Руководство администратора Clearway CA 26.3.0

Документация Clearway CA

6 November, 2026

Table of Contents

1 Общие сведения о продукте ... 9
1.1 Назначение документа ... 9
1.2 Область применения ... 10
1.3 Целевая аудитория ... 10
1.4 Основные функции ... 10
1.5 Бесплатная и платная редакции продукта ... 12
2 Архитектура и состав системы ... 14
2.1 Состав компонентов ... 14
2.2 Службы и сетевые порты ... 16
2.3 Хранение данных ... 18
2.4 Взаимодействие компонентов ... 19
2.5 Схемы развертывания ... 21
2.6 Типовые архитектуры внедрения ... 22
2.6.1 1. Установка на одном сервере ... 22
2.6.2 2. Установка на двух серверах ... 23
2.6.3 3. Установка на трёх серверах ... 24
2.6.4 4. Установка на четырёх серверах ... 25
2.6.5 5. Кластерная установка ... 26
2.7 Технические требования ... 27
2.7.1 Рекомендуемые технические требования к серверному оборудованию ... 27
2.7.2 Требования к АРМ оператора ... 29
2.7.3 Требования к системному ПО ... 30
2.7.4 Требования к сети ... 33
2.8 Технологический стек ... 37
3 Требования к среде ... 38
4 Описание ролевой модели ... 47
4.1 Перечень прав ... 48
4.1.1 Права и пункты меню Контрольной панели ... 49
4.2 Составные роли-маркеры ... 51

4.3 Стандартные роли и матрица доступа ... 51
4.3.1 Добавление новых прав 26.3.0 в карту групп ... 53
4.4 Разграничение доступа к шаблонам ... 54
5 Установка ... 56
5.1 Общие сведения об установке ... 56
5.1.1 Установщик ... 56
5.1.2 Группы установки и компоненты ... 58
5.1.3 Порядок установки ... 62
5.1.4 Пререквизиты ... 65
5.1.5 После установки ... 68
5.2 Ручное развертывание сервисов (без инсталлятора) ... 69
5.2.1 Общая подготовка сервера ... 69
5.2.2 Подготовка СУБД PostgreSQL ... 71
5.2.3 Развертывание базы данных ЦС ... 75
5.2.4 Развертывание ядра ЦС (minica) ... 78
5.2.5 Общие операции для вспомогательных сервисов ... 88
5.2.6 Развертывание сервиса аутентификации OpenID ... 94
5.2.7 Развертывание Keycloak (вместо OpenID) ... 101
5.2.8 Развертывание Контрольной панели ... 108
5.2.9 Развертывание сервиса публикации ... 113
5.2.10 Развертывание OCSP-ответчика (mOCSP) ... 117
5.2.11 Развертывание Архиватора ... 120
5.2.12 Развертывание адаптера SCEP ... 122
5.3 Настройка ядра ЦС (minica.yml) ... 126
5.3.1 Порядок изменения параметров ... 126
5.3.2 Сеть и HTTP/HTTPS-сервер ... 127
5.3.3 TLS и взаимная аутентификация (mTLS) ... 128
5.3.4 Контроль рассогласования времени (max-dt-sec) ... 129
5.3.5 Ключ и сертификат ЦС ... 129
5.3.6 Шаблоны, поля CDP/AIA и сроки действия ... 132
5.3.7 Шифрование данных ЦС (ca.encryption) ... 135
5.3.8 Восстановление ключей субъектов (KRA) ... 135

5.3.9 База данных ... 135
5.3.10 Сервисные роли JWT (ролевая модель) ... 136
5.3.11 Управление службой ... 138
5.4 Настройка вспомогательных сервисов ... 139
5.4.1 Конфигурационные файлы .NET-сервисов ... 141
5.4.2 Сервис аутентификации OpenID ... 142
5.4.3 Контрольная панель ... 144
5.4.4 Сервис публикации ... 149
5.4.5 OCSP-респондер mOCSP ... 150
5.4.6 Архиватор ... 156
5.4.7 Сервис резервного копирования ... 157
5.4.8 SCEP-адаптер ... 162
5.4.9 Точка распространения CRL и сертификатов (CDP) ... 164
6 Обеспечение безопасной среды ... 167
6.1 Защита соединений центра сертификации (TLS/mTLS) ... 168
6.1.1 Включение взаимной аутентификации (mTLS) ... 169
6.2 Защита подключений к СУБД и каталогу LDAP ... 171
6.2.1 Подключение ядра ЦС и mOCSP к СУБД ... 171
6.2.2 Сервер PostgreSQL ... 173
6.2.3 Остальные подключения ... 174
6.3 Ограничение сетевого доступа ... 176
6.3.1 Веб-интерфейс установщика ... 177
6.4 Ограничение прав доступа к файлам ... 178
6.4.1 Проверка и исправление прав ... 181
6.4.2 Утилита change_right.sh ... 183
6.5 Хранение секретов во внешнем хранилище (EXC) ... 183
6.5.1 Как это работает ... 183
6.5.2 Путь к секрету ... 184
6.5.3 Подготовка ... 184
6.5.4 Какие секреты можно получать из EXC ... 184
6.5.5 Секция ess ядра ЦС и mOCSP ... 185
6.5.6 Контрольная панель, сервис публикации и архиватор ... 186

6.5.7 Пароли в формах Контрольной панели ... 188
6.5.8 Проверка после настройки EXC ... 189
6.6 Хранение ключа ЦС на HSM/Rutoken (PKCS#11) ... 189
6.7 Шифрование данных сертификатов в БД ... 190
6.8 Механизм KRA (резервное копирование и восстановление ключей) ... 190
6.9 Синхронизация времени и защита от повторов ... 191
6.10 Защита учётных записей и входа ... 192
6.10.1 Встроенная учётная запись OpenID ... 192
6.10.2 Параметры клиента OpenID Контрольной панели ... 194
6.10.3 Предупреждение при входе ... 194
6.11 Разграничение доступа ... 195
6.11.1 Минимальные права сервисов ... 195
7 Типичные операции ... 196
7.1 Управление доступом ... 196
7.1.1 Назначение ролей через группы AD/LDAP ... 196
7.1.2 Выпуск сертификатов с одобрением ... 200
7.2 Добавление сертификата ЦС в доверенные ... 200
7.2.1 Какой сертификат добавлять ... 201
7.2.2 Проверка подлинности файла ... 202
7.2.3 Linux ... 202
7.2.4 Windows ... 203
7.2.5 Типовые проблемы ... 204
7.3 Ротация ключа ЦС ... 205
7.3.1 Что меняется при ротации ... 205
7.3.2 Способы ротации ... 206
7.3.3 Ротация из Контрольной панели ... 207
7.3.4 Ротация с новой ключевой парой ... 212
7.3.5 Как ядро ЦС работает с прежними ключами ... 221
7.3.6 Как mOCSP обслуживает прежний и новый ЦС ... 224
7.3.7 Типовые проблемы ... 226
7.3.8 Файлы, задействованные при ротации ... 227
7.4 Управление службами ... 228

