настраиваем

инфраструктура из нескольких хостов (1С + RDS + СУБД), нужен один дашборд и алертинг на всё

Как мы это делаем

1. Ставим node_exporter на каждый хост, Prometheus scrape_interval=15 с, отдельный job для 1С-хостов
2. Пишем кастомный exporter под метрики 1С (число активных сеансов, блокировки СУБД) на порт /metrics
3. Описываем alerting rules в Prometheus с for: 5m, а маршрутизацию по каналам держим в Alertmanager
4. Собираем Grafana-дашборд один раз, дальше расширяем PromQL-запросами под каждого клиента без правки агентов

РЕЗУЛЬТАТ

Даём заказчику единую точку правды по всей инфраструктуре с историей на месяцы вперёд и алертами в Telegram/почту, при этом Netdata остаётся инструментом первого касания для быстрой диагностики на месте.

КЛЮЧЕВОЙ НЮАНС

Pull-модель Prometheus требует открытого порта exporter'а на каждом хосте — для терминалок за NAT/pfSense явно прокидываем порт node_exporter, иначе сервер просто не достучится до цели.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Netdata как единственный монитор**
  без Zabbix/Prometheus теряем долгую историю (retention по умолчанию 1 ГБ/тир) — для SLA и трендов ставим второй контур с длинным хранением.
- ✕ **Дефолтный retention dbengine**
  1 ГБ на tier забивается за часы-дни на нагруженном сервере — пересчитываем лимит места и page cache под конкретную задачу.
- ✕ **Не задан apps_groups.conf**
  без явной группировки rphost/ragent Netdata показывает общий CPU процесса, а не кластера 1С — теряется весь смысл диагностики.
- ✕ **Слишком частый Zabbix item interval**
  10-15 с на десятках item создают лишнюю нагрузку на сервер и БД — держим 60 с для базовых метрик, короче — только для критичных.
- ✕ **Порог warn/crit без hysteresis**
  без conditional-оператора алерт дребезжит на границе — задаём отдельные warn и crit с запасом 10-15% друг от друга.
- ✕ **Открытый дашборд :19999 наружу**
  Netdata слушает 0.0.0.0 по умолчанию — закрываем порт фаерволом или биндим на 127.0.0.1 и проксируем через nginx с basic-auth.
- ✕ **Prometheus без relabel по ролям**
  без job/label по типу хоста (1С/терминалка/СУБД) PromQL-запросы и алерты превращаются в кашу — размечаем ещё в scrape_config.
- ✕ **Housekeeping не настроен**
  без Override item history/trend period старые данные копятся в БД Zabbix бесконтрольно — задаём 14 дней history и 5 лет trends.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Ставим Netdata kickstart на проблемный сервер для разового снимка нагрузки
- Настраиваем apps_groups.conf под rphost/ragent/1C-процессы
- Закрываем порт 19999 фаерволом, доступ только с нашего IP

НОРМАЛЬНО

- Держим Zabbix agent2 с item interval 60 с на терминалке и сервере 1С
- Настраиваем warn/crit пороги с hysteresis по CPU/RAM/диску
- History 14 дней, trends 5 лет через Housekeeping
- Netdata остаётся вторым контуром для live-диагностики инцидентов

ХОРОШО

- Единый Prometheus + Grafana контур с node_exporter на всех хостах
- Кастомный exporter под метрики 1С (сеансы, блокировки СУБД)
- Prometheus alerting rules с for: 5m, маршрутизация в Telegram через Alertmanager
- Netdata Cloud room для мгновенного visual triage по всем клиентам

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Установлен ли Netdata на серверах 1С и терминалках для быстрой диагностики?
- [ ] Настроен ли apps_groups.conf под процессы rphost/ragent/1C?
- [ ] Закрыт ли порт 19999 Netdata фаерволом от внешнего доступа?
- [ ] Увеличен ли лимит dbengine tier с дефолтного 1 ГБ под задачу клиента?
- [ ] Задан ли Update interval Zabbix-агента не реже 60 с для базовых метрик?
- [ ] Настроена ли hysteresis warn/crit порогов, чтобы не спамить алертами?
- [ ] Настроен ли Housekeeping — сроки history/trends под задачи клиента?
- [ ] Есть ли алерты в Telegram или почту при превышении порогов CPU/RAM/диска?
- [ ] Прокинут ли порт node_exporter/agent через NAT для серверов за pfSense?
- [ ] Хранится ли отдельно долгая история для планирования апгрейда железа?

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 Netdata под ключ на сервере 1С или терминалке за один визит — диагностика нагрузки за 5 минут
- Настраиваем постоянный мониторинг на Zabbix или Prometheus с алертами в Telegram под SLA клиента
- Пишем кастомные exporter'ы под метрики 1С (сеансы, блокировки СУБД) и терминального сервера
- Настраиваем retention, hysteresis и дашборды под конкретную инфраструктуру, а не дефолты из коробки
- Берём на абонентское обслуживание весь контур мониторинга — от установки агентов до разбора инцидентов

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 Netdata Agent Configuration / Daemon Reference (learn.netdata.cloud — 2026)

02 Database Configuration Reference (dbengine tiers) (learn.netdata.cloud — 2026)

03 Configure Health Alerts / Alert Reference (learn.netdata.cloud — 2026)

04 Install Netdata with kickstart.sh (github.com/netdata — 2026)

05 Zabbix Agent 2 concepts (zabbix.com — 7.0)

06 History and trends / Housekeeping (zabbix.com — 7.2)

07 Monitoring Linux host metrics with Node Exporter (prometheus.io — 2026)

08 Prometheus Configuration (scrape_interval) (prometheus.io — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.