

ТЕХНИЧЕСКИЙ РАЗБОР

Zabbix 7 для офиса: мониторинг 200 хостов одним сервером

Как мы разворачиваем Zabbix 7.0 LTS на одной VM: 200-500 хостов, PostgreSQL 16 + TimescaleDB



Ай-Ти Фреш

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Мониторинга либо нет, либо он «жёлтый пять лет, и его игнорируют». О падении сервера 1С, контроллера домена или почты компания узнаёт от разъярённого пользователя, а не от системы. Мы закрываем этот разрыв: разворачиваем Zabbix 7.0 LTS на одной VM под 200-500 хостов, берём 100% серверов и выборочно критичные РС, чтобы инцидент ловился за минуту, а не за час простоя.

Почему это важно бизнесу

- Час простоя 1С или AD в офисе на 50 человек — это десятки тысяч рублей потерь; мониторинг окупается одним предотвращённым инцидентом.
- Реактивная модель «ждём звонка пользователя» означает, что о проблеме узнают позже всех — в разгар отчётности это стоит дороже всего.
- Без алертов деградация (забитый диск, растущие блокировки 1С) копится незаметно и выстреливает отказом в самый неудобный момент.
- Zabbix под лицензией AGPL-3.0 — 0 ₽: платите только за VM и внедрение, стоимость не растёт при масштабировании парка.



Ключевые параметры реализации

7.0 LTS

Ставим только ветку LTS: полная поддержка до 30.06.2027, ограниченная до 30.06.2029 — 5 лет без вынужденных апгрейдов
по докам zabbix.com

до 1000

Асинхронные пайлеры 7.0 (agent, HTTP agent, SNMP) держат до 1000 одновременных проверок на процесс
Zabbix 7.0

7 дней

TimescaleDB-компрессия в Housekeeping, минимум «старше 7 дней» — ужимает историю в разы
по докам zabbix.com

32M

CacheSize по умолчанию (конфиг-кеш); под 200 хостов поднимаем до 256M, иначе «increase CacheSize» в логе
наш стандарт

1-30 с

Глобальный Timeout (умолч. 3), в 7.0 переопределяется per-item; тяжёлым проверкам 1С даём больше
Zabbix 7.0

4 vCPU

Наш профиль VM под 200-500 хостов: 4 vCPU / 8 ГБ ОЗУ / 200 ГБ SSD, история 90 дней на TimescaleDB
наш стандарт



Ядро мониторинга: одна VM, Zabbix 7.0 + PostgreSQL 16 + TimescaleDB

Что настраиваем

Один сервер Ubuntu 24.04 LTS (4 vCPU / 8 ГБ / 200 ГБ SSD): server, БД, фронт, Agent 2

Как мы это делаем

- 1 Репозиторий zabbix-release 7.0, ставим zabbix-server-pgsql, zabbix-frontend-php, zabbix-nginx-conf, zabbix-sql-scripts, zabbix-agent2
- 2 PostgreSQL 16 + TimescaleDB: CREATE EXTENSION timescaledb, импорт server.sql.gz и timescaledb/schema.sql под пользователем zabbix
- 3 postgresql.conf: shared_preload_libraries='timescaledb', shared_buffers 2GB, effective_cache_size 5GB, work_mem 32MB
- 4 zabbix_server.conf под 200 хостов: StartPollers 20, StartDBSyncers 4, CacheSize 256M, HistoryCacheSize 128M, ValueCacheSize 128M
- 5 systemctl enable --now zabbix-server zabbix-agent2 nginx php8.3-fpm; мастер, часовой пояс Europe/Moscow, смена дефолта Admin/zabbix

РЕЗУЛЬТАТ

Полноценное ядро на 200-500 хостов поднимается за вечер на одной машине, без прокси-зоопарка. TimescaleDB держит 90 дней истории компактно, а Agent 2 сразу мониторит сам сервер мониторинга.

КЛЮЧЕВОЙ НЮАНС

Дефолтные кэши (CacheSize 32M, HistoryCacheSize 16M, ValueCacheSize 8M) малы под 200 хостов — поднимаем заранее, иначе в логе «please increase CacheSize» и потеря значений. Пароль Admin/zabbix меняем сразу.



