



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Публикация 1С через Apache: веб-клиент без терминального сервера

Как мы публикуем базы: Apache 2.4 + wsap24, default.vrd,
HTTPS, пулы сеансов, HTTP-сервисы

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Филиалам, удалёнке и подрядчикам нужен доступ к 1С, а терминальный сервер под это — лишние лицензии RDS и его обслуживание. Мы публикуем базы через Apache: веб-клиент работает в браузере без установки платформы. Но публикация «по умолчанию» — это открытый HTTP наружу, лишние сервисы в интернете и сервер, который падает под 20 пользователями из-за неверного MPM и нехватки RAM.

Почему это важно бизнесу

- Веб-клиент экономит на лицензиях RDS и терминальном сервере: сотруднику нужен только браузер
- Небезопасная публикация 1С наружу — прямой путь к утечке базы: клиенты, обороты, зарплаты
- Падение публикации останавливает выписку документов в филиалах — простой продаж идёт на часы
- HTTP-сервисы связывают 1С с сайтом и CRM: без них обмен ведут руками, с ошибками и задержкой в день

Ключевые параметры реализации

8.3.27

Ветка платформы на стендах;
модуль wsap24 ставим из пакета
1c-enterprise-8.3.27-ws той же
сбор...

по докам 1C:Предприятие 8.3

prefork

MPM Apache на Linux: модуль 1C не
потокбезопасен, event и worker
недопустимы

по докам 1C + Apache 2.4

poolSize 10

Пул сеансов ws/HTTP-сервисов по
умолчанию (poolTimeout 5 с); при
исчерпании вызов получает 406...

default.vrd, доки 1C

200–500 МБ

RAM на сеанс веб-клиента; на 20
пользователей закладываем 16 ГБ
под процессы Apache

наш стандарт

90 дней

Срок сертификата Let's Encrypt;
авто-продление certbot.timer,
deploy-hook перечитывает Apache

certbot.eff.org

1 год

HSTS max-age=31536000 на
публикации: браузер сотрудника
больше не пойдёт на голый HTTP

наш стандарт



Развёртывание веб-публикации на Linux-узле

Что настраиваем

Отдельная VM веб-узла Apache 2.4 (Ubuntu 22.04) рядом с кластером 1C; файловые и клиент-серверные базы

Как мы это делаем

- 1 Ставим apache2 и ws-пакет единого дистрибутива 1c-enterprise-8.3.27-ws той же сборки, что кластер; включаем ssl, headers, rewrite (a2enmod)
- 2 Подключаем модуль: LoadModule _1cws_module /opt/1cv8/x86_64/<версия>/wsap24.so в mods-available/1c.load, затем a2enmod 1c
- 3 Публикуем: webinst -apache24 -wsdir erp -dir /var/www/1c/erp -connstr "Srvr=srv-1c;Ref=erp" -confpath /etc/apache2/1c.conf — все базы в один conf
- 4 Переключаем MPM: a2dismod mpm_event, a2enmod mpm_prefork; MaxRequestWorkers считаем от RAM, а не берём дефолт
- 5 Права: chown -R www-data:www-data /var/www/1c; проверка apache2ctl -M | grep 1cws и тестовый вход веб-клиентом

РЕЗУЛЬТАТ

Пользователь открывает <https://server/erp> в Chrome/Firefox/Edge без установки платформы. Публикация воспроизводима: один conf-файл на все базы, добавление новой базы — одна команда webinst, а не ручная правка httpd.conf.

КЛЮЧЕВОЙ НЮАНС

Версии платформы на веб-узле и в кластере должны совпадать до сборки: чужой wsap24 даёт «Сервер 1C:Предприятия не обнаружен». Путь к модулю содержит номер версии — держим его в единственном файле 1c.load.



HTTPS, контроль доступа и тюнинг default.vrd

Что настраиваем

VirtualHost :443 на веб-узле; сертификат Let's Encrypt, IP-фильтры и параметры пула сеансов публикации

Как мы это делаем

- 1 certbot --apache -d 1c.company.ru; порт 80 оставляем только под редирект 301 на HTTPS и ACME-челлендж
- 2 В <Location /erp> прописываем Require ip: офисные подсети и статические IP филиалов; остальным — 403
- 3 Заголовки: Strict-Transport-Security max-age=31536000, X-Frame-Options DENY, X-Content-Type-Options nosniff
- 4 В default.vrd задаём poolSize/poolTimeout под профиль нагрузки — конфигуратор эти атрибуты не пишет, правим файл руками
- 5 httpServices publishByDefault="false"; нужные сервисы включаем поимённо (enable="true", свой rootUrl)

РЕЗУЛЬТАТ

Наружу торчит только 443 с валидным сертификатом и IP-фильтром. Интеграции не съедают сеансы веб-клиента: пул ограничен, при исчерпании внешняя система получает 406 Not Acceptable и уходит в ретрай, а не вешает базу.

КЛЮЧЕВОЙ НЮАНС

Переопубликация из конфигулятора перезаписывает default.vrd и стирает ручные атрибуты (poolSize, openId, точечные сервисы). После первичной публикации все правки — только в vrd-файле, сам файл держим в git.



HTTP-сервисы для интеграций и регламент эксплуатации

Что настраиваем

REST-обмен 1С с сайтом/CRM на той же публикации; мониторинг и обновление платформы без потери публикации

Как мы это делаем

- 1 Под каждую интеграцию заводим отдельного пользователя 1С с минимальными правами и запретом интерактивного входа; Basic-аутентификация только поверх HTTPS
- 2 Для сервисов ставим `reuseSessions="autouse"` и короткий `sessionMaxAge`, чтобы не плодить сеанс на каждый вызов
- 3 Мониторим: внешняя HTTP-проба публикации, `telnet srv-1c 1541` с веб-узла, `error.log` Apache, свободная RAM (`free -h`)
- 4 Обновление платформы: ставим новую версию рядом, меняем путь в `1c.load`, `restart apache2`; `default.vrd` не трогаем — он от версии не зависит

РЕЗУЛЬТАТ

Сайт и CRM получают данные из 1С онлайн вместо ручных выгрузок. Обновление платформы занимает минуты и не ломает публикацию: меняется одна строка — путь к модулю в `1c.load`.

КЛЮЧЕВОЙ НЮАНС

Скомпрометированный интеграционный логин с полными правами — это вся база наружу. Права пользователя HTTP-сервиса режем до нужных объектов и проверяем, что вход в веб-клиент под ним невозможен.



Подводные камни

✗ MPM event по умолчанию

Ubuntu ставит Apache с `mpm_event`, а `wsap24.so` не потокобезопасен: случайные падения под нагрузкой. Всегда переключаем на `mpm_prefork`.

✗ Переопубликация затирает `vrld`

Конфигуратор пишет `default.vrld` заново: пропадают `poolSize`, `openid`, точечные сервисы. После выката правим `vrld` только руками и храним в `git`.

✗ Путь к модулю с номером версии

После обновления платформы путь `/opt/1cv8/.../wsap24.so` меняется — Apache не стартует. Путь держим в одном файле `1c.load` и меняем по регламенту.

✗ `publishByDefault="true"`

Наружу выезжают все HTTP- и веб-сервисы конфигурации, включая служебные. Ставим `false` и включаем сервисы поимённо с отдельным `rootUrl`.

✗ Голый HTTP наружу

Логин и пароль 1С уходят по сети открытым текстом. HTTP допустим только в изолированном сегменте; наружу — только 443, HSTS и IP-фильтр.

✗ `MaxRequestWorkers` без расчёта RAM

Каждый процесс `prefork` ест 200–500 МБ; лишние воркеры загоняют сервер в `swap` — веб-клиент «зависает». Лимит считаем от фактической памяти.

✗ SELinux на RHEL-семействе

403 при корректном конфиге: нужен `setsebool -P httpd_can_network_connect on`. Диагностируем по `audit.log`, а не отключаем SELinux целиком.

✗ Nginx вместо Apache

Модуль веб-расширения 1С работает только с Apache/IIS. Nginx ставим лишь фронтом (TLS, rate-limit), бэкендом остаётся Apache с `wsap24`.



Как правильно

МИНИМУМ

- Apache 2.4 + mpm_prefork, модуль wsap24 той же сборки, что кластер 1C
- HTTPS с Let's Encrypt, порт 80 — только редирект на 443
- publishByDefault="false" и доступ по IP-whitelist в <Location>

НОРМАЛЬНО

- default.vrd в git, poolSize/poolTimeout подобраны под нагрузку
- Отдельный пользователь 1C на каждую интеграцию, без интерактивного входа
- Мониторинг публикации: внешняя HTTP-проба, error.log, свободная RAM
- Регламент обновления платформы: правка 1c.load + restart apache2

ХОРОШО

- Nginx/HAProxy фронтом: терминация TLS, rate-limit, fail2ban по логам
- Веб-узел на отдельной VM, изолирован от сервера 1C и СУБД
- 2FA/OpenID на входе веб-клиента через reverse проху
- Нагрузочная проверка: пиковое число сеансов × RAM на процесс prefork



Чек-лист самопроверки

☐ Публикация доступна только по HTTPS, а порт 80 отдаёт лишь редирект?

☐ MPM Apache на веб-узле — prefork, а не event/worker?

☐ Версия модуля wsap24 совпадает со сборкой кластера 1С?

☐ В default.vrd отключена публикация сервисов по умолчанию (publishByDefault="false")?

☐ У каждой интеграции отдельный пользователь 1С с минимальными правами?

☐ MaxRequestWorkers рассчитан от фактической RAM (200–500 МБ на сеанс)?

☐ Продление сертификата автоматизировано и проверено (certbot renew --dry-run)?

☐ Есть внешняя проба доступности публикации с алертом дежурному?

☐ default.vrd лежит в системе контроля версий и переживёт переопубликацию?

☐ Прописан регламент обновления платформы без потери публикации?

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 публикацию 1С под ключ: Apache, HTTPS, default.vrd, IP-фильтры — за один рабочий день
- Проектируем REST-интеграции 1С с сайтом и CRM: HTTP-сервисы, отдельные учётки, лимиты пула сеансов
- Переносим публикации при обновлении платформы и миграции серверов без простоя пользователей
- Ставим мониторинг веб-публикации: доступность, сеансы, RAM, срок сертификата — с алертами в Telegram
- Проводим аудит существующей публикации: безопасность, производительность, отказоустойчивость

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** Руководство администратора, гл. «Настройка веб-серверов» (its.1c.ru — 8.3.27)
- 02** Руководство администратора, прил. «Описание и структура файла default.vrd»... (its.1c.ru — 8.3.27)
- 03** Технологии интеграции 1С:Предприятия 8.3: повторное использование сеансов... (its.1c.ru — 8.3.27)
- 04** Apache HTTP Server: mpm_prefork, mod_ssl, mod_headers (httpd.apache.org — 2.4)
- 05** Certbot: выпуск и авто-продление сертификатов для Apache (certbot.eff.org — 2026)
- 06** Шаблон ITfresh: vrd-профиль публикации и чек-лист выката (наш стандарт — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

