Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Keycloak 26: SSO и управление доступом в офисе

Как мы разворачиваем IdP на Debian, федерацию с Active Directory и OIDC-клиенты

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

У компании на 30-50 мест уже 5-8 веб-сервисов: GitLab, Grafana, Nextcloud, Zabbix, портал, 1С. В каждом свои учётки, увольнение закрывается вручную и не везде, 2FA в лучшем случае в одном сервисе, аудит входов не собрать. Мы решаем это одной точкой аутентификации: Keycloak как IdP поверх AD, доступ выдаётся и отзывается в домене, приложения ходят по OIDC/SAML.

Почему это важно бизнесу

- Уволенный сотрудник теряет доступ ко всем сервисам за минуты, а не за недели ручной чистки
- Пароль один и доменный: падает поток заявок «сбросьте мне доступ» в поддержку
- 2FA включается централизованно на realm, а не докупается и настраивается в каждом сервисе
- Аудит входов и активных сессий в одной консоли — закрывает вопросы проверок и расследований
- Новый сервис подключается за час созданием OIDC-клиента, без ведения ещё одной базы учёток

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

26.7.x

ветка Keycloak: hostname v2, транспорт кластера jdbc-ping, OpenJDK 21

доки keycloak.org

17/21/25

поддерживаемые OpenJDK LTS для Keycloak 26; PostgreSQL — с 14.x по 18.x

доки keycloak.org

x2

минимум два узла с cache=ispn: обновление и перезагрузка без простоя SSO

доки keycloak.org

5 мин

Access Token Lifespan в realm corp: короткое окно для перехваченного токена

наш стандарт

30 мин/10 ч

SSO Session Idle и SSO Session Max: баланс удобства и скорости отзыва доступа

наш стандарт

READ_ONLY

Edit Mode LDAP-федерации: источник истины — AD, Sync Registrations выключен

наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Узел Keycloak на Debian 12 за nginx

Что настраиваем

Отдельная VM 4 vCPU / 8 ГБ, внешний PostgreSQL 16, TLS на nginx, сам Keycloak слушает 127.0.0.1 и management-порт 9000

Как мы это делаем

1. Ставим OpenJDK 21 (в Debian 12 — из bookworm-backports), дистрибутив в /opt/keycloak, пользователь keycloak без shell, chown -R keycloak:keycloak
2. conf/keycloak.conf: db=postgres, db-url=jdbc:postgresql://db:5432/keycloak, hostname=https://sso.corp.ru, proxy-headers=xforwarded, health-enabled=true
3. Один раз bin/kc.sh build (health-enabled и metrics-enabled — build-time), дальше только start --optimized; unit systemd с Restart=always, After=postgresql
4. nginx: TLS 1.2/1.3, проброс X-Forwarded-For/Proto/Host/Port, наружу /realms и /resources, /admin — только из офисных подсетей и через VPN
5. Пробы /health/ready и /health/live и метрики /metrics снимаем на management-порту 9000 в Zabbix; http-max-queued-requests ограничиваем от отказа

РЕЗУЛЬТАТ

Сервис поднимается за один прогон плейбука и стартует за секунды, а не пересобирает конфигурацию на каждом рестарте. Балансировщик выводит узел из ротации по /health/ready, поэтому обновление версии проходит по одному узлу без разрыва пользовательских сессий.

КЛЮЧЕВОЙ НЮАНС

Опции hostname v1 удалены с 26.0 — конфиги старых инсталляций не стартуют. Внешние URL строятся из hostname и заголовков прокси: без proxy-headers=xforwarded ломаются redirect_uri.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Федерация с Active Directory и маппинг групп

Что настраиваем

Realm corp, LDAP-провайдер на ldaps://dc01.corp.ru:636 с резервным dc02, сервисная учётка только на чтение, группы из OU=Groups

Как мы это делаем

1. User Federation → Add LDAP provider: Vendor = Active Directory, Edit Mode = READ_ONLY, Sync Registrations = OFF — Keycloak не создаёт и не правит учётки в домене
2. Корневой сертификат УЦ домена — в truststore-paths в keycloak.conf, Use Truststore SPI = Always; в Connection URL два ldaps-URI через пробел
3. Periodic Full Sync раз в сутки ночью, Periodic Changed Users Sync каждые 900 секунд, Batch Size 1000 — каталог не выгребается целиком на каждом цикле
4. Mapper group-ldap-mapper: Groups DN OU=Groups, Membership LDAP Attribute member, User Groups Retrieve Strategy = LOAD_GROUPS_BY_MEMBER_ATTRIBUTE_RECURSIVELY
5. Проверяем отзыв: блокируем учётку в AD, в Sessions делаем Logout all sessions — доступ пропадает в пределах жизни access-токена (5 минут)

РЕЗУЛЬТАТ

Заведение и увольнение сотрудника остаются процессом в AD: доступ ко всем подключённым сервисам появляется и исчезает автоматически. Роли в приложениях выдаются членством в доменной группе, а не ручными правками в каждой панели администрирования.