7.4.1 Условия работы пользовательских юнитов ... 229
7.4.2 Команды управления ... 229
7.5 Обновление продукта ... 230
7.5.1 Подготовка ... 231
7.5.2 Установка обновления ... 231
7.5.3 Проверка результата ... 232
7.6 Резервное копирование ... 232
7.6.1 Что копировать ... 233
7.6.2 Задания резервного копирования в Контрольной панели ... 235
7.6.3 Ручное резервное копирование ... 245
7.7 Архивирование истекших сертификатов ... 248
7.7.1 Изменение расписания и подключения к БД ... 249
7.7.2 Проверка работы ... 249
7.8 Периодический контроль целостности исполняемых файлов ... 250
7.9 Типовые задачи замены сертификатов ... 252
7.9.1 Включение функции ... 252
7.9.2 Требования ... 253
7.9.3 Порядок работы ... 253
7.9.4 Проверка результата ... 254
7.9.5 Типовые проблемы ... 254
8 Журналирование и мониторинг ... 256
8.1 Параметры журналирования (секции log и rotate) ... 258
8.1.1 Параметры журнала в командной строке службы ... 260
8.1.2 Просмотр журналов ... 261
8.2 Контроль целостности записей журнала ... 262
8.3 Отказоустойчивый журнал событий безопасности (secLog) ... 264
8.3.1 Общие параметры ... 264
8.3.2 Контроль занятости диска (watchdog) ... 264
8.3.3 Уведомления ... 265
8.3.4 Буфер событий в оперативной памяти ... 266
8.3.5 Фильтры событий ... 267
8.4 Мониторинг и самодиагностика ... 267

8.4.1 Проверка доступности ЦС ... 267
8.4.2 Показатели работы ЦС ... 268
8.4.3 Состояние служб ... 270
8.4.4 Проверка сервиса публикации и архиватора по HTTP ... 270
8.4.5 Оповещения по электронной почте ... 271
9 Диагностика и устранение неисправностей ... 272
9.1 Средства диагностики ... 272
9.1.1 Проверка доступности ЦС (mclient) ... 272
9.1.2 Проверка из Контрольной панели ... 273
9.1.3 Статус служб и журналы ... 273
9.2 Коды ответов API ... 274
9.3 Типовые проблемы и их устранение ... 278
9.3.1 Служба ЦС не запускается ... 279
9.3.2 Ошибка проверки времени (код 100, bad request time) ... 281
9.3.3 Нет доступа к базе данных (код 500, can't access to database) ... 282
9.3.4 Ошибки TLS, JWT и подписи (коды 5 и 10) ... 283
9.3.5 OCSP-ответчик mOCSP ... 284
9.3.6 Адаптер SCEP ... 286
9.3.7 MS-WSTEP ... 288
9.3.8 Контрольная панель ... 288
9.3.9 Резервное копирование и восстановление ... 290
9.3.10 Переполнение диска под журналы (watchdog журнала безопасности) ... 294
9.4 Быстрый чек-лист диагностики ... 295
10 Приложения ... 296
10.1 Приложение А. Ключевые параметры minica.yml ... 296
10.1.1 Сеть, TLS и контроль времени ... 296
10.1.2 Ключ ЦС, сроки действия и расширения ... 297
10.1.3 JWT и подпись запросов ... 298
10.1.4 База данных ... 299
10.1.5 Журналирование ... 299
10.2 Приложение Б. Сетевые порты ... 300
10.2.1 Порты для клиентов ... 300

10.2.2 Порты для взаимодействия компонентов 301
10.2.3 Служебные порты 303
10.3 Приложение В. Матрица ролей и прав 304
10.4 Приложение Г. Основные команды mclient 306
10.4.1 Подготовка к работе 306
10.4.2 Команды 307

1 Общие сведения о продукте

Настоящий документ представляет собой Руководство администратора программного обеспечения «Система централизованного анализа и управления гетерогенными инфраструктурами — Clearway CA» версии 26.3.0 (далее Clearway CA или продукт). Разработчик: ООО «Клируэй Текнолоджис».

Clearway CA является центром сертификации (Certification Authority, ЦС) на основе стандарта X.509v3. Он работает в составе инфраструктуры открытых ключей (PKI) организации, то есть системы ключей и сертификатов, по которым серверы, устройства и пользователи подтверждают свою подлинность. ЦС выпускает сертификаты, отзывает их и сообщает клиентам, какие сертификаты больше недействительны. Возможности продукта перечислены в разделе «Основные функции».

Продукт состоит из отдельных микросервисов и разворачивается на серверах под управлением ОС Linux. Запросы, сертификаты, журналы и списки отзыва хранятся в базе данных. Администраторы и пользователи работают с ЦС через веб-интерфейс (Контрольную панель) и через API.

1.1 Назначение документа

Руководство описывает, как установить, настроить, привести в безопасное состояние и сопровождать Clearway CA. Описаны два способа установки: автоматическая установка штатным установщиком и ручное развертывание и настройка отдельных сервисов. Кроме того, руководство содержит справочные сведения по сообщениям журналов, сетевым портам, ролям доступа и типовым эксплуатационным задачам.

Где искать сведения для основных задач администратора:

Задача администратора	**Где описано**
Понять, из каких компонентов состоит продукт и как они взаимодействуют	Глава «Архитектура и состав системы»
Проверить, отвечают ли серверы, СУБД, сеть и рабочие места требованиям	Глава Требования к среде (see page 38)
Спланировать права пользователей и их назначение через группы каталога	Глава Описание ролевой модели (see page 47)
Установить продукт, настроить ядро ЦС и вспомогательные сервисы	Глава «Установка», разделы Настройка ядра ЦС (minica.yml) (see page 126) и Настройка вспомогательных сервисов (see page 139)

Привести продукт и среду его работы в безопасное состояние	Глава Обеспечение безопасной среды (see page 167)
Управлять доступом и службами, выполнить ротацию ключа ЦС, обновление, резервное копирование, типовые задачи замены сертификатов	Глава «Типичные операции»
Настроить журналы и мониторинг	Глава Журналирование и мониторинг (see page 256)
Найти причину сбоя	Глава «Диагностика и устранение неисправностей»
Найти параметр `minica.yml`, сетевой порт, право доступа или команду `mclient`	Глава «Приложения»

1.2 Область применения

Документ применяется при развертывании и эксплуатации Clearway CA версии 26.3.0 в инфраструктуре открытых ключей организации:

- при первичной установке продукта;
- при настройке компонентов продукта и СУБД;
- при обеспечении безопасного режима работы;
- при обслуживании и обновлении сервисов.

1.3 Целевая аудитория

Руководство предназначено для администраторов, которые устанавливают, настраивают и сопровождают Clearway CA и отвечают за его безопасную работу. Предполагается, что читатель:

- владеет базовыми навыками администрирования ОС Linux, в том числе управления службами через `systemctl`;
- умеет работать с СУБД PostgreSQL;
- понимает принципы работы инфраструктуры открытых ключей, сертификатов X.509 и протоколов TLS и mTLS (взаимная проверка подлинности клиента и сервера по сертификатам).

1.4 Основные функции

Ниже возможности Clearway CA сгруппированы по назначению. Ссылки в скобках ведут к подробному описанию.

Жизненный цикл сертификатов

- выпуск сертификатов X.509v3 по запросам на сертификат (CSR) в формате PKCS#10: запрос содержит открытый ключ и сведения о владельце, закрытый ключ в него не входит;
- выпуск по настраиваемым шаблонам: шаблоном называется набор ограничений и расширений, по которому ЦС проверяет запрос и выпускает сертификат;
- отзыв сертификатов с указанием причины;
- формирование списков отзыва: полного CRL и разностного Delta CRL, который содержит только изменения с момента выпуска последнего полного CRL;
- ответы о статусе отдельного сертификата по протоколу OCSP (сервис mOCSP): приложение проверяет конкретный сертификат в момент использования, не загружая список отзыва;
- автоматический выпуск сертификатов по протоколу SCEP (адаптер SCEP) для сетевых устройств, серверов Linux и других клиентов этого протокола: право на выпуск клиент подтверждает одноразовым паролем;
- выпуск сертификатов компьютерам и пользователям Windows в домене Active Directory по протоколу MS-WSTEP (адаптер MS-WSTEP): право на выпуск подтверждается доменной учётной записью (Kerberos).



Протокольные адаптеры

Штатный установщик Clearway CA разворачивает адаптеры SCEP и MS-WSTEP: для них предусмотрены группы установки «SCEP» и «MS-WSTEP» (см. «Группы установки и компоненты»). Для MS-WSTEP нужен домен Active Directory. Адаптеры ACMEv2, CMP, EST и MS-RPC установщик не разворачивает, они поставляются отдельно.

Хранение и защита данных

- хранение запросов, сертификатов, шаблонов, журналов и списков отзыва в базе данных PostgreSQL;
- журнал событий с контролем целостности записей (см. «Контроль целостности записей журнала»);
- отдельный отказоустойчивый журнал событий безопасности (см. «Отказоустойчивый журнал событий безопасности (secLog)»);
- шифрование данных сертификатов в базе данных ЦС, которое включается в шаблоне сертификата (см. «Шифрование данных сертификатов в БД»);
- хранение закрытых ключей выпущенных сертификатов в зашифрованном виде и их восстановление при утрате: ключи шифруются сертификатом агента восстановления ключей (см. «Механизм KRA (резервное копирование и восстановление ключей)»);
- опциональное хранение ключа ЦС на аппаратном модуле HSM через интерфейс PKCS#11: ключ не может быть несанкционированно извлечён из модуля (см. «Хранение ключа ЦС на HSM/Rutoken (PKCS#11)»).

Управление и доступ

- управление через веб-интерфейс (Контрольную панель) и API;
- аутентификация и авторизация через сервер OpenID Connect с разграничением доступа по ролям (см. Описание ролевой модели (see page 47)).

Сопровождение из Контрольной панели

- резервное копирование и восстановление ЦС заданиями: задания выполняет отдельный сервис резервного копирования, а Контрольная панель передаёт ему команды; задание, запущенное по расписанию или вручную, копирует базу данных ЦС, а с узлов ЦС по SSH копирует файлы конфигурации и исполняемый файл сервиса; из копии восстанавливаются база данных ЦС и выбранные узлы (см. «Резервное копирование»);
- ротация ключа ЦС: Контрольная панель формирует запрос на сертификат ЦС на действующем ключе («Продление») или на новой ключевой паре («Перевыпуск»), вышестоящий ЦС выпускает по запросу сертификат, и администратор загружает его на той же странице (см. «Ротация ключа ЦС»);
- типовые задачи (пошаговые мастера замены TLS-сертификата на серверах nginx и PostgreSQL): Контрольная панель подключается к серверу по SSH, выпускает сертификат по выбранному шаблону, записывает его в файлы и при необходимости перезапускает сервис.

Типовые задачи включаются параметром `Features.TypicalTask` в конфигурации серверной части Контрольной панели. Мастера доступны пользователю с правом `rleMiniCACertReq-w`; права и требования к целевым серверам описаны в разделе «Типовые задачи замены сертификатов». Порядок работы с мастерами изложен в Руководстве пользователя, раздел «Типовые задачи».

Рабочее место пользователя

- конструктор CSR-запросов в Контрольной панели: формирует команду openssl для Linux, INF-файл и команду certreq для Windows или создаёт ключ и запрос прямо на аппаратном носителе; закрытый ключ в ЦС не передаётся;
- раздел Контрольной панели «Аппаратные токены»: проверяет, установлены ли на компьютере пользователя компоненты для работы с Рутокеном, JaCarta и КриптоПро CSP, показывает содержимое носителя и позволяет добавить сертификат к ключевой паре, переименовать ключевые пары и сертификаты, удалить ключевые пары, сертификаты и контейнеры.

Операции с носителями выполняются локально в браузере пользователя: сведения о носителях и PIN-коды в ЦС не передаются. Раздел «Аппаратные токены» доступен всем пользователям, отдельного права не требуется. Общие требования к рабочему месту приведены в главе Требования к среде (see page 38), а компоненты для работы с носителями и порядок работы с ними описаны в Руководстве пользователя (раздел «Аппаратные токены»).

1.5 Бесплатная и платная редакции продукта

Продукт выпускается в двух редакциях: бесплатной и платной. Ограничения бесплатной редакции приведены в таблице ниже.

Clearway CA распространяется в бесплатной и платной редакциях. Функциональные ограничения бесплатной редакции:

Параметр	Бесплатная редакция	Платная редакция
Скорость выпуска сертификатов	Не более 10 в минуту	Без ограничений
Общее количество сертификатов	Не более 2000 единиц*	Без ограничений
Стартовый серийный номер	Фиксирован (нумерация с 1)	Настраивается
Кластеризация Active-Active	Недоступно	Настраивается
Хранение ключа ЦС на HSM (PKCS#11)	Недоступно	Доступно

* В лимит входят все сертификаты, выпущенные этим ЦС, в том числе служебные TLS для nginx, OpenID и других компонентов установки.

2 Архитектура и состав системы

Глава описывает, из каких компонентов состоит Clearway CA, какие службы и порты они используют, где хранят данные и как взаимодействуют друг с другом. Эти сведения нужны, чтобы спланировать размещение компонентов по серверам и правила межсетевого экрана.

Clearway CA устроен как модульная система центра сертификации (ЦС). Основу составляет ядро ЦС: сервис minica (MiniCA). Вокруг него работают сервисы аутентификации, управления, проверки статуса и распространения сертификатов.

Прикладные сервисы работают как пользовательские юниты systemd технологической учётной записи (по умолчанию itc-svc) и управляются командой `systemctl --user` от её имени (см. «Управление службами»). Nginx и PostgreSQL работают как системные службы ОС. Контрольная панель, сервис публикации и адаптеры SCEP и MS-WSTEP доступны клиентам через обратный прокси Nginx, который принимает запросы клиентов по HTTPS (порт 443) и передаёт их сервисам.

2.1 Состав компонентов

В таблице перечислены компоненты Clearway CA и их назначение. Какие из них ставить и на какие серверы, зависит от выбранной схемы развертывания (см. «Схемы развертывания» ниже).

Компонент	**Назначение**
Ядро ЦС (minica)	Сетевой REST-сервис центра сертификации: выпуск и отзыв сертификатов X.509 по шаблонам, выпуск списков отзыва (полного CRL и разностного Delta CRL), разграничение доступа сервисов по ролям в JWT (токене доступа, который ЦС выпускает сервису). Одна и та же программа работает и корневым, и выдающим ЦС: роль задаётся конфигурацией. Конфигурация хранится в файле `/opt/itc/minica/conf/minica.yml`
Сервис аутентификации OpenID (openid)	Поставщик OpenID Connect на основе Dex: единый вход для компонентов, проверка учётных записей в каталоге LDAP/Active Directory, встроенная учётная запись для входа без каталога, сопоставление групп каталога с правами Clearway CA, выдача токенов доступа (access) и обновления (refresh)
Контрольная панель (ControlPanel)	Веб-интерфейс для работы с одним или несколькими ЦС: серверная часть ITC.Api.MiniCA.ControlPanel (ASP.NET), одностраничное приложение ITC.MiniCA.Spa, которое работает в браузере, и конфигурация Nginx, которая раздаёт приложение и передаёт запросы серверной части. Вход выполняется через сервис OpenID. Из Контрольной панели также управляются задачи публикации, резервное копирование и восстановление ЦС, ротация ключа ЦС и типовые задачи замены сертификатов

Сервис публикации (Publication)	Сервис ITC.Api.MiniCA.Publication: публикует по заданиям с расписанием в формате cron списки отзыва (в каталог на сервере, на удалённый сервер по SCP или в каталог LDAP) и выпущенные сертификаты (в каталог LDAP)
OCSP-ответчик (mOCSP)	Сервис mocsp: отвечает на запросы о статусе отдельного сертификата по протоколу OCSP, так что клиенту не нужно загружать весь CRL. По умолчанию определяет статус по БД ЦС в момент запроса; источником сведений может быть и CRL (см. «OCSP-респондер mOCSP»)
Архиватор (Archiver)	Сервис ITC.Api.MiniCA.Archiver: по расписанию (по умолчанию `@daily`, раз в сутки в 00:00 UTC) переносит истекшие сертификаты вместе с запросами и событиями из рабочей БД ЦС в отдельную архивную БД
Сервис резервного копирования (Backup)	Сервис ITC.Api.MiniCA.Backup: выполняет задания резервного копирования и восстановления ЦС. По расписанию cron и по командам из Контрольной панели копирует БД ЦС, а с узлов ЦС по SSH копирует файлы конфигурации и исполняемый файл ЦС, хранит копии и восстанавливает из них ЦС. Запросы раздела «Задачи → Резервное копирование» ему передаёт серверная часть Контрольной панели
Адаптер SCEP (scepAdapter)	Сервис itc.acm.scepAdapter.clearwayCa: выпуск и обновление сертификатов по протоколу автоматического выпуска SCEP, по которому устройства и серверы сами запрашивают сертификаты. С адаптером работают сетевые устройства, серверы Linux (служба certmonger) и другие клиенты SCEP. Право на выпуск клиент подтверждает одноразовым паролем, который пользователь получает у адаптера с токеном OpenID. Состояние адаптера хранится в SQLite
Адаптер MS-WSTEP	Сервисы itc.acm.wstep.cep (служба политик CEP) и itc.acm.wstep.cesAdapter.clearwayCa (служба регистрации CES): выпуск сертификатов компьютерам и пользователям Windows в домене Active Directory. CEP сообщает компьютеру доступные шаблоны, CES принимает запрос, передаёт его в ЦС и возвращает сертификат. Проверка подлинности выполняется по Kerberos: право на выпуск подтверждается доменной учётной записью
Точка распространения CRL и сертификатов (CDP)	Веб-сервер Nginx раздаёт по HTTP файлы CRL и Delta CRL из каталога `/opt/itc/web/cdp` и сертификаты ЦС из каталога `/opt/itc/web/aia`. Адреса файлов в этих каталогах ЦС записывает в выпускаемые сертификаты (параметры `ca.cdp` и `ca.aia`, см. «Шаблоны, поля CDP/AIA и сроки действия»): CDP указывает, где взять CRL, а AIA указывает, где взять сертификат издателя

Консольный клиент (mclient)	Утилита командной строки для обращения к ядру ЦС: выпуск сертификатов, управление шаблонами, работа с JWT. Устанавливается на узел ЦС вместе с ядром. Права клиента ограничены ролью в его токене

Адаптеры SCEP и MS-WSTEP разворачивает штатный установщик; адаптеры ACMEv2, CMP, EST и MS-RPC поставляются отдельно (см. «Группы установки и компоненты»).

2.2 Службы и сетевые порты

Таблица содержит службы компонентов и порты, которые они слушают при установке по умолчанию. Столбец «Как доступен клиентам» показывает, обращаются ли клиенты к порту напрямую или через Nginx.

Компонент	Юнит systemd	Порт по умолчанию	Протокол	Как доступен клиентам
Ядро ЦС (minica)	minica.service	9443 / 9080	HTTPS / HTTP	Напрямую. Установщик включает TLS, и ЦС принимает подключения на порту 9443. Порт 9080 используется, только если TLS выключен
Сервис аутентификации OpenID	openid.service	443	HTTPS	Напрямую. Адрес и порт задаёт параметр `web.https` в `/opt/itc/openid/config.yaml`
OpenID, служебный интерфейс	openid.service	5559	HTTP	Клиентам не публикуется. Адрес задаёт параметр `adminServer.http`
Контрольная панель, серверная часть	controlPanel.service	8407	HTTP	Через Nginx (443), путь `/api/minica`. Файлы веб-интерфейса Nginx раздаёт сам из каталога `/opt/itc/itc.minica.spa`
Сервис публикации	publication.service	9407	HTTP	Через Nginx (443), путь `/api/publication`

OCSP-ответчик (mOCSP)	mocsp.service	8888	HTTP	Напрямую, адрес `http://<fqdn>:8888/ocsp`
Архиватор	archiver.service	9507	HTTP	Клиентам не публикуется
Сервис резервного копирования	Служба сервиса резервного копирования	Задаётся параметром запуска `--urls`, в примерах 8409	HTTP	Клиентам не публикуется: к нему обращается серверная часть Контрольной панели по адресу из `Services.BackupUrl`
Адаптер SCEP	scepAdapter.service	8440	HTTP	Через Nginx (443), путь `/api/v1/scep`
Адаптер MS-WSTEP, служба политик CEP	wstepCep.service	8441	HTTPS	Через Nginx (443), путь `/CEP_Kerberos/`
Адаптер MS-WSTEP, служба регистрации CES	wstepCesAdapter.service	8442	HTTPS	Через Nginx (443), путь `/CES_Kerberos/`
Nginx (обратный прокси)	nginx (системная служба)	443	HTTPS	Точка входа для Контрольной панели, сервиса публикации и адаптеров SCEP и MS-WSTEP
Точка распространения (CDP)	nginx (системная служба)	80	HTTP	Напрямую: `http://<fqdn>/cdp/` и `http://<fqdn>/aia/`
PostgreSQL	системная служба PostgreSQL	5432	TCP	Подключения сервисов продукта к своим базам данных

Пояснения к таблице:

- взаимная проверка сертификатов клиента и сервера (mTLS) на порту 9443 после установки выключена; она включается параметром `tls.verify` (см. «TLS и взаимная аутентификация (mTLS)»);

- служба OpenID работает от непривилегированной учётной записи, поэтому для порта 443 установщик выдаёт исполняемому файлу возможность `cap_net_bind_service` (команда setcap);
- службы CEP и CES адаптера MS-WSTEP доступны клиентам через Nginx (443) по адресам `https://<fqdn>/CEP_Kerberos/service.svc/CEP` и `https://<fqdn>/CES_Kerberos/service.svc/CES`;
- сервис резервного копирования слушает также локальный адрес 127.0.0.1, порт `SshAskPass.LocalPort` (по умолчанию 5225): через этот канал askpass узлы ЦС получают пароль для sudo по туннелю SSH, извне порт недоступен.

Закройте прикладные порты на межсетевом экране

Серверная часть Контрольной панели (8407), сервис публикации (9407), архиватор (9507), адаптер SCEP (8440), сервис резервного копирования (в примерах 8409) и службы CEP и CES адаптера MS-WSTEP (8441, 8442) принимают подключения на всех сетевых интерфейсах сервера. Клиенты обращаются к ним через Nginx, а к сервису резервного копирования обращается только серверная часть Контрольной панели, поэтому закройте эти порты для подключений извне. Правила и проверка описаны в разделе «Ограничение сетевого доступа» главы Обеспечение безопасной среды (see page 167).

Порты, которые нужно открыть на межсетевом экране, приведены в главе Требования к среде (see page 38) и в разделе «Приложение Б. Сетевые порты».

2.3 Хранение данных

Состояние ЦС и сервисов хранится в базах данных PostgreSQL, адаптер SCEP использует локальную базу SQLite.

База данных	Что хранит	Кто подключается
БД ЦС, PostgreSQL (по умолчанию `cwica`)	Запросы на сертификаты, сертификаты, списки отзыва, шаблоны, JWT и журнал событий ЦС	Ядро ЦС, OCSP-ответчик mOCSP, архиватор, сервис резервного копирования (см. «Резервное копирование»)
БД OpenID, PostgreSQL (по умолчанию `openid`)	Данные сервиса аутентификации	Сервис OpenID
Архивная БД, PostgreSQL (по умолчанию `cwica_archive`)	Данные, перенесённые архиватором из БД ЦС	Архиватор

Локальная БД адаптера SCEP, SQLite (`/opt/itc/itc/sqlite/scep.db`)	Состояние адаптера, в том числе выданные одноразовые пароли	Адаптер SCEP

Задания резервного копирования хранятся не в БД, а в файлах конфигурации сервиса резервного копирования. Копии лежат в каталоге из его параметра `StoragePath` на сервере сервиса.

2.4 Взаимодействие компонентов

Компоненты обмениваются данными через HTTP(S) API. Пользователи и сервисы подтверждают права по-разному: пользователь предъявляет токен OpenID с правами `rleMiniCA*` , а сервис предъявляет JWT, выпущенный самим ЦС.

Пользователь → Контрольная панель. Браузер загружает веб-интерфейс с Nginx и направляет пользователя на вход в сервис OpenID. OpenID проверяет учётную запись в каталоге LDAP/AD или встроенную учётную запись OpenID и выдаёт токены.

Права пользователя (значения вида `rleMiniCA<Право>`) передаются в токене, в claim `roles` . Права назначаются по членству в группах каталога: карта соответствия групп и прав задаётся в конфигурации OpenID (см. Описание ролевой модели (see page 47)).

Сервисы → ядро ЦС. Серверная часть Контрольной панели, сервис публикации и адаптеры обращаются к ядру ЦС по HTTPS (порт 9443). Каждый сервис предъявляет свой JWT и подписывает запросы своим закрытым ключом. Права сервиса определяет роль в его JWT (см. «Сервисные роли JWT (ролевая модель)»).

У Контрольной панели, сервиса публикации и адаптера SCEP подключение к ЦС описывается в конфигурации блоком `MiniCa` : адрес ЦС, TLS-сертификат, ключ подписи и JWT. Подпись запроса сервиса ядро ЦС проверяет по открытому ключу, к которому JWT сервиса привязан при выпуске; сведения о ключах хранятся в таблице jwt БД ЦС. Консольный клиент mclient тоже предъявляет JWT и подписывает запросы, но его подпись ЦС проверяет по мастер-ключу (параметр `master-key` в minica.yml, по умолчанию `conf/mclient.pem`).

Сервисы → OpenID. Серверная часть Контрольной панели, сервис публикации, адаптер SCEP и сервис резервного копирования проверяют токены пользователей у сервиса OpenID по его адресу издателя (issuer) вида `https://<fqdn>` , порт 443.

Контрольная панель → сервис публикации. Серверная часть Контрольной панели обращается к сервису публикации по адресу из параметра `Services.PublicationCrlUrl` . По умолчанию это `http://localhost:9407` , то есть сервис публикации установлен на том же сервере. Через это подключение Контрольная панель показывает состояние и версию сервиса публикации, список задач публикации и запускает задачи вне расписания. Задачи, изменённые в интерфейсе, Контрольная панель записывает в конфигурацию сервиса публикации.

Контрольная панель → сервис резервного копирования. Запросы раздела «Задачи → Резервное копирование» серверная часть Контрольной панели передаёт сервису резервного копирования по адресу из параметра `Services.BackupUrl`, например `http://localhost:8409`, если сервис работает на том же сервере. Вместе с запросом передаётся токен пользователя, права проверяет сам сервис. Пока адрес не задан, раздел не работает.

Сервис резервного копирования → узлы и БД ЦС. Сервис входит на узлы ЦС по SSH под учётной записью с правом sudo: копирует и восстанавливает файлы, останавливает и запускает службу ЦС. Дамп БД ЦС снимают и восстанавливают утилиты `pg_dump` и `pg_restore` на сервере сервиса. При необходимости сервис обращается к EXC за паролями и к SMTP-серверу для писем о неуспешных операциях. Порты приведены ниже, в описании функций сопровождения.

Сервис публикации → точки распространения. По заданиям сервис публикации записывает CRL в каталог точки распространения CDP, копирует их на удалённый сервер по SCP (по умолчанию порт 22) или публикует в каталог LDAP (по умолчанию порт 389). Сертификаты он публикует в каталог LDAP.

Клиенты → CDP и OCSP. Приложения, которые проверяют сертификаты, загружают CRL и сертификаты ЦС с точки распространения по HTTP (порт 80) и запрашивают статус отдельного сертификата у OCSP-ответчика (порт 8888).

Несколько ЦС в одной Контрольной панели. Каждое подключение к ЦС описывается отдельным блоком `MiniCa.<имя>` в конфигурации серверной части Контрольной панели. Подключение создаёт установщик; администратор может добавить его в интерфейсе («Настройки → Конфигурация») или в файле конфигурации. Пользователь выбирает текущий ЦС на странице «Конфигурация», и все операции Контрольной панели выполняются для выбранного ЦС.

Функции сопровождения. Из Контрольной панели выполняются резервное копирование и восстановление ЦС, ротация ключа ЦС и типовые задачи замены TLS-сертификата на серверах nginx и PostgreSQL. Резервное копирование и восстановление выполняет сервис резервного копирования, Контрольная панель передаёт ему запросы. Типовые задачи включаются параметром `Features.TypicalTask` серверной части Контрольной панели. Подробнее см. разделы «Резервное копирование», «Ротация ключа ЦС» и «Типовые задачи замены сертификатов».

Для этих функций нужен сетевой доступ:

- с сервера Контрольной панели к сервису резервного копирования по адресу из `Services.BackupUrl`, если сервис работает на другом сервере;
- с сервера сервиса резервного копирования к серверу БД ЦС (порт задаётся в задании, по умолчанию 5432);
- с сервера сервиса резервного копирования по SSH к узлам ЦС (порт задаётся в задании, по умолчанию 22), под учётной записью с правом sudo;
- с сервера Контрольной панели по SSH к целевым серверам nginx и PostgreSQL для типовых задач (порт 22).

Если сервис резервного копирования установлен на сервере Контрольной панели, подключения к узлам и БД ЦС идут с этого сервера, но выполняет их сервис резервного копирования. Сама Контрольная панель подключается к серверам по SSH только в типовых задачах.

Синхронизация времени. Ядро ЦС проверяет подпись каждого запроса и JWT с учётом метки времени. Если часы отправителя и ЦС расходятся больше чем на `max-dt-sec` (по умолчанию 600 секунд, то есть окно ±10 минут), запрос отклоняется. Поэтому на всех узлах ЦС и на сервере СУБД настройте синхронизацию времени (NTP). Подробнее см. разделы «Контроль рассогласования времени (max-dt-sec)» и «Синхронизация времени и защита от повторов».

2.5 Схемы развертывания

Компоненты можно разместить на одном сервере или распределить по нескольким узлам. Типовые конфигурации:

Схема	Размещение компонентов	Назначение и особенности
Базовая (1 сервер)	Ядро ЦС, PostgreSQL и нужные сервисы (OpenID, Контрольная панель, OCSP, CDP, при необходимости адаптеры SCEP и MS-WSTEP) устанавливаются на одну машину	Стенды, пилотные проекты и небольшие инсталляции
Без резервирования (2 сервера)	Первый сервер отводится под корневой ЦС, как правило вынесенный и работающий без подключения к сети (офлайн). Второй сервер отводится под выдающий ЦС с сопутствующими сервисами и БД	Разделение снижает риск компрометации ключа корневого ЦС
Распределённая	Каждая функциональная группа (корневой ЦС, выдающий ЦС, аутентификация OpenID, Контрольная панель, публикация, OCSP, архивирование, SCEP, MS-WSTEP, CDP) выносится на отдельные серверы. БД разворачиваются на выделенных узлах PostgreSQL	Масштабирование и изоляция ролей

Установка компонентов

Компоненты устанавливает веб-установщик: он подключается к целевым серверам по SSH и выполняет на них установку. Сам установщик на целевые серверы не устанавливается. Порядок установки описан в главе «Установка».

2.6 Типовые архитектуры внедрения

Компоненты Clearway CA могут размещаться на физических серверах или виртуальных машинах. Вариант установки выбирают с учётом нагрузки, требований к доступности и необходимости раздельного размещения сервисов.

Ниже приведены пять типовых вариантов установки. На схемах обозначение OIDC соответствует сервису идентификации и аутентификации, а CRL, AIA обозначает точки распространения списков отзыва и сертификатов центров сертификации. Компоненты, отмеченные как опциональные, устанавливают при необходимости.

2.6.1 1. Установка на одном сервере

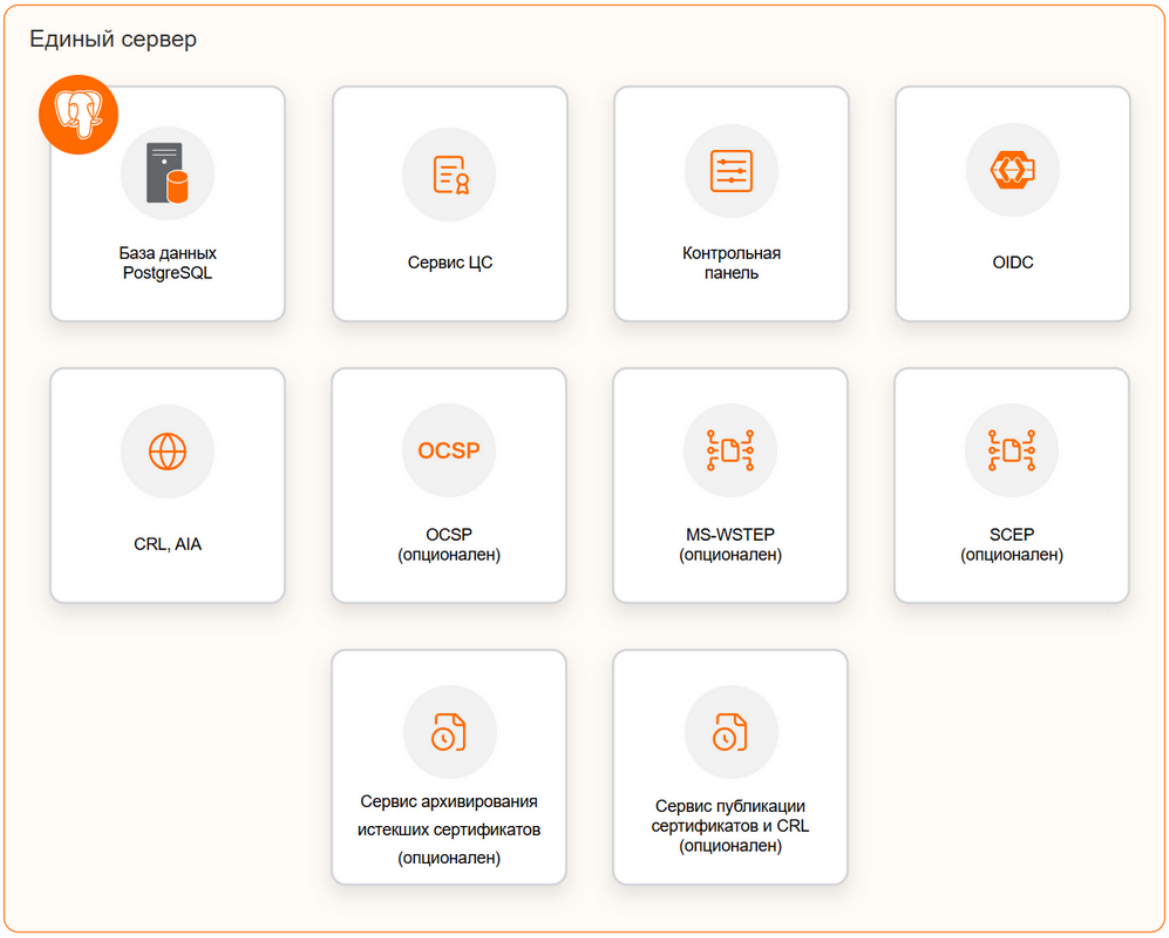

Схема «Один сервер»

Все компоненты установки размещаются на одном сервере: база данных PostgreSQL, сервис центра сертификации, контрольная панель, сервис идентификации и аутентификации, точки распространения CDP/AIA. На этом же сервере размещаются используемые опциональные компоненты: OCSP, MS-WSTEP, SCEP, сервис архивирования истекших сертификатов и сервис публикации сертификатов и списков отзыва.

Этот вариант требует минимального числа серверов и подходит для тестовых стендов и установок с небольшой нагрузкой. Все компоненты используют общие вычислительные ресурсы. Отказ или остановка сервера приводит к недоступности всей установки Clearway CA.

2.6.2 2. Установка на двух серверах

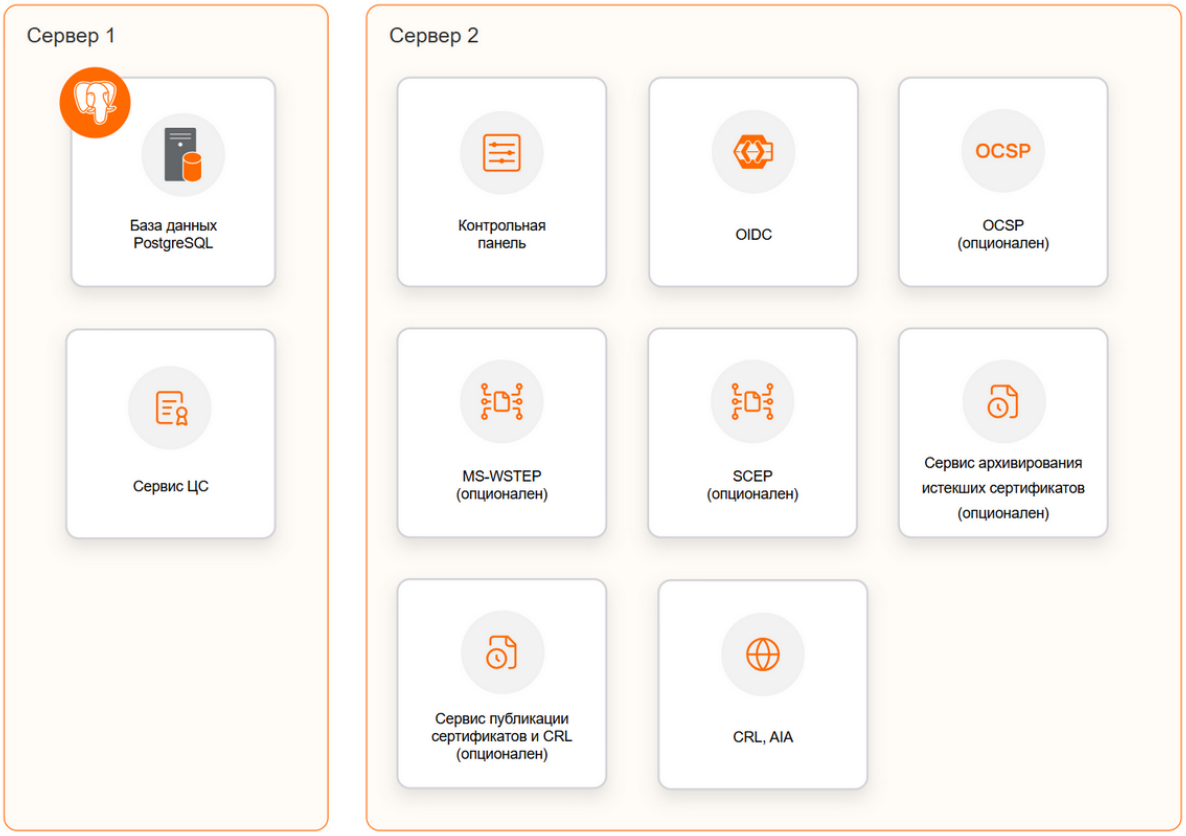

Схема «Два сервера»

Компоненты распределяются между двумя серверами:

- **Сервер 1:** база данных PostgreSQL и сервис центра сертификации.
- **Сервер 2:** контрольная панель, сервис идентификации и аутентификации, точки распространения CDP/AIA, а также используемые опциональные компоненты: OCSP, MS-WSTEP, SCEP, сервис архивирования истекших сертификатов и сервис публикации сертификатов и списков отзыва.

Такое размещение отделяет сервис центра сертификации и его базу данных от пользовательских интерфейсов и вспомогательных сервисов. Вычислительные ресурсы каждого сервера можно подбирать с учётом размещённых на нём компонентов.

Резервирование в этой конфигурации не предусмотрено. Доступность каждого сервиса зависит от работоспособности сервера, на котором он установлен, и доступности связанных с ним компонентов.

2.6.3 3. Установка на трёх серверах

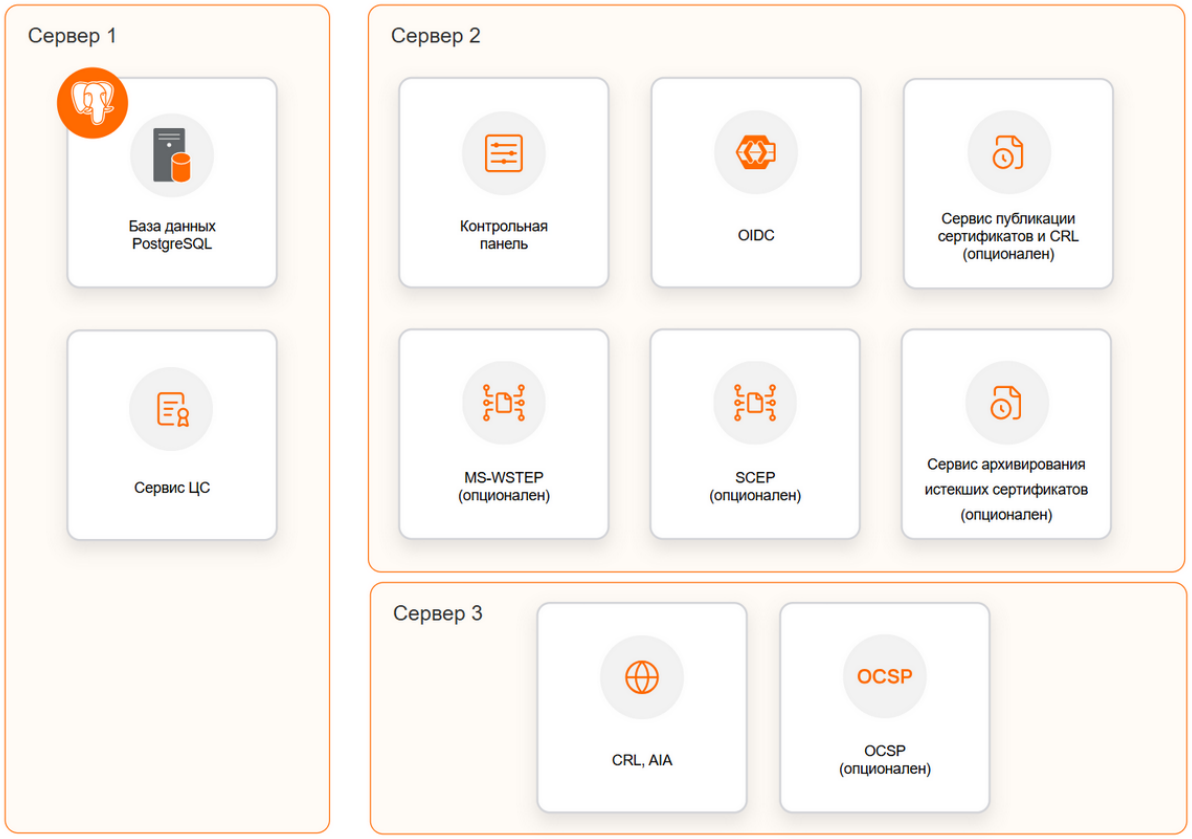

Схема «Три сервера»

В этом варианте точки распространения и сервис проверки статуса сертификатов вынесены на отдельный сервер:

- **Сервер 1:** база данных PostgreSQL и сервис центра сертификации.
- **Сервер 2:** контрольная панель, сервис идентификации и аутентификации, а также используемые опциональные компоненты: MS-WSTEP, SCEP, сервис архивирования истекших сертификатов и сервис публикации сертификатов и списков отзыва.
- **Сервер 3:** точки распространения CDP/AIA и опциональный сервис OCSP.

Выделенный сервер обслуживает обращения потребителей сертификатов за списками отзыва, сертификатами центров сертификации и ответами OCSP. Это позволяет отдельно выделять ресурсы для таких обращений и задавать правила сетевого доступа к точкам распространения.

Сервис публикации остаётся на сервере 2 и доставляет подготовленные сертификаты и списки отзыва на точки распространения. Резервные экземпляры компонентов в этой конфигурации не предусмотрены.

2.6.4 4. Установка на четырёх серверах

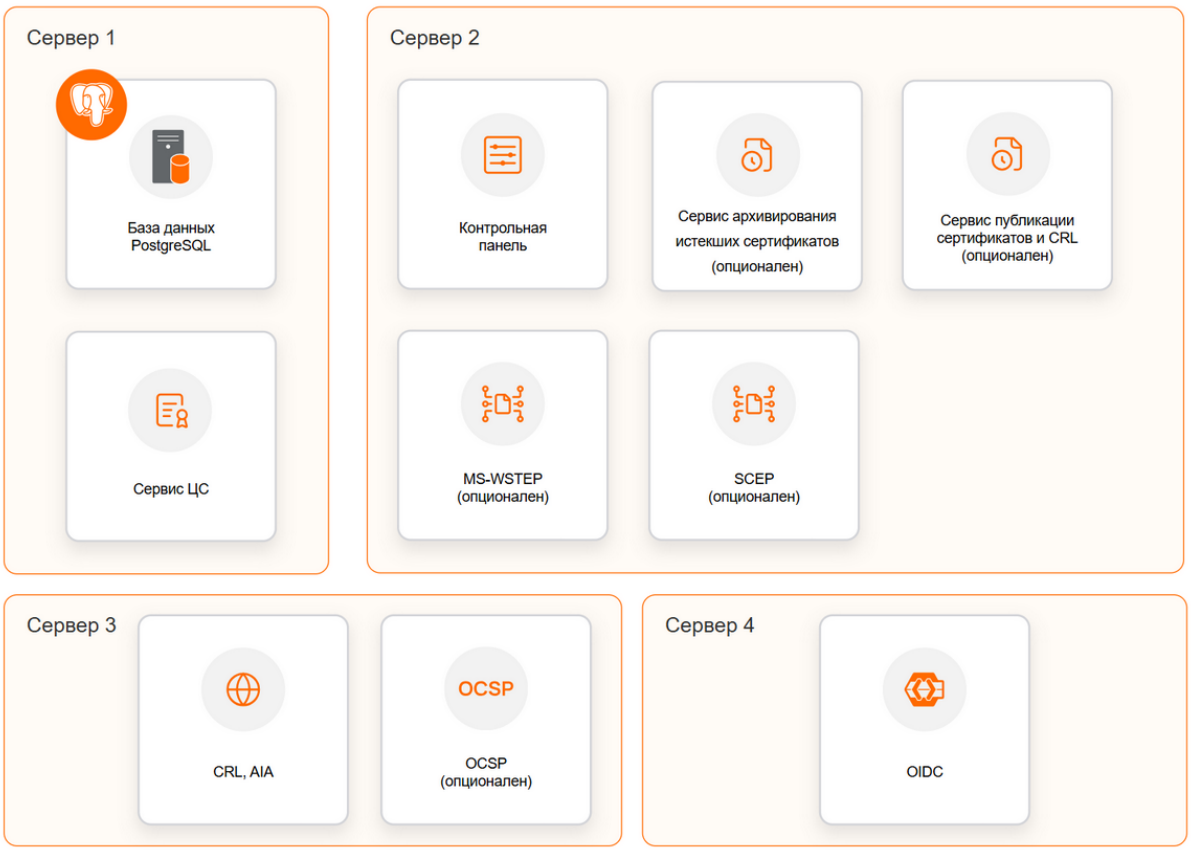

Схема «Четыре сервера»

Сервис идентификации и аутентификации дополнительно выделяется на отдельный сервер:

- **Сервер 1:** база данных PostgreSQL и сервис центра сертификации.
- **Сервер 2:** контрольная панель, а также используемые опциональные компоненты: MS-WSTEP, SCEP, сервис архивирования истекших сертификатов и сервис публикации сертификатов и списков отзыва.
- **Сервер 3:** точки распространения CDP/AIA и опциональный сервис OCSP.
- **Сервер 4:** сервис идентификации и аутентификации OIDC.

Допускается использование существующего в инфраструктуре заказчика сервера OIDC при подтверждённой совместимости с используемой поставкой Clearway CA.

Такое размещение разделяет функции центра сертификации, администрирования, распространения данных и аутентификации. Для сервиса OIDC можно независимо выделять вычислительные ресурсы, настраивать сетевой доступ и организовывать обслуживание.

Распределение компонентов по четырём серверам само по себе не обеспечивает отказоустойчивость: каждый компонент по-прежнему представлен одним экземпляром.

2.6.5 5. Кластерная установка

