Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Keycloak: SSO и централизованное управление доступом

Разворачиваем единый вход на Keycloak 26: AD-федерация, OIDC-клиенты, 2FA и отказоустойчивость

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Заказчик живёт с 8–12 внутренними сервисами (GitLab, Grafana, Wiki, CRM, servicedesk) и отдельным логином в каждом. Учётки заводятся и отключаются вручную, при увольнении доступ гарантированно остаётся в 2–3 системах, единого аудита входов нет. Мы закрываем это одним Keycloak: вход по доменной учётке AD, 2FA и отключение сотрудника в одной точке за 30 секунд.

Почему это важно бизнесу

- Уволенный с живым доступом к CRM или финансовым сервисам — прямой риск утечки клиентской базы
- Одинаковые пароли в 12 сервисах: компрометация одного даёт злоумышленнику все остальные
- Тикеты «забыл пароль» и ручное заведение учёток съедают часы админа каждую неделю
- Без единого журнала входов нельзя ответить аудитору или разобрать инцидент «кто и когда заходил»

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

26.7

Версия Keycloak, которую ставим: Quarkus-дистрибутив, Java 21, персистентные сессии в БД

keycloak.org, ветка 26.x

300 с

Access Token Lifespan: короткий токен, чтобы отключение учётки срабатывало максимум за 5 минут

наш стандарт (Realm → Tokens)

5 мин

Changed users sync period LDAP-федерации: новые и заблокированные учётки AD подтягиваются сами

keycloak.org, Server Admin Guide (User Federation)

9000

Отдельный management-порт под /health и /metrics — мониторинг не торчит на пользовательском 80…

keycloak.org, Server Guide (management interface)

x2

Минимум узлов в кластере: Infinispan-стек jdbc-ping через общий PostgreSQL, без внешнего кэша

Keycloak HA Guide / наш стандарт

636

Только LDAPS к контроллерам домена: bind-пароль сервисной учётки не ходит открытым текстом

наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Развёртывание ядра: Keycloak 26 + PostgreSQL за reverse-proxy

Что настраиваем

Одна VM 4 vCPU / 8 GB: контейнеры quay.io/keycloak/keycloak:26.7 и postgres:16, TLS-терминация на nginx фронта

Как мы это делаем

1. Собираем оптимизированный образ: kc.sh build --db=postgres, запускаем строго kc.sh start --optimized, а не start-dev
2. Фиксируем KC_HOSTNAME=auth.<домен>, KC_HTTP_ENABLED=true и KC_PROXY_HEADERS=xforwarded — HTTP слушает только localhost за nginx
3. Выносим health/metrics: KC_HEALTH_ENABLED/KC_METRICS_ENABLED=true, management-порт 9000 закрыт снаружи, читает только Zabbix
4. Задаём контейнеру memory limit: heap = -XX:MaxRAMPercentage=70 от него, при необходимости переопределяем через JAVA_OPTS_KC_HEAP
5. Настраиваем ежедневный kc.sh export --dir + pg_dump: realm с клиентами и flow восстанавливается за минуты

РЕЗУЛЬТАТ

Стенд поднимается воспроизводимо из compose-шаблона за час; проверка живости — GET /health/ready на 9000, метрики уходят в Zabbix. Падение узла не теряет сессии: с 26.x они лежат в PostgreSQL, после рестарта пользователи не перелогиниваются.

КЛЮЧЕВОЙ НЮАНС

Hostname v2 (Keycloak 24+): без KC_HOSTNAME и KC_PROXY_HEADERS редиректы OIDC уходят на внутренний адрес. Сверяем issuer в /realms/<realm>/.well-known/openid-configuration до подключения первого клиента.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Федерация с Active Directory и перенос групп в роли

Что настраиваем

Домен AD клиента: read-only LDAP-федерация с двух контроллеров, сервисная учётка с правами только на чтение OU

Как мы это делаем

1. User Federation → ldap: vendor Active Directory, Connection URL ldaps://dc01:636 ldaps://dc02:636, Edit Mode READ_ONLY
2. Bind DN — выделенная учётка (CN=svc-keycloak,OU=Service) с правом чтения нужных OU и запретом интерактивного входа; Domain Admin не используем никогда
3. Username LDAP Attribute = sAMAccountName, маппер group-ldap-mapper тянет группы из OU=Groups и отображает их в роли клиентов
4. Sync Settings: Changed users sync period = 300 с, Full sync period = 86400 с — блокировка в AD гасит вход в SSO за 5 минут
5. Включаем Brute Force Detection на realm: лимит неудачных входов и нарастающая задержка вместо вечного lockout AD-учётки

РЕЗУЛЬТАТ

Учётки не дублируются: источник истины остаётся AD, пароли Keycloak не хранит. Отключение сотрудника — одно действие в AD, через ≤5 минут все OIDC-сервисы недоступны; активные сессии добиваем через Sign out all active sessions в консоли.

КЛЮЧЕВОЙ НЮАНС

Edit Mode WRITABLE опасен на старте: Keycloak начнёт писать атрибуты обратно в AD. Начинаем с READ_ONLY и включаем запись только там, где нужен self-service смены пароля через LDAPS.

