Центр
Управления
Гетерогенными
Инфраструктурами

Ручное обновление сервисов Clearway CA 26.2.0

ООО «Клируэй Текнолоджис»

2026

Оглавление

1 Введение

Настоящий документ — инструкция по ручному обновлению компонентов Clearway CA до новой версии, без использования графического инсталлятора. Она применяется, когда система развёрнута и обслуживается вручную (см. «Ручное развёртывание сервисов»), а также как справочник по составу и порядку операций обновления.

Предполагается, что Clearway CA уже установлен (инсталлятором или вручную) со значениями по умолчанию: каталог /opt/itc, технологическая учётная запись-владелец itc-svc, службы — пользовательские юниты systemctl --user, база данных — PostgreSQL. При иных значениях подставляйте свои. Документ охватывает обновление ядра ЦС (minica), сервиса аутентификации OpenID, Контрольной панели, Сервиса публикации, OCSP-респондера, Архиватора, адаптера SCEP и точки распространения (CDP).

Важно

Обновление затрагивает действующий удостоверяющий центр. Выполняйте его в согласованное окно обслуживания, обязательно сделайте резервные копии до начала работ и подготовьте план отката. Приватные ключи и учётные данные при обновлении не заменяются — не удаляйте и не перезаписывайте их.

Графический инсталлятор выполняет установку компонентов и ведёт историю установок, но не предназначен для пошагового обновления.

2 Принципы и подход к обновлению

Обновление сервиса — это замена артефактов приложения (исполняемых файлов и пакетов) новой версией при полном сохранении конфигурации, сертификатов, ключей и данных. Такой подход возможен потому, что в Clearway CA чётко разделены три независимых слоя:

Слой	Где расположен	Что происходит при обновлении
Приложение	Каталоги сервисов (/opt/itc/minica, /opt/itc/openid, /opt/itc/itc.minica.*, /opt/itc/ocsp, /opt/itc/itc.acm.scepAdapter.clearwayCa)	Заменяется новыми артефактами
Конфигурация	/opt/itc/itc/config, /opt/itc/itc/env, /opt/itc/minica/conf, /opt/itc/ocsp/conf, блоки Nginx, systemd-юниты	Сохраняется; при необходимости дополняется новыми ключами из релиза
Данные	PostgreSQL (cwica, openid, cwica_archive), SQLite (scep.db), сертификаты и ключи (certs, ca)	Сохраняются; схема БД мигрируется только если этого требует релиз

Из этого следуют базовые правила:

- Совместимость версий: перед обновлением изучите release notes: изменения схемы БД, формата конфигураций, протоколов взаимодействия между сервисами, требования к версиям ОС и пререквизитов. Обновляйте компоненты согласованно — не оставляйте в связке несовместимые версии ядра ЦС, Контрольной панели и OpenID.
- Простой: обновление сервиса требует его остановки. На время обновления ядра ЦС недоступны выпуск и отзыв; на время обновления OCSP/CDP — проверка статуса по соответствующему узлу. Планируйте окно обслуживания.
- Обратимость: сохранённые прежние артефакты, резервные копии БД и конфигураций позволяют вернуться к предыдущей версии (см. «Откат к предыдущей версии»).

Текущие версии перед обновлением определяются так:

- ядро ЦС — командой:

```
mclient ping    # в ответе версия и SKI ЦС
```

- Контрольная панель — раздел "Настройки" (версии ЦС, панели и сервиса публикации);
- любая служба:

```
systemctl --user status <юнит>
```

- артефакт — по сопроводительному файлу версии релиза. Зафиксируйте эти значения — они понадобятся для проверки и отката.

3 Подготовка к обновлению

1. Получите дистрибутив новой версии и release notes; при наличии — сверьте контрольные суммы артефактов.
2. По release notes определите: изменяется ли схема БД, появились ли новые ключи конфигурации, изменились ли требования к версиям ОС, PostgreSQL, Nginx, OpenSSL, есть ли особенности порядка обновления.
3. Согласуйте окно обслуживания и уведомите пользователей.
4. Проверьте пререквизиты (как правило, не меняются): поддерживаемая ОС; PostgreSQL 11–18; установленные Nginx и OpenSSL для соответствующих компонентов; свободное место под резервные копии.
5. Зафиксируйте текущие версии всех компонентов (для последующей проверки и отката).
6. Разместите артефакты новой версии на целевых серверах (например, в */tmp*).