КЛЮЧЕВОЙ НЮАНС

Аутентификация делегируется в AD при каждом входе, поэтому недоступный DC кладёт весь SSO. Мы прописываем два DC в Connection URL и мониторим не порт 636, а факт успешной синхронизации.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Подключение приложений: OIDC-клиенты, роли, 2FA

Что настраиваем

Grafana, GitLab, Nextcloud и внутренний портал как confidential-клиенты realm corp; роли считаются из доменных групп

Как мы это делаем

1. Клиент делаем confidential (Client authentication = On): Standard Flow включён, Direct Access Grants выключен, Redirect URIs и Web Origins — без звёздочек
2. В dedicated scope — mapper Group Membership, Full group path = On, claim groups; в Grafana его разбирает role_attribute_path по contains(groups[*], ...)
3. Client Secret храним в Битвардене, ротация раз в год; интеграциям вместо личных учёток выдаём Service Account с точечными ролями
4. 2FA: Required Action Configure OTP плюс conditional-флоу с условием user role по группе админов; TOTP 6 цифр, период 30 секунд; Brute force detection на realm
5. У каждого клиента задаём Backchannel logout URL и Backchannel logout session required = On — выход из Keycloak гасит сессию в приложении, а не только куку IdP

РЕЗУЛЬТАТ

Подключение очередного сервиса занимает час и не добавляет новой базы паролей. Пользователь логинится один раз за смену, администраторы получают вторым фактором TOTP, а выход из системы действительно завершает сессии во всех приложениях сразу.

КЛЮЧЕВОЙ НЮАНС

Wildcards в Valid Redirect URIs и Web Origins — главный источник дыр в OIDC. Каждый URL описываем явно, экспорт клиента (kc.sh export) держим в git, чтобы правки не терялись.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Keycloak выставлен в интернет напрямую**
  Админ-консоль и публичные эндпоинты живут на одном порту. Закрываем nginx с TLS: наружу /realms и /resources, /admin — только по ACL и VPN.

- ✕ **Не выставлен proxy-headers**
  Без proxy-headers=xforwarded сервер строит ссылки по внутреннему хосту и после логина выбрасывает на http или недоступный URL.

- ✕ **Старт без --optimized**
  Каждый запуск пересобирает конфигурацию и добавляет десятки секунд к рестарту. Делаем kc.sh build на деплое, сервис стартует только с --optimized.

- ✕ **Работа в realm master**
  В master держим только админов Keycloak. Сотрудники и приложения живут в corp: иначе компрометация одной учётки открывает управление всем сервером.

- ✕ **Edit Mode = WRITABLE у LDAP**
  Keycloak начинает писать атрибуты в AD и конфликтует с кадровым процессом. Держим READ_ONLY и Sync Registrations = OFF, изменения — только в домене.

- ✕ **Осталась встроенная dev-БД**
  По умолчанию db=dev-file (H2) — только для разработки. Ставим внешний PostgreSQL с отдельным пользователем и ежедневным pg_dump.

- ✕ **Нет бэкапа realm и ключей подписи**
  Потеря базы — это потеря ключей, которыми подписаны все токены. Бэкапим БД и держим kc.sh export --realm corp в git вместе с конфигурацией клиентов.

- ✕ **Один узел без кластера**
  Любое обновление гасит вход во все сервисы. Ставим два узла cache=ispn со стеком jdbc-ping, балансировщик снимает узел по /health/ready.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Отдельный PostgreSQL 14-18, TLS на nginx, /admin закрыт по IP и VPN
- Realm corp отдельно от master, админ master с длинным паролем и TOTP
- LDAP-федерация с AD в READ_ONLY, сервисная учётка только на чтение
- Ежедневный pg_dump базы keycloak и хранение копии вне узла

НОРМАЛЬНО

- Два узла cache=ispn со стеком jdbc-ping за балансировщиком по /health/ready
- Экспорт realm и клиентов через kc.sh export в git, деплой плейбуком
- 2FA для админов и внешних доступов, Brute force detection на realm
- metrics-enabled и event-metrics-user-enabled, в Zabbix — LOGIN_ERROR

ХОРОШО

- Backchannel logout у всех клиентов и короткий Access Token Lifespan 5 минут
- Conditional-флоу: 2FA по группе AD и по признаку входа извне периметра
- Отдельный realm для партнёров и подрядчиков, без доступа к внутренним клиентам
- Регламент ротации client secret и служебных учёток раз в год с проверкой

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Keycloak закрыт обратным прокси, а консоль /admin недоступна из интернета?
- [ ] Используется внешний PostgreSQL, а не встроенная dev-file (H2)?
- [ ] Сервис стартует с --optimized, а kc.sh build выполняется на этапе деплоя?
- [ ] В Connection URL LDAP указано минимум два контроллера домена и только ldaps?
- [ ] Edit Mode федерации READ_ONLY и Sync Registrations выключен?
- [ ] Есть ли второй узел cache=ispn и проверялось ли обновление без простоя?
- [ ] Проверяли ли фактический отзыв доступа после блокировки учётки в AD?
- [ ] У всех клиентов Valid Redirect URIs заданы точно, без wildcard?
- [ ] Включены 2FA для администраторов и Brute force detection на уровне realm?
- [ ] Есть бэкап БД и экспорт realm, восстановление проверено на стенде?

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 Keycloak под ключ: узел, PostgreSQL, nginx с TLS, systemd, мониторинг
- Подключаем Active Directory через LDAP-федерацию с маппингом групп и ролей
- Интегрируем ваши сервисы по OIDC и SAML: Grafana, GitLab, Nextcloud, порталы, 1С
- Настраиваем 2FA, политики сессий, брутфорс-защиту и аудит входов
- Делаем отказоустойчивость из двух узлов, бэкапы realm и регламент обновлений

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 Configuring Keycloak for production (keycloak.org — 26.7)

02 Configuring distributed caches: ispn, jdbc-ping (keycloak.org — 26.7)

03 Server Administration Guide: User Federation — LDAP (keycloak.org — 26.7)

04 Configuring the hostname (v2) и Upgrading Guide (keycloak.org — 26.x)

05 Supported Configurations: OpenJDK и PostgreSQL (keycloak.org — 26.7)

06 Monitoring user activities with event metrics (keycloak.org — 26.7)

07 Configure Keycloak OAuth2 authentication (grafana.com — 2026)

08 Наш шаблон развёртывания IdP и чек-лист приёмки (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.