РЕАЛИЗАЦИЯ 3 ИЗ 3

OIDC-клиенты, legacy-приложения и обязательная 2FA

Что настраиваем

8–12 сервисов: Grafana, GitLab, Wiki.js — нативный OIDC; старые интранет-приложения — через oauth2-proxy перед nginx

Как мы это делаем

1. На каждый сервис — свой confidential-клиент: точный Valid Redirect URI без «*», Client authentication = On, secret в Vault, не в compose
2. Grafana: auth.generic_oauth со scopes openid profile email и role_attribute_path из групп; GitLab — omniauth openid_connect
3. Legacy без OIDC прячем за oauth2-proxy: приложению отдаём заголовки X-Forwarded-User/Email, снаружи только SSO-логин
4. 2FA: Required Action «Configure OTP» по умолчанию, TOTP 6 цифр / 30 с; админам — WebAuthn-ключи отдельным browser-flow
5. Включаем Login Events и Admin Events с ретенцией: журнал входов доступен из консоли и по Admin API (kcadm.sh)

РЕЗУЛЬТАТ

Один логин на все сервисы, 2FA покрывает 100% пользователей без установки агентов. Разбор инцидента — фильтр по событиям LOGIN/LOGIN_ERROR за минуты; выдача доступа новому сотруднику — членство в группе AD, а не 12 заявок админу.

КЛЮЧЕВОЙ НЮАНС

Wildcard в Redirect URI — типовая дыра: authorization code можно увести на подставной callback. Прописываем точные URI и явные Web Origins, «*» не используем даже на тесте.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **start-dev в проде**
  Dev-режим включает встроенную H2 и открытый HTTP. Собираем kc.sh build + start с PostgreSQL — иначе база теряется при пересоздании контейнера

- ✕ **Не задан hostname**
  Без KC_HOSTNAME/KC_PROXY_HEADERS issuer в токенах указывает на внутренний адрес — часть клиентов за прокси молча перестаёт логиниться

- ✕ **Domain Admin в Bind DN**
  Компрометация Keycloak отдаёт весь домен. Заводим svc-учётку с правом чтения конкретных OU и запретом интерактивного входа

- ✕ **Wildcard Redirect URI**
  «*» в Valid Redirect URIs позволяет увести authorization code на чужой callback. Только точные URI и явные Web Origins

- ✕ **Длинные access-токены**
  Токен на часы = уволенный работает до истечения. Держим Access Token Lifespan 5 мин, сессии гасим через Sign out all active sessions

- ✕ **Один узел без экспорта**
  Падение единственной ноды = стоп всех новых логинов. Минимум — ежедневный kc.sh export + pg_dump; целевое — 2 ноды jdbc-ping

- ✕ **Секреты клиентов в compose**
  client_secret в git/compose расходится по бэкапам. Держим в .env вне репозитория или Vault, ротируем при смене админа

- ✕ **Апгрейд через мажор**
  Прыжок 22→26 ломает темы и SPI-плагины. Обновляем последовательно по минорным веткам, прогоняя миграции БД на стейдже

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Keycloak 26.x + PostgreSQL 16, kc.sh start за nginx с TLS, никакого start-dev
- LDAP-федерация READ_ONLY: sAMAccountName, синк изменений раз в 5 минут
- Confidential-клиенты с точными Redirect URI на каждый сервис
- Ежедневный kc.sh export + pg_dump с хранением вне хоста

НОРМАЛЬНО

- Обязательный TOTP для всех через Required Action, WebAuthn для админов
- Brute Force Detection на realm + Login/Admin Events с ретенцией 90 дней
- health/metrics на порту 9000 в Zabbix, алерт на /health/ready
- LDAPS 636 к двум контроллерам, секреты в Vault/.env вне git

ХОРОШО

- 2 ноды Keycloak: Infinispan jdbc-ping, сессии в БД, балансировщик со sticky
- oauth2-proxy для legacy-приложений — 100% сервисов за SSO
- Realm-конфигурация как код: kcadm.sh/Terraform, изменения через git
- Стейджинг-realm для обкатки апгрейдов по минорным версиям

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Все внутренние сервисы входят через одну точку SSO, локальных логинов не осталось?
- [ ] Блокировка сотрудника в AD закрывает вход во все сервисы не дольше чем за 5 минут?
- [ ] Активные сессии уволенного завершаются принудительно (Sign out all active sessions), а не ждут истечения токена?
- [ ] 2FA обязательна для всех пользователей, а не «по желанию»?
- [ ] Сервисная учётка LDAP ограничена чтением нужных OU и не является админом домена?
- [ ] Redirect URI всех OIDC-клиентов прописаны точно, без wildcard?
- [ ] Есть проверенный восстановлением бэкап: экспорт realm + дамп PostgreSQL?
- [ ] Мониторинг видит /health/ready и метрики Keycloak и срабатывает раньше жалоб пользователей?
- [ ] Журнал входов (Login Events) включён и хранится достаточно для разбора инцидентов?
- [ ] Обновления идут по минорным версиям через стейдж, а не скачком в проде?

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 26 под ключ: PostgreSQL, reverse-proxy, hostname, бэкапы, мониторинг в наш Zabbix
- Интегрируем с вашим Active Directory: федерация, маппинг групп в роли, отключение сотрудника за 30 секунд
- Подключаем к SSO весь парк сервисов — от Grafana и GitLab до legacy-приложений через oauth2-proxy
- Включаем 2FA (TOTP/WebAuthn) и журналирование входов, обучаем сотрудников за один день
- Берём на поддержку: обновления по минорным веткам, контроль сертификатов и ротация client secret

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 Server Guide — Configuring the database, Running in a container (keycloak.org — 26.7)

02 Server Guide — Configuring the hostname (v2), Using a reverse proxy (keycloak.org — 26.7)

03 Server Guide — Configuring the Management Interface (health, metrics) (keycloak.org — 26.7)

04 Server Administration Guide — User Federation (LDAP/AD) (keycloak.org — 26.7)

05 Server Administration Guide — Authentication flows, OTP и WebAuthn (keycloak.org — 26.7)

06 High Availability Guide — Infinispan, jdbc-ping, persistent sessions (keycloak.org — 26.7)

07 Securing Applications Guide — OIDC clients, kcadm CLI (keycloak.org — 26.7)

08 Наш шаблон docker-compose + nginx для IdP-стенда (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.