Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Мониторинг Debian-сервера на Prometheus

Как мы собираем стек Prometheus + node_exporter + Alertmanager + Grafana на Debian 13

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Сервер без метрик — чёрный ящик: о забитом /var, деградировавшем RAID-массиве, OOM-killer и упавшей службе узнают, когда встала 1С или почта. Мы закрываем это стеком Prometheus + node_exporter + Alertmanager + Grafana на отдельном Debian-хосте: сбор метрик каждые 15-60 с, правила с выдержкой for, алерты в Telegram и дашборды с историей за 15 суток.

Почему это важно бизнесу

- Простой 1С или почты на два-три часа обходится дороже, чем весь стек: он собирается из пакетов Debian и не требует лицензий.
- Деградация RAID и заполнение раздела видны за дни до отказа — плановая замена диска вместо срочного восстановления из бэкапа.
- Инженер приходит с готовой картиной: какой хост, какая метрика, с какого времени, вместо часа выяснений «что вообще случилось».
- Метрики за 15 суток дают факты для разговора с провайдером, хостером и подрядчиком 1С — вместо «нам кажется, что тормозит».
- Ложные алерты убивают доверие к системе: мы стартуем с 10-15 правил с выдержкой for, а не со 150 правил в первый день.

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

3.13 LTS

LTS-ветка Prometheus: релиз 01.07.2026, поддержка до 31.07.2027 — ставим её, а не latest

prometheus.io, Release cycle

2.53.3

Версия prometheus в пакетах Debian 13; апстрим ветку 2.53 закрыл 31.07.2025

packages.debian.org, trixie

15 с

Наш scrape_interval для боевых узлов вместо дефолтных 1m в конфиге Prometheus

наш стандарт (дефолт по докам — 1m)

15d

Дефолт --storage.tsdb.retention.time; годовую историю уводим через remote_write

по докам Prometheus, раздел Storage

1-2 байта

Средний размер сэмпла в TSDB — по нему считаем диск под retention

prometheus.io, Storage

0.28.1

Версия Alertmanager в Debian 13: telegram_configs из коробки, без бота-прослойки

packages.debian.org, trixie

РЕАЛИЗАЦИЯ 1 ИЗ 3

Агенты node_exporter на всём парке Debian

Что настраиваем

Все Debian- и Ubuntu-узлы: серверы 1С, файловые, почтовые, гипервизоры; агент слушает 9100 на внутреннем интерфейсе

Как мы это делаем

1. apt install prometheus-node-exporter (1.9.0 в trixie) или апстрим-бинарь 1.12.1 в /usr/local/bin с юнитом systemd — версию держим единой по парку
2. Включаем --collector.systemd и --collector.processes: оба выключены по умолчанию, без них половина панелей дашборда 1860 пустая
3. Слушаем только внутренний адрес: --web.listen-address=10.0.0.5:9100, снаружи 9100 закрыт правилом nftables
4. Задаём --web.config.file: tls_server_config с cert_file/key_file и min_version: TLS12 плюс basic_auth_users с bcrypt-хэшем
5. Свои метрики отдаём через textfile-коллектор: --collector.textfile.directory, файлы *.prom пишет наш скрипт по cron

РЕЗУЛЬТАТ

Одинаковый набор метрик на всех узлах: CPU, память, диски и файловые системы, состояние systemd-юнитов и mdadm-массивов. Новый сервер попадает под наблюдение за минуты: роль Ansible ставит пакет, кладёт web-config и дописывает JSON в /etc/prometheus/targets.

КЛЮЧЕВОЙ НЮАНС

Порт 9100 без ограничений — открытая инвентаризация хоста: слушаем только внутренний интерфейс и сразу включаем TLS и basic_auth через --web.config.file, а не «потом закроем».

РЕАЛИЗАЦИЯ 2 ИЗ 3

Сервер Prometheus, правила и алерты в Telegram

Что настраиваем

Отдельный Debian-хост: prometheus, prometheus-alertmanager, prometheus-blackbox-exporter 0.26.0; не рядом с боевыми сервисами

Как мы это делаем

1. global: scrape_interval 15s и evaluation_interval 15s вместо дефолта 1m, external_labels с площадкой — метки уезжают в Alertmanager
2. Цели не руками: file_sd_configs с /etc/prometheus/targets/*.json из Ansible, метки env, role, team на каждом job
3. Правила в /etc/prometheus/rules.d/*.yml: NodeDown up == 0 for 2m (critical), DiskSpaceLow < 10% for 10m, RaidDegraded по node_md_disks for 5m
4. Перед перезагрузкой конфига: promtool check config и promtool check rules, затем systemctl reload prometheus — битый YAML не роняет сервис
5. Alertmanager 0.28.1: group_by [alertname, instance], group_wait 30s, group_interval 5m, repeat_interval 4h; critical — свой telegram_configs с repeat_interval 1h

РЕЗУЛЬТАТ

Дежурный получает сгруппированное сообщение, а не поток из сорока однотипных строк: критичное идёт отдельным маршрутом с укороченным repeat_interval, остальное повторяется раз в 4 часа до устранения, а восстановление приходит resolved-уведомлением.

КЛЮЧЕВОЙ НЮАНС

Alertmanager умеет Telegram нативно — telegram_configs с bot_token_file, chat_id и parse_mode; промежуточный webhook-бот не нужен, это минус один сервис, который сам может упасть.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Grafana, внешние probe и хранение истории

Что настраиваем

Grafana 13.2 и blackbox_exporter на 9115 на хосте мониторинга: HTTP, ICMP, TCP и DNS-проверки снаружи периметра

Как мы это делаем

1. Job blackbox-http: metrics_path /probe, module http_2xx, relabel_configs подменяет __address__ на 127.0.0.1:9115
2. 10-15 внешних целей: сайты, MX и 25-й порт, DNS-серверы, RDP-шлюз; алерт probe_success == 0 for 3m
3. Grafana: datasource http://localhost:9090 и дашборды из /etc/grafana/provisioning, allowUiUpdates: false, foldersFromFilesStructure: true
4. Импортируем 1860 (Node Exporter Full) и 7587 (Prometheus Blackbox Exporter), правим под парк и коммитим JSON в Git
5. Retention 15 суток локально, длительная история — remote_write наружу; диск считаем как retention × samples/s × 1-2 байта

РЕЗУЛЬТАТ

Руководитель видит одну страницу-светофор по площадке, инженер — детализацию до диска и файловой системы. Стенд мониторинга переживает переезд: конфиг и дашборды разворачиваются из репозитория за одну раскатку, а не восстанавливаются руками по скриншотам.

КЛЮЧЕВОЙ НЮАНС

Сервер мониторинга не ставим рядом с боевыми сервисами: упавший гипервизор не должен уносить то, что о нём сообщает. Бэкапим /etc/prometheus, /var/lib/prometheus и /var/lib/grafana.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

✕ Пакет из apt считают свежим Prometheus

В Debian 13 это 2.53.3+ds1-2, а апстрим-поддержка ветки 2.53 закончилась 31.07.2025. Где нужны фиксы — ставим бинарь 3.13 LTS.

✕ Апгрейд 2.x → 3.x «в лоб»

В 3.0 скрейп падает без корректного Content-Type цели, диапазоны стали left-open, le/quantile нормализуются во float — правила гоняем promtool.

✕ Правила без выдержки for

Правило без for срабатывает на одном пике скрейпа. Ставим for 2m для NodeDown и 10m для дисков и памяти — дребезг уходит, аварии остаются.

✕ 150 правил в первый день

Alert fatigue: дежурный перестаёт читать. Стартуем с 10-15 правил (доступность, диск, память, RAID, probe_success), остальное — по инцидентам.

✕ static_configs на 50+ хостов

Правка руками = забытые узлы. Переходим на file_sd_configs с JSON из Ansible: метки env, role, team проставляет генератор, а не человек.

✕ Мониторинг на боевом гипервизоре

Падает прод — падает и то, что должно об этом сообщить. Prometheus, Alertmanager и Grafana выносим на отдельный узел вне боевого кластера.

✕ Дашборды живут только в базе Grafana

После переустановки восстанавливать нечего. Держим JSON в Git и раскатываем провижинингом с allowUiUpdates: false, чтобы правки не расходились.

✕ Диск под TSDB подбирают на глаз

Считаем по формуле из доков: retention × сэмплов/с × 1-2 байта, и ставим --storage.tsdb.retention.size как жёсткий предохранитель.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- node_exporter на всех Debian-узлах, порт 9100 только на внутреннем интерфейсе
- Prometheus и Alertmanager на отдельном хосте, scrape_interval 15-60 с
- 10-15 базовых правил: NodeDown, диск, память, RAID, probe_success
- Telegram-канал дежурных и promtool check rules перед каждым reload

НОРМАЛЬНО

- file_sd_configs с targets из Ansible, метки env, role, team на каждом job
- blackbox_exporter: 10-15 внешних probe по HTTP, DNS, SMTP и RDP
- Grafana за HTTPS, дашборды из /etc/grafana/provisioning, а не из БД
- Бэкап /etc/prometheus, /var/lib/prometheus и /var/lib/grafana по расписанию

ХОРОШО

- Prometheus 3.13 LTS, единая версия node_exporter 1.12.1 по всему парку
- TLS и basic_auth на экспортерах через --web.config.file
- remote_write во внешнее хранилище: год истории при 15 сутках локально
- Правила и дашборды в Git, раскатка из репозитория, allowUiUpdates: false

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Есть ли отдельный хост мониторинга, не совмещённый с боевыми 1С, почтой и гипервизором?
- [ ] Закрыт ли 9100 снаружи: listen-address на внутреннем интерфейсе или правило nftables?
- [ ] Знаете ли вы версию Prometheus и поддерживается ли она апстримом как LTS-ветка?
- [ ] У всех ли правил задан for, чтобы одиночный пик не поднимал дежурного среди ночи?
- [ ] Приходят ли алерты в канал, который дежурный реально читает, и проверялась ли доставка?
- [ ] Добавляются ли новые серверы в мониторинг автоматически через file_sd_configs?
- [ ] Есть ли внешние probe на сайт, почту и DNS, а не только метрики изнутри периметра?
- [ ] Лежат ли дашборды и правила в Git и поднимется ли Grafana с нуля за час?
- [ ] Посчитан ли диск под TSDB и включён ли --storage.tsdb.retention.size?
- [ ] Резервируются ли /etc/prometheus, /var/lib/prometheus и /var/lib/grafana?

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 Prometheus, node_exporter, Alertmanager и Grafana на Debian под ключ за один рабочий день
- Раскатываем агенты плейбуком Ansible на весь парк и заводим file_sd-инвентарь с метками env и role
- Пишем правила под ваш парк, настраиваем маршруты Alertmanager и оповещения дежурных в Telegram
- Собираем дашборды в Git, включаем провижининг Grafana, HTTPS и аутентификацию с первого дня
- Берём мониторинг на сопровождение: обновление версий, ревизия порогов, разбор ложных срабатываний

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 Prometheus: Configuration — global, scrape_configs, file_sd_configs (prometheus.io — 3.13 LTS)

02 Prometheus: Storage — retention, размер сэмпла, расчёт диска (prometheus.io — 3.13)

03 Prometheus: Release cycle — таблица LTS и сроки поддержки (prometheus.io — 2026)

04 Alertmanager: Configuration — route, group_wait, telegram_config (prometheus.io — 0.28.1)

05 Prometheus: Migration guide — несовместимости 3.x с веткой 2.x (prometheus.io — 3.0)

06 Exporter Toolkit: web-configuration — TLS и basic_auth (github.com — 2026)

07 Grafana: Provisioning — datasources, dashboards, allowUiUpdates (grafana.com — 13.2)

08 Debian: пакеты prometheus, node-exporter, alertmanager в trixie (packages.debian.org — 13 trixie)

Основано на официальной документации продуктов и нашей практике внедрения.