4 Резервное копирование

> **Важно**
>
> Без актуальной резервной копии откат невозможен. Выполняйте резервное копирование ДО любых изменений и убедитесь, что копии созданы и читаемы.

Базы данных PostgreSQL — рабочая БД ЦС cwica, БД OpenID openid и архивная БД cwica_archive (если развёрнут Архиватор). Формат custom (-Fc) пригоден для pg_restore

```
# на сервере СУБД:
pg_dump -Fc -U cwica -h <db-host> cwica         > /backup/cwica_<версия>.dump
pg_dump -Fc -U cwi   -h <db-host> openid        > /backup/openid_<версия>.dump
pg_dump -Fc -U cwica -h <db-host> cwica_archive > /backup/
cwica_archive_<версия>.dump
```

Конфигурации, сертификаты, ключи и юниты — на каждом сервере с компонентами:

```
tar -czf /backup/itc-config_<версия>.tgz \
  /opt/itc/itc/config /opt/itc/itc/env /opt/itc/itc/nginx \
  /opt/itc/minica/conf /opt/itc/ocsp/conf ~/.config/systemd/user
tar -czf /backup/itc-certs_<версия>.tgz /opt/itc/itc/certs /opt/itc/minica/ca
cp /opt/itc/itc/sqlite/scep.db /backup/scep_<версия>.db        # состояние адаптера
 SCEP
```

Текущие каталоги приложений — для быстрого отката (либо архивом, либо переименованием при обновлении каждого сервиса):

```
tar -czf /backup/apps_<версия>.tgz \
  /opt/itc/minica/minica /opt/itc/minica/mclient /opt/itc/minica/bin/mjwt \
  /opt/itc/openid/openid /opt/itc/ocsp/mocsp \
  /opt/itc/itc.minica.controlPanel /opt/itc/itc.minica.spa \
  /opt/itc/itc.minica.publication /opt/itc/itc.minica.archiver \
  /opt/itc/itc.acm.scepAdapter.clearwayCa
```

5 Общий алгоритм обновления сервиса

Каждый сервис обновляется по единой последовательности; специфика отдельных сервисов приведена в разделе «Обновление отдельных сервисов».

1. Зафиксируйте текущую версию сервиса.
2. Остановите службу:

```
systemctl --user stop <юнит>.
```

3. Сохраните текущие артефакты приложения для отката (переименуйте каталог в *.bak-<версия> или заранее заархивируйте — см. «Резервное копирование»).
4. Разверните новые артефакты в каталог приложения. Конфигурацию, сертификаты, ключи и данные не трогайте.
5. При необходимости (по release notes) обновите схему БД — см. «Обновление схемы базы данных».
6. Сверьте конфигурацию: сравните поставляемый с релизом пример с текущим файлом и добавьте только новые ключи, не перезаписывая рабочие значения.
7. Если изменился файл systemd-юнита — замените его и выполните:

```
 systemctl --user daemon-reload.
```

8. Запустите службу:

```
systemctl --user start <юнит>
```

9. Проверьте: служба активна, порт слушается, версия соответствует новой, функциональный тест проходит.

Типовой пример для сервиса, поставляемого архивом каталога (например, Сервис публикации):

```
UNIT=publication.service                          # systemd-юнит сервиса
APPDIR=/opt/itc/itc.minica.publication             # каталог приложения
systemctl --user stop "$UNIT"
mv "$APPDIR" "${APPDIR}.bak-<версия>"              # сохранить прежнюю версию для
отката
mkdir -p "$APPDIR"
tar -xzf /tmp/<новый-артефакт>.tar.gz -C "$APPDIR"/
systemctl --user daemon-reload                     # если менялся файл юнита
systemctl --user start "$UNIT"
systemctl --user status "$UNIT"
```

Для сервисов, поставляемых отдельными исполняемыми файлами (ядро ЦС minica/mclient/mjwt, openid, mocsp), вместо распаковки каталога заменяйте файлы с сохранением прав на исполнение: install -m 0755 /tmp/<файл> <путь-назначения>.

6 Порядок обновления компонентов

Компоненты обновляют по зависимостям — сначала то, на что опираются остальные (та же логика, что в группах установки). Рекомендуемая последовательность:

1. Резервное копирование всех БД и файлов.
2. Схема БД ЦС cwica — миграция, если её требует релиз.
3. Ядро ЦС (minica) — сначала Корневой, затем Выдающий ЦС (каждый экземпляр отдельно).
4. Сервис аутентификации OpenID.
5. Контрольная панель.
6. Сервис публикации.
7. OCSP-респондер (mOCSP).
8. Архиватор (+ схема архивной БД cwica_archive, если требуется).
9. Адаптер SCEP.
10. Точка распространения (CDP).

При обновлении в пределах минорной версии допускается точечное обновление отдельных сервисов. При значительных изменениях обновляйте все компоненты в одном окне обслуживания, чтобы не оставлять в связке несовместимые версии.

7 Обновление отдельных сервисов

Ниже — специфика обновления каждого сервиса: обновляемые артефакты, особые шаги и проверка. Общие действия (остановка службы, сохранение прежних артефактов для отката, daemon-reload, запуск) выполняются по разделу «Общий алгоритм обновления сервиса».

7.1 Ядро ЦС (minica)

Обновляются только исполняемые файлы **ядра ЦС** — */opt/itc/minica/minica* (служба), */opt/itc/minica/mclient* (клиент) и */opt/itc/minica/bin/mjwt* (утилита JWT); служба развёрнута как пользовательский systemd-юнит minica.service (учётная запись itc-svc, каталог установки */opt/itc/minica*, порты 9080 (HTTP) и 9443 (HTTPS при включённом TLS)).

Если развёрнуты **Корневой** и **Выдающий ЦС**, обновляйте каждый экземпляр отдельно, начиная с **Корневого**. Все команды выполняются от имени itc-svc.

Обновление выполняется по разделу **«Общий алгоритм обновления сервиса»**; ниже приведена специфика ядра ЦС.

Остановите службу:

```
systemctl --user stop minica.service
```

Сохраните текущие исполняемые файлы для отката (см. раздел «Резервное копирование»):

```
cp -a /opt/itc/minica/minica     /opt/itc/minica/minica.<версия>.bak
cp -a /opt/itc/minica/mclient    /opt/itc/minica/mclient.<версия>.bak
cp -a /opt/itc/minica/bin/mjwt   /opt/itc/minica/bin/mjwt.<версия>.bak
```

Разверните новые артефакты поверх старых и выставьте право на исполнение (0755):

```
cp <путь_к_новым_артефактам>/minica  /opt/itc/minica/minica
cp <путь_к_новым_артефактам>/mclient /opt/itc/minica/mclient
cp <путь_к_новым_артефактам>/mjwt    /opt/itc/minica/bin/mjwt
chmod 0755 /opt/itc/minica/minica /opt/itc/minica/mclient /opt/itc/minica/bin/mjwt
```

Если в release notes указано изменение схемы БД — после резервного копирования базы примените поставляемый с релизом minica.up.sql к базе cwica (скрипт идемпотентный, CREATE … IF NOT EXISTS, данные не затрагивает):

```
psql -h <хост_БД> -U <пользователь_БД> -d cwica -f <путь_к_релизу>/minica.up.sql
```

Сверьте конфигурацию: при появлении в release notes новых ключей добавьте их в файлы */opt/itc/minica/conf/* (minica.yml, mclient.yml, openssl.cnf), не переписывая рабочие значения. Если изменился файл юнита *~/.config/systemd/user/minica.service*, перечитайте определения юнитов и запустите службу:

```
systemctl --user daemon-reload   # только при изменении юнита
systemctl --user start minica.service
```

При обновлении заменяются только исполняемые файлы приложения; конфигурации */opt/itc/minica/conf/*, ключи и сертификат ЦС */opt/itc/minica/ca/* (ca.key, ca.crt, chain.pem), ключи JWT/mclient, журнал logs/ и данные в БД cwica не затрагиваются.

Проверка: убедитесь, что служба активна и отвечает, а версия ЦС соответствует ожидаемой:

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
systemctl --user status minica.service
/opt/itc/minica/mclient ping
/opt/itc/minica/mclient stat
```

В ответе mclient ping сверьте версию ЦС и SKI с ожидаемой новой версией.

7.2 Сервис аутентификации OpenID

Обновляется единственный артефакт — исполняемый файл */opt/itc/openid/openid*; сервис аутентификации OpenID работает как пользовательский systemd-юнит openid.service из каталога */opt/itc/openid* и слушает порты 443 (HTTPS, вход клиентов) и 5559 (HTTP, администрирование).

Выполняйте обновление по разделу «Общий алгоритм обновления сервиса», учитывая специфику данного сервиса. Обновляется только исполняемый файл приложения — конфигурация */opt/itc/openid/config.yaml*, TLS-сертификат */opt/itc/itc/certs/openid-tls.** и данные в БД не затрагиваются.

Остановите службу:

```
systemctl --user stop openid.service
```

Сохраните текущий бинарь для отката (резервное копирование — по разделу «Резервное копирование»):

```
cp -p /opt/itc/openid/openid /opt/itc/openid/openid.<версия>.bak
```

Разверните новый бинарь на место /opt/itc/openid/openid и выставьте право на исполнение (0755):

```
install -m 0755 <путь-к-новым-артефактам>/openid /opt/itc/openid/openid
```

Повторно выдайте бинарю разрешение слушать привилегированный порт 443. Это ключевой шаг данного сервиса: право setcap привязано к файлу и теряется при его замене, без него служба не сможет открыть порт 443:

```
sudo setcap 'cap_net_bind_service=+ep' /opt/itc/openid/openid
```

Миграция схемы БД вручную не требуется: сервис сам создаёт и обновляет схему в БД PostgreSQL openid (сессии, refresh-токены, ключи) при старте.

Сверьте config.yaml на появление новых ключей согласно примечаниям к версии, сохранив рабочие значения (issuer, storage, connectors LDAP, ldapGroupRolesMap, staticClients); рабочую конфигурацию не перезаписывайте. Юнит не изменяется, поэтому systemctl --user daemon-reload не требуется.

Запустите службу:

```
systemctl --user start openid.service
```

Проверка: убедитесь, что служба активна и порт 443 слушается, затем проверьте вход пользователя в Контрольную панель (аутентификация проходит через OpenID):

```
1  systemctl --user status openid.service
2  ss -tlnp | grep :443
```

7.3 Контрольная панель

Контрольная панель состоит из двух артефактов приложения — backend ITC.Api.MiniCA.ControlPanel (пользовательский юнит controlPanel.service, порт 8407 локально) и статический SPA ITC.MiniCA.Spa; наружу панель публикуется через Nginx по HTTPS на порту 443.

Последовательность соответствует разделу «Общий алгоритм обновления сервиса»; ниже — специфика Контрольной панели (обновляются только два указанных артефакта).

Остановите backend (SPA — статические файлы, обслуживаемые Nginx, отдельного юнита не имеет и не останавливается):

```
systemctl --user stop controlPanel.service
```

Сохраните текущие каталоги артефактов для отката:

```
1  mv /opt/itc/itc.minica.controlPanel /opt/itc/itc.minica.controlPanel.bak-<версия>
2  mv /opt/itc/itc.minica.spa          /opt/itc/itc.minica.spa.bak-<версия>
```

Разверните новые артефакты в штатные каталоги:

```
1  mkdir -p /opt/itc/itc.minica.controlPanel /opt/itc/itc.minica.spa
2  tar -xzf ITC.Api.MiniCA.ControlPanel.tar.gz -C /opt/itc/itc.minica.controlPanel
3  tar -xzf ITC.MiniCA.Spa.tar.gz              -C /opt/itc/itc.minica.spa
```

Обновление схемы БД не выполняется: у Контрольной панели собственной базы данных нет.

Сверьте конфигурацию, не переписывая рабочие значения, — при необходимости только добавьте новые ключи, появившиеся в релизе:

- backend */opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json* (OpenId.Authority, Services);
- клиент */opt/itc/itc/env/client_env.js* (issuer, clientId).

Эти файлы, TLS-сертификат Nginx */opt/itc/itc/certs/nginx-tls.**, учётные данные подключённых ЦС в */opt/itc/itc/certs/* и блоки Nginx */opt/itc/itc/nginx/* лежат вне каталогов артефактов и при распаковке не затрагиваются.

Блоки Nginx обычно не меняются. Если релиз их изменил — замените файлы в */opt/itc/itc/nginx/* и примените изменения:

```
sudo nginx -t && sudo systemctl reload nginx
```

Запустите backend (daemon-reload — только если изменился сам юнит controlPanel.service; см. «Общий алгоритм обновления сервиса»):

```
systemctl --user start controlPanel.service
```

Замена артефактов затрагивает только код приложения: конфигурации, сертификаты/ключи и учётные данные ЦС остаются на прежних местах.

Проверка: убедитесь, что backend активен:

```
systemctl --user status controlPanel.service
```

Откройте Контрольную панель по HTTPS (https://<fqdn>/) и в разделе Настройки проверьте, что показана новая версия контрольной панели, а статусы компонентов — «работает». Поскольку SPA статичен, при кэшировании обновите страницу без кэша (жёсткая перезагрузка, Ctrl+F5).

7.4 Сервис публикации

Обновляется приложение фонового .NET-сервиса ITC.Api.MiniCA.Publication — публикация сертификатов и CRL на внешние точки CDP/AIA и в каталоги LDAP по расписанию; юнит publication.service, порт 9407, наружу публикуется через Nginx (TLS 443, путь */api/publication*).

Обновление выполняется по разделу «Общий алгоритм обновления сервиса» (при необходимости — с предварительным резервным копированием по разделу «Резервное копирование»); ниже приведена только специфика данного сервиса.

Остановите службу:

```
systemctl --user stop publication.service
```

Сохраните текущий каталог приложения для отката (перед распаковкой), затем развернуть новые артефакты из ITC.Api.MiniCA.Publication.tar.gz в */opt/itc/itc.minica.publication/*:

```
# сохранить текущую версию для отката
mv /opt/itc/itc.minica.publication /opt/itc/itc.minica.publication.bak-<версия>

# развернуть новую версию
mkdir -p /opt/itc/itc.minica.publication
tar -xzf ITC.Api.MiniCA.Publication.tar.gz -C /opt/itc/itc.minica.publication
```

Схема БД не обновляется: собственной базы данных у сервиса нет, файла SQLite и миграций нет — задания публикации хранятся в конфигурации, поэтому шаг «обновить схему БД» из общего алгоритма пропускается.

Сверить конфигурацию, не переписывая рабочие значения: в существующий файл */opt/itc/itc/config/itc_api_miniCA_publication_config.json* при необходимости добавить только новые ключи, появившиеся в версии <версия> (секции OpenId, PublishingCrlTasks, PublishingCertificationTasks). Рабочие значения заданий публикации и параметры OpenId, а также TLS-сертификат и JWT доступа к ЦС в */opt/itc/itc/certs/* оставить без изменений.

Реконфигурация reverse-proxy не требуется: порт 9407 и путь публикации */api/publication* (Nginx, TLS 443) не меняются, поэтому nginx -t и перезагрузка не нужны; setcap для привязки к порту также не выполняется — 9407 непривилегированный.

При изменении файла юнита выполните:

```
systemctl --user daemon-reload
```

Далее запустите службу (см. общий алгоритм):

```
systemctl --user start publication.service
```

Конфигурация (*itc_api_miniCA_publication_config.json*), сертификаты и ключи (*/opt/itc/itc/certs/*) и задания публикации при обновлении не затрагиваются — заменяются только артефакты приложения в */opt/itc/itc.minica.publication/*.

Проверка: убедиться, что служба активна и порт 9407 слушается, затем по журналу сервиса проверить отработку заданий публикации CRL/сертификатов и время следующей публикации:

```
systemctl --user status publication.service
ss -ltnp | grep ':9407'
journalctl --user -u publication.service -e
```

7.5 OCSP-респондер (mOCSP)

Обновляется единственный артефакт — исполняемый файл /opt/itc/ocsp/mocsp; сервис работает под пользовательским юнитом mocsp.service и отвечает на OCSP-запросы напрямую на порту 8888 по пути /ocsp, без Nginx. Общий порядок — по разделу «Общий алгоритм обновления сервиса», ниже только специфика mOCSP.

Остановите службу:

```
systemctl --user stop mocsp.service
```

Сохраните текущий бинарь для отката (полное резервное копирование каталога — по разделу «Резервное копирование»):

```
cp -a /opt/itc/ocsp/mocsp /opt/itc/ocsp/mocsp.<версия>.bak
```

Разверните новый исполняемый файл в */opt/itc/ocsp/mocsp* и обязательно выставьте право на исполнение (setcap не требуется — порт 8888 непривилегированный):

```
cp <путь-к-новому-mocsp> /opt/itc/ocsp/mocsp
chmod 0755 /opt/itc/ocsp/mocsp
```

Миграция БД не требуется: собственной базы у респондера нет, статусы читаются напрямую из БД ЦС. Сверьте */opt/itc/ocsp/conf/mocsp.yml* (подключение к БД ЦС, listen-port, url-path) с примером новой версии на предмет новых ключей, не переписывая рабочие значения.

При изменении файла юнита выполните:

```
systemctl --user daemon-reload
```

Далее запустите службу:

```
systemctl --user start mocsp.service
```

При обновлении заменяется только приложение (бинарь mocsp): конфигурация mocsp.yml, ключ и сертификат подписи OCSP */opt/itc/ocsp/conf/mocsp.key*, mocsp.crt, а также сертификат ca.crt не затрагиваются.

Проверка: убедитесь, что служба активна и порт 8888 слушается, затем выполните функциональный OCSP-запрос:

```
systemctl --user status mocsp.service
ss -lntp | grep 8888
openssl ocsp -issuer /opt/itc/ocsp/conf/ca.crt -cert <проверяемый>.crt -url http://
<fqdn-ocsp>:8888/ocsp -resp_text -no_nonce
```

7.6 Архиватор

Обновляется приложение Архиватора (ITC.Api.MiniCA.Archiver) в */opt/itc/itc.minica.archiver/* — пользовательский юнит archiver.service, порт 9507, — которое по расписанию @daily переносит истёкшие сертификаты из рабочей БД ЦС в отдельную архивную БД.

Обновление выполняется по разделу Общий алгоритм обновления сервиса (резервное копирование — по разделу Резервное копирование) со следующей спецификой сервиса.

Остановите службу (перенос запускается по расписанию @daily, выбирайте для обновления окно вне очередного прогона):

```
systemctl --user stop archiver.service
```

Сохраните текущий каталог приложения для отката и разверните новые артефакты из ITC.Api.MiniCA.Archiver.tar.gz:

```
mv /opt/itc/itc.minica.archiver /opt/itc/itc.minica.archiver.<версия>.bak
mkdir -p /opt/itc/itc.minica.archiver
tar -xzf ITC.Api.MiniCA.Archiver.tar.gz -C /opt/itc/itc.minica.archiver
```

Конфигурация */opt/itc/itc/config/itc_api_miniCA_archiver_config.json* при обновлении сохраняется. Сверьте в ней появившиеся ключи, не переписывая рабочие значения заданий (Source — рабочая БД, Target — архивная БД, Cron).

Схема БД (специфика сервиса). Архивная БД (по умолчанию cwica_archive) имеет ту же схему, что и рабочая. Если в release notes изменена схема БД, примените поставляемый minica.up.sql не только к рабочей, но и к архивной БД — после резервного копирования (раздел Резервное копирование):

```
psql -h <fqdn> -U <пользователь> -d cwica_archive -f minica.up.sql
```

При изменении юнита выполните:

```
systemctl --user daemon-reload
```

Далее запустите службу:

```
systemctl --user start archiver.service
```

Конфигурация, сертификаты/ключи и данные обеих баз (рабочей и архивной) при обновлении не затрагиваются — заменяются только артефакты приложения в */opt/itc/itc.minica.archiver/*.

Проверка: убедитесь, что служба активна, и по журналу проследите очередной прогон переноса — подключения к рабочей и архивной БД успешны, ошибок нет:

```
systemctl --user status archiver.service
journalctl --user -u archiver.service -f
```

7.7 Адаптер SCEP

Обновляется приложение адаптера itc.acm.scepAdapter.clearwayCa (self-contained, отдельная установка dotnet не требуется), развёрнутое в */opt/itc/itc.acm.scepAdapter.clearwayCa/* и управляемое пользовательским юнитом scepAdapter.service (порт 8440, внешняя публикация через Nginx — TLS, порт 443, путь */api/scep*).

Выполняйте по разделу «Общий алгоритм обновления сервиса», учитывая специфику адаптера.

Остановите службу:

```
systemctl --user stop scepAdapter.service
```

Сохраните текущий каталог приложения для отката (резервное копирование данных — по разделу «Резервное копирование»):

```
mv /opt/itc/itc.acm.scepAdapter.clearwayCa /opt/itc/itc.acm.scepAdapter.clearwayCa.bak-
<версия>
```

Разверните новые артефакты приложения из дистрибутива:

```
tar -xzf itc.acm.scepAdapter.clearwayCa.tar.gz -C /opt/itc/
```

Сверьте конфигурацию */opt/itc/itc/config/itc_acm_scepAdapter_clearwayCa_config.json*: при появлении в релизе новых ключей добавьте их, не переписывая рабочие значения (параметры OpenID, учётные данные ЦС, пути к сертификатам).

Файл состояния SQLite */opt/itc/itc/sqlite/scep.db* НЕ удаляйте и НЕ перезаписывайте: в нём хранится локальное состояние адаптера. Схема БД для этого сервиса не пересоздаётся.

Nginx правьте только если релиз изменил блок */api/scep*: замените его и примените изменения:

```
sudo nginx -t && sudo systemctl reload nginx
```

При изменении юнита выполните:

```
systemctl --user daemon-reload
```

Далее запустите службу:

```
systemctl --user start scepAdapter.service
```

Конфигурация, TLS/SCEP-сертификаты и цепочка (*/opt/itc/itc/certs/*), учётные данные ЦС (JWT/ключ) и файл состояния scep.db расположены вне каталога приложения (*/opt/itc/itc/…*) и при обновлении не затрагиваются — заменяются только артефакты приложения.

Проверка: убедитесь, что служба активна, порт слушается и внешний адрес отвечает:

```
systemctl --user status scepAdapter.service
ss -tlnp | grep 8440
curl -kI https://<fqdn>/api/scep
```

При наличии выполните функциональный тест SCEP-клиентом uScepClient.

7.8 Точка распространения (CDP)

Точка распространения (CDP) раздаёт статический контент — списки отзыва (CRL) и сертификаты ЦС — по протоколу HTTP (стандартный порт 80) средствами веб-сервера Nginx из каталога */opt/itc/web*; собственного исполняемого файла и отдельного systemd-юнита у сервиса нет (контент отдаёт юнит nginx), поэтому обновление, как правило, не требует замены бинарных файлов и сводится к замене статического контента и/или Nginx-блока точки распространения.

Общие шаги (сохранение артефактов для отката, сверка конфигурации, финальная проверка) выполняйте по разделу «Общий алгоритм обновления сервиса», резервное копирование — по разделу «Резервное копирование». Ниже приведена специфика данного сервиса.

Остановка отдельной службы не требуется: у точки распространения нет собственного юнита, а изменения применяются перезагрузкой конфигурации Nginx без остановки веб-сервера (systemctl reload nginx).

Сохраните для отката текущий статический контент и Nginx-блок точки распространения:

```
sudo cp -a /opt/itc/web /opt/itc/web.bak-<версия>
sudo cp -a /etc/nginx/<конфиг-CDP>.conf /etc/nginx/<конфиг-CDP>.conf.bak-<версия>
```

Если релиз поставляет обновлённый статический контент, разверните его в */opt/itc/web*, не затрагивая уже опубликованные файлы CRL/сертификатов и настроенные пути — эти файлы формирует и обновляет Сервис публикации:

```
sudo cp -a <путь-к-новому-контенту>/. /opt/itc/web/
```

Если релиз меняет конфигурацию, обновите Nginx-блок точки распространения и сверьте новые ключи, сохранив рабочие значения (пути публикации, server_name/<fqdn-cdp> и т. п.), не переписывая действующие настройки:

```
sudo cp -a <новый-конфиг-CDP>.conf /etc/nginx/<конфиг-CDP>.conf
```

Проверьте синтаксис и примените конфигурацию перезагрузкой Nginx:

```
sudo nginx -t && sudo systemctl reload nginx
```

При обновлении затрагиваются только артефакты приложения (статический контент и/или конфигурация Nginx): конфигурации с рабочими значениями, сертификаты/ключи и опубликованные данные CRL/сертификатов не изменяются.

Проверка: убедитесь, что настроенный URL точки распространения (CDP/AIA) отвечает по HTTP и отдаёт актуальные списки CRL и сертификаты ЦС:

```
curl -I http://<fqdn-cdp>/<путь-к-crl>
```

8 Обновление схемы базы данных

Схема БД ЦС задаётся файлом minica.up.sql из состава релиза. Скрипт идемпотентный (CREATE TABLE ... IF NOT EXISTS, CREATE INDEX ... IF NOT EXISTS): повторное применение создаёт недостающие объекты и не затрагивает существующие данные.

Применяйте миграцию только если release notes указывают на изменение схемы, и только после резервного копирования БД. При остановленной службе ЦС выполните на сервере СУБД:

```
PGPASSWORD='<ПАРОЛЬ>' psql -U cwica -h <db-host> -d cwica         -f /tmp/
minica.up.sql

# та же схема — для архивной БД, если развёрнут Архиватор:
PGPASSWORD='<ПАРОЛЬ>' psql -U cwica -h <db-host> -d cwica_archive -f /tmp/
minica.up.sql
```

- если релиз требует структурных изменений существующих таблиц (не покрываемых IF NOT EXISTS), выполняйте миграционные шаги строго по инструкции из release notes соответствующей версии;
- база данных OpenID (openid) мигрируется сервисом автоматически при старте — ручных действий не требуется;
- локальное состояние адаптера SCEP (SQLite /opt/itc/itc/sqlite/scep.db) сохраняется как есть, миграции не требует.

> **Важно**
>
> Не применяйте изменения схемы на работающем ЦС во время активного выпуска. Выполняйте миграцию в окне обслуживания при остановленной службе ядра ЦС и наличии свежего дампа БД.

9 Проверка после обновления

По каждому обновлённому сервису убедитесь, что служба активна:

```
systemctl --user status <юнит>
```

Результат: нужный порт слушается, а версия соответствует новой. Ориентир по компонентам:

Компонент	Юнит	Порт	Быстрая проверка
Ядро ЦС (minica)	minica.service	9080/9443	mclient ping (версия + SKI), mclient stat
OpenID	openid.service	443, 5559	порт 443 слушается; вход в панель проходит
Контрольная панель	controlPanel.service	8407 (за Nginx 443)	панель открывается; «Настройки» показывают новую версию
Сервис публикации	publication.service	9407	задания публикации CRL/ сертификатов отрабатывают
OCSP (mOCSP)	mocsp.service	8888	openssl ocsp … -url http:// <fqdn>:8888/ocsp
Архиватор	archiver.service	9507	прогон переноса без ошибок (журнал)
Адаптер SCEP	scepAdapter.service	8440 (за Nginx 443)	https://<fqdn>/api/scep отвечает
Точка распространения	— (Nginx)	443/HTTP	URL CDP/AIA отдаёт актуальные CRL/сертификаты

Дополнительно выполните сквозной функциональный тест: выпуск тестового сертификата по шаблону tls, его отзыв и updcrl на ЦС; проверку статуса по OCSP; вход в Контрольную панель и просмотр раздела «Настройки» (все компоненты в состоянии «работает», версии соответствуют). Просмотрите журнал событий ЦС и журналы служб на отсутствие ошибок после запуска.

10 Откат к предыдущей версии

Если проверка выявила проблему, выполните откат в порядке, обратном обновлению:

1. Остановите обновлённую службу:

```
systemctl --user stop <юнит>.
```

2. Верните прежние артефакты приложения из резервной копии или переименованного каталога *.bak-<версия>.
3. Если применялась миграция схемы БД — восстановите базу из дампа (при остановленной службе ЦС):

```
systemctl --user stop minica.service
PGPASSWORD='<ПАРОЛЬ>' pg_restore -U cwica -h <db-host> -d cwica --clean --if-exists /backup/cwica_<версия>.dump
```

Верните прежние конфигурации и юниты, если они заменялись, затем выполните:

```
systemctl --user daemon-reload
```

Запустите службу и повторите проверку.

> **Важно**
>
> Восстановление БД из дампа возвращает данные на момент копии — изменения, сделанные после неё (выпуски, отзывы), будут потеряны. По возможности выполняйте откат до возобновления выпуска сертификатов.

11 Чек-лист обновления

- изучены release notes; получены и сверены артефакты новой версии;
- зафиксированы текущие версии всех компонентов;
- согласовано окно обслуживания; пользователи уведомлены;
- выполнено резервное копирование: БД (cwica, openid, cwica_archive), конфигурации, сертификаты и ключи, SQLite SCEP, каталоги приложений;
- при необходимости применена миграция схемы БД к cwica (и cwica_archive);
- сервисы обновлены в рекомендованном порядке: ЦС → OpenID → панель → публикация → OCSP → архиватор → SCEP → CDP;
- для OpenID повторно выполнен setcap; для файловых сервисов выставлены права на исполнение;
- сверены новые ключи конфигурации без перезаписи рабочих значений;
- пройдены функциональные проверки по каждому сервису; журналы без ошибок;
- версии соответствуют ожидаемым; резервные копии и каталоги *.bak сохранены до подтверждения стабильной работы.