



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Angie вместо nginx: миграция без даунтайма

Наш регламент переезда с nginx: встроенный ACME,
метрики Prometheus, upstream без reload

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

У клиента веб-периметр держится на nginx: сайт, личный кабинет, публикация 1С, почтовый веб. Пакеты приходят из зарубежного репозитория, сертификаты продлевает certbot из cron, а мониторинг сводится к stub_status. Для тендера по 223-ФЗ нужно ПО из реестра, а при аварии никто не скажет, какой upstream отдаёт 502 и сколько это длится.

Почему это важно бизнесу

- Отсутствие ПО из реестра Минцифры закрывает участие в закупках по 223-ФЗ и 44-ФЗ — заявку отклоняют на формальном основании.
- Просроченный сертификат роняет сайт и приём оплат: браузер показывает предупреждение, конверсия падает в тот же день.
- Без метрик по upstream авария диагностируется по звонкам клиентов, а не по графику — MTTR растёт с минут до часов.
- Обновление безопасности приходится ждать от зарубежного мейнтейнера, окно уязвимости открыто неопределённо долго.
- Каждый деплой через reload рвёт keep-alive и даёт всплеск 502 у пользователей личного кабинета.



Ключевые параметры реализации

1.12.1

Актуальная ветка Angie на клиентских фронтах, релиз 17.07.2026

[angie.software](#) / История версий

30d

Дефолт `renew_before_expiry` в `http_acme`: за столько до истечения идёт перевыпуск

[angie.software](#) / `http_acme`

1m

Зона разделяемой памяти upstream в нашем шаблоне: без `zone` нет ни метрик, ни `resolve`

Наш стандарт шаблона upstream

5s / 1 / 1

Дефолты `upstream_probe` (PRO): `interval`, `fails`, `passes`; в шаблоне ужесточаем до 3/3

[angie.software](#) / `http_upstream_probe` (PRO)

№ 17604

Реестровая запись Минцифры на Angie PRO — основание для закупок по 223-ФЗ

Реестр российского ПО, запись № 17604

0 сек

Целевой даунтайм миграции: ноду выводим из ротации, переключаем, возвращаем

Наш стандарт миграции



Переезд фронта с nginx на Angie за одно техокно

Что настраиваем

Веб-периметр торговой компании, 40 РМ: 6 сайтов и личный кабинет на двух nginx-нодах за балансировщиком, Debian 12

Как мы это делаем

- 1 Снимаем эталон: `nginx -T > /root/nginx-full.conf`, архивируем `/etc/nginx` и `/var/log/nginx`, фиксируем вывод `nginx -V` со списком модулей.
- 2 Подключаем официальный репозиторий Angie с GPG-ключом, ставим пакет `angie` и нужные динамические: `angie-module-njs`, `angie-module-xslt`.
- 3 `rsync -a --no-links /etc/nginx/ /etc/angie/`, переименовываем `nginx.conf` в `angie.conf`, правим `include`, `pid /run/angie.pid`, логи `/var/log/angie`.
- 4 `grep -rn nginx /etc/angie` — вычищаем хвосты, `angie -t`, снимаем ноду с балансировщика, стартуем `angie`, смоук-тест по 5 URL.
- 5 Возвращаем ноду в ротацию, вторую переводим тем же чек-листом; `nginx` гасим: `systemctl disable --now nginx` и `apt-mark hold nginx`.

РЕЗУЛЬТАТ

Обе ноды переведены за одно окно, даунтайм нулевой — трафик всё время держала оставшаяся нода. Конфиг заработал без правки директив, кроме путей и имени главного файла. Заказчик получил веб-сервер из реестра и прошёл формальный отбор по 223-ФЗ.

КЛЮЧЕВОЙ НЮАНС

Опасен не конфиг, а окружение: `logrotate`, `fail2ban`, скрипты бэкапа и `Zabbix`-агент всё ещё смотрят в `/etc/nginx` и `/var/log/nginx`. Правим их в том же окне, а не после аварии.



Убираем certbot: встроенный ACME и Prometheus-метрики

Что настраиваем

Юридические услуги, 25 PM: 9 доменов, сертификаты продлевал cron с certbot, мониторинг — одна строка stub_status

Как мы это делаем

- 1 В блоке http объявляем resolver с valid=300s: без резолвера в том же контексте acme_client не стартует — это требование модуля.
- 2 acme_client le <URI ACME-директории УЦ> key_type=ecdsa renew_before_expiry=30d; в нужных server — acme le, состояние клиента в acme_client_path.
- 3 Валидацию оставляем на HTTP-01 (acme_http_port 80); для wildcard — challenge=dns, acme_dns_port 53 и делегирование NS для _acme-challenge.
- 4 Подключаем include prometheus_all.conf, отдаём location =/p8s с директивой prometheus all; вешаем status_zone на server и location, zone на upstream.
- 5 Prometheus скрейпит /p8s на 127.0.0.1; ставим angie-console-light на /console/ с вложенным api /status/ и allow 127.0.0.1, алерты по 5xx и latency.

РЕЗУЛЬТАТ

Сертификаты перевыпускает сам веб-сервер, cron-скриптов и certbot-хуков больше нет. Статус выпуска и время следующей попытки видны в /status/http/acme_clients/. Появились графики по каждому server_zone и upstream — 502 стало видно до звонка клиента.

КЛЮЧЕВОЙ НЮАНС

Экспортёр не выдаёт ничего без разделяемых зон: нет status_zone и zone — нет метрик. При ошибке ACME повтор идёт не сразу: дефолт retry_after_error 2h. Сам /p8s держим на 127.0.0.1.



Динамические upstream: Docker-метки и PRO-проверки

Что настраиваем

Производство, 30 PM: приложение в контейнерах на одном Docker-хосте, деплой раньше требовал reload и давал всплеск 502

Как мы это делаем

- 1 Пишем `docker_endpoint` `unix:/var/run/docker.sock` в `http`; `upstream` обязательно с `zone app 1m` — без разделяемой зоны модуль контейнер не подхватит.
- 2 На контейнер вешаем метку `angie.http.upstreams.app.port=8080`; при нескольких сетях — `angie.network`, вес — `angie.http.upstreams.app.weight`.
- 3 Сокет Docker отдаём `read-only` и только локально; `docker_max_object_size` поднимаем с дефолтных 64k, если объектов много.
- 4 Где нужны активные проверки, берём PRO: `upstream_probe health uri=/healthz interval=5s fails=3 passes=3 essential` — дефолты `fails/passes` равны 1.
- 5 В Community состав пиров держим через DNS: `server app.svc service=http resolve` плюс `zone` и `resolver` — тоже без reload.

РЕЗУЛЬТАТ

Старт и остановка контейнера отражаются в апстриме в реальном времени, reload при выкатке не нужен — keep-alive не рвутся. Всплески 502 в момент деплоя ушли, состав пиров всегда виден в `/status/http/upstreams/app`.

КЛЮЧЕВОЙ НЮАНС

Запись в `/config/http/upstreams/` — функция PRO, в Community API только на чтение. Если процесс выкатки завязан на изменение апстрима без reload, редакцию надо закладывать в бюджет заранее.



Подводные камни

✗ Оставили пакет nginx активным

Системный dist-upgrade поднимает nginx обратно, и два сервера дерутся за 80/443. Делаем `systemctl disable --now nginx` и `apt-mark hold nginx`.

✗ Забыли переименовать nginx.conf

Главный файл в `/etc/angie` называется `angie.conf`. Скопированный `nginx.conf` служба просто не читает, а `angie -t` проверяет уже новый путь.

✗ Нет resolver — ACME молча не работает

Модуль `http_acme` требует `resolver` в том же контексте, что и `acme_client`. Без него клиент не поднимается и сертификат не запрашивается вовсе.

✗ Метрики без status_zone и zone

Экспортёр отдаёт пустой набор: значения берутся из разделяемых зон. Нет `zone` на `upstream` и `status_zone` на `server` — данных не будет.

✗ Экспортёр торчит наружу

`/p8s` и `/console/` раскрывают структуру бэкендов и версии. Слушаем только `127.0.0.1` либо закрываем `allow/deny` и клиентским сертификатом.

✗ Сторонние модули не пересобраны

Динамические модули от nginx не грузятся: другая сборка. Ставим пакеты `angie-module-*` и правим `load_module` на `modules/<имя>.so`.

✗ Мониторинг смотрит в /var/log/nginx

`Zabbix`, `logrotate`, `fail2ban` и скрипты бэкапа читают старые пути и молчат об авариях. Переводим их на `/var/log/angie` в том же окне.

✗ Ждали writable API в Community

Изменение апстрима через `/config/` доступно только в PRO. В Community остаются `DNS-resolve`, `Docker`-метки или обычный `reload`.

Как правильно

МИНИМУМ

- Angie из официального репозитория, nginx отключён и зафиксирован apt-mark hold
- Конфиг в /etc/angie/angie.conf, обязательная проверка angie -t до рестарта
- Встроенный ACME с resolver вместо certbot, key_type=ecdsa, HTTP-01 по умолчанию
- Бэкап nginx -T и архив /etc/nginx до начала работ

НОРМАЛЬНО

- status_zone на server и location, zone на каждом upstream
- prometheus_all.conf, location =/p8s с prometheus all, Prometheus и Grafana
- angie-console-light на /console/ с api /status/ и доступом только из LAN
- Миграция понодно за балансировщиком, смоук-тест по 5 ключевым URL

ХОРОШО

- Angie PRO: upstream_probe interval=5s fails=3 passes=3 на всех апстримах
- Docker-метки angie.http.upstreams.* — выкатка без reload
- metric_zone/metric_complex_zone (OSS с 1.11) под свои SLA: гистограммы и средние
- Алерты на 5xx, latency апстрима и статус сертификата из /status/http/acme_clients/



Чек-лист самопроверки

☐ Есть ли свежий вывод `nginx -T` и архив `/etc/nginx` перед началом миграции?

☐ Отключён ли пакет `nginx` и зафиксирован ли он от возврата при обновлении системы?

☐ Переведены ли все `include`, `pid` и пути логов на `/etc/angie` и `/var/log/angie`?

☐ Проходит ли `angie -t` на боевом конфиге до перезапуска службы?

☐ Настроен ли `resolver` в том же контексте, что и `acme_client`, иначе ACME не стартует?

☐ Заданы ли `status_zone` на серверах и `zone` на апстримах — иначе метрик не будет?

☐ Закрыты ли `/p8s` и `/console/` от внешнего доступа (127.0.0.1 либо `allow/deny`)?

☐ Заменены ли сторонние модули на пакеты `angie-module-*` с `load_module modules/<имя>.so`?

☐ Обновлено ли пути в `Zabbix`, `logrotate`, `fail2ban` и скриптах резервного копирования?

☐ Знаем ли мы, какие нужные функции требуют редакции PRO, и заложены ли они в бюджет?

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).

- Аудит текущего nginx: nginx -T, список модулей, карта сертификатов и зависимостей мониторинга.
- Миграция понодно за одно техокно с выводом из балансировщика — целевой даунтайм ноль.
- Перевод сертификатов с certbot на встроенный ACME Angie, включая DNS-01 для wildcard.
- Метрики: status_zone, zone, prometheus_all.conf, Prometheus и дашборд Grafana с алертами.
- Разбор Community против PRO по вашим сценариям: /config API, upstream_probe, кластер, поддержка.

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** Angie: установка из пакетов, поддерживаемые ОС и модули `angie-module-*` (`angie.software` — 1.12.1)
- 02** Миграция с `nginx` на Angie: `/etc/angie`, `angie.conf`, пути логов и `pid` (`angie.software` — 1.12.1)
- 03** Модуль `http_acme`: `acme_client`, `renew_before_expiry`, `acme_http_port` (`angie.software` — 1.12.1)
- 04** Модуль `http_prometheus` и шаблон `prometheus_all.conf` (`angie.software` — 1.12.1)
- 05** Модуль `http_api`: дерево `/status` и раздел `/config` (PRO) (`angie.software` — 1.12.1)
- 06** Модуль `http_docker`: `docker_endpoint` и метки `angie.http.upstreams.*` (`angie.software` — 1.12.1)
- 07** Angie PRO: `upstream_probe` — активные проверки апстримов (`angie.software` — 1.12.1)
- 08** Модуль `http_metric`: `metric_zone` и `metric_complex_zone` (`angie.software` — 1.12.1)
- 09** Шаблон ITfresh: чек-лист миграции веб-фронта (внутренний — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