Агенты и шаблоны: Windows-серверы, MSSQL, 1С:Предприятие

Что настраиваем

Zabbix Agent 2 на Windows-серверах и критичных PC; шаблоны из коробки + наш шаблон 1С

Как мы это делаем

- 1 Agent 2 через GPO/MSI, порт 10050 TCP открыт в Windows Firewall на вход и исход; HostMetadata для автрегистрации по ролям
- 2 Шаблоны из коробки: Windows by Zabbix agent active, Active Directory by Zabbix agent, MSSQL by Zabbix agent 2, Hyper-V by Zabbix agent 2, MikroTik by SNMP, APC UPS by SNMP
- 3 MSSQL — загружаемый плагин Agent 2 (ставится отдельно, $\geq 7.0.10$): Plugins.MSSQL.Default.Uri, учётка zabbix_monitor с правами только на VIEW SERVER STATE
- 4 1С на трёх уровнях: процессы rghost/ragent/rmngnr, метрики через gac (сеансы, блокировки), бизнес-KPI (время типового отчёта)
- 5 HTTP/HTTPS-эндпоинты: Website certificate by Zabbix agent 2 + HTTP agent items на срок сертификата и код ответа

РЕЗУЛЬТАТ

Однородный сбор с Windows, SQL, Hyper-V, сети и 1С в одном продукте, без node_exporter на каждой железке. Автрегистрация по HostMetadata сама расставляет хосты по группам и шаблонам.

КЛЮЧЕВОЙ НЮАНС

В 7.0 плагины MSSQL/PostgreSQL/Mongo — загружаемые, а не «встроенные»: без отдельного пакета и версии плагина $\geq 7.0.10$ метрик SQL не будет. Штатного шаблона 1С нет — делаем и сопровождаем свой.



Алерты, антишум и хранение: Telegram, зависимости, компрессия

Что настраиваем

Media type Telegram, действия по severity с эскалацией, TimescaleDB-компрессия и maintenance-окна

Как мы это делаем

- 1 Media type Telegram (бот через @BotFather), parse_mode HTML; на нестабильных каналах — резервный SOCKS5-прокси в настройках медиа-типа
- 2 Action по severity: >=Warning дежурному, Disaster — эскалация через 5 и 15 минут; Information/Warning уводим в служебный канал
- 3 Trigger dependencies: триггер сервера зависит от триггера аплинка/коммутатора — при падении сети не сыпем дублями
- 4 Компрессия истории — в Administration → Housekeeping (минимум «старше 7 дней»), а не ручной add_compression_policy
- 5 Перед регламентами — Maintenance windows на хост/группу, чтобы не будить дежурных алертами «host unreachable»

РЕЗУЛЬТАТ

Дежурные видят реальные инциденты, а не поток дублей. История ужимается в разы, housekeeper и дашборды не тормозят, а плановые работы не порождают ложных ночных алертов.

КЛЮЧЕВОЙ НЮАНС

Компессию в 7.0 держит сам Zabbix через Housekeeping (мин. 7 дней) — ручные политики TimescaleDB с ней конфликтуют. Сжатые чанки read-only: правку схемы и бэкфилл планируем заранее.

Подводные камни

× **Забыли включить компрессию — БД пухнет**

Без History/Trends compression в Housekeeping база растёт до сотен ГБ за месяцы: бэкап тянется часами, housekeeper и дашборды виснут. Включаем сразу на новом сервере.

× **Ручная компрессия вместо Housekeeping**

В 7.0 компрессией управляет сам Zabbix; ручные политики TimescaleDB конфликтуют. Настраиваем в Administration → Housekeeping, минимум «старше 7 дней».

× **Дефолтные кешы под 200 хостов**

CacheSize 32M, HistoryCacheSize 16M, ValueCacheSize 8M по умолчанию малы: в логе «increase CacheSize» и потеря значений. Поднимаем до 256M/128M заранее.

× **Все алерты в один чат**

Общий канал со всеми severity превращается в шум, который перестают читать. Разводим по severity: статистику — в служебный, Disaster — в отдельный громкий канал.

× **Нет trigger dependencies**

Упал аплинк — Zabbix шлёт и про коммутатор, и про все недоступные за ним хосты. Без зависимостей дежурный тонет в дублях в самый момент аварии.

× **Забыли Maintenance перед регламентом**

Плановая перезагрузка без maintenance-окна будит дежурных пачкой «host unreachable» ночью. Ставим хост/группу в обслуживание за пару кликов до работ.

× **MSSQL-плагин не поставлен отдельно**

В 7.0 MSSQL/PG/Mongo — загружаемые плагины Agent 2, ставятся отдельным пакетом (≥7.0.10). Без него шаблон висит, метрик SQL нет.

× **PHP-FPM pm.max_children=5 по умолчанию**

Дефолтные 5 воркеров плюс фон из Grafana дают фронт по 8 секунд. Поднимаем pm.max_children до 20, pm.start_servers 5 — открывается за полсекунды.

Как правильно

МИНИМУМ

- Zabbix 7.0 LTS на одной VM, 100% серверов под наблюдением, алерты в Telegram по severity
- PostgreSQL 16, смена дефолта Admin/zabbix, порт агента 10050 только во внутренней сети
- Готовые шаблоны Windows/Linux by Zabbix agent на все серверы, интервал опроса 60 с

НОРМАЛЬНО

- PostgreSQL 16 + TimescaleDB, компрессия в Housekeeping «старше 7 дней», retention 90 дней
- Agent 2 через GPO, автoreгистрация по HostMetadata, trigger dependencies на апплики
- Action с эскалацией по severity, Maintenance-окна на регламенты, кешы подняты под нагрузку
- SNMP-шаблоны на MikroTik, APC UPS, принтеры; Website certificate на HTTP-эндпоинты

ХОРОШО

- NOC-дашборд дежурным + Executive-отчёт (аптайм, MTTR) руководителю раз в неделю
- Zabbix-прокси с memory buffer на удалённых площадках со слабым 4G-каналом
- Свой шаблон 1C: процессы rphost/rmngr, метрики gas, бизнес-KPI по времени отчётов
- Grafana через официальный плагин для отчётов, резервный SOCKS5 для Telegram-алертов

Чек-лист самопроверки

- Стоит ли ветка именно 7.0 LTS (поддержка до 2029), а не промежуточный не-LTS релиз?
- Включена ли TimescaleDB-компрессия в Administration → Housekeeping и задан ли retention истории?
- Подняты ли CacheSize/HistoryCacheSize/ValueCacheSize под реальное число хостов и NVPS?
- Взяты ли под наблюдение 100% серверов, а критичные рабочие станции — выборочно?
- Настроены ли Action по severity с эскалацией и отдельными Telegram-каналами?
- Проставлены ли trigger dependencies на апплинки и коммутаторы, чтобы не было дублей?
- Есть ли процесс постановки хостов в Maintenance перед плановыми работами?
- Сменён ли дефолтный пароль Admin/zabbix и закрыт ли порт агента 10050 наружу?
- Поставлен ли отдельно загружаемый MSSQL-плагин Agent 2 ($\geq 7.0.10$) там, где есть SQL?
- Есть ли резервный маршрут для алертов (SOCKS5) на случай блокировки api.telegram.org?

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



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 под ключ на одной VM: server, PostgreSQL 16 + TimescaleDB, Nginx, Agent 2
- Заводим все серверы, сеть, ИБП и 1С: шаблоны из коробки + наш шаблон 1С, автарегистрация по HostMetadata
- Настраиваем алерты в Telegram с эскалацией по severity и резервным SOCKS5-каналом
- Собираем два дашборда: NOC-экран дежурным и Executive-отчёт руководителю (аптайм, MTTR)
- Проводим тюнинг под нагрузку: кэши, компрессия, retention, зависимости триггеров, maintenance-окна

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

Техническая база

- 01** What's new in Zabbix 7.0.0 (async pollers, item-level timeouts) (zabbix.com — 7.0)
- 02** Zabbix server configuration (zabbix_server.conf) (zabbix.com — 7.0)
- 03** TimescaleDB setup и compression (Housekeeping) (zabbix.com — 7.0)
- 04** Zabbix agent 2 — MSSQL loadable plugin (zabbix.com — 7.0)
- 05** Zabbix Life Cycle and Release Policy (7.0 LTS) (zabbix.com — 7.0)
- 06** Наш шаблон 1С:Предприятие (rphost/рас/бизнес-KPI) (itfresh.ru — 2026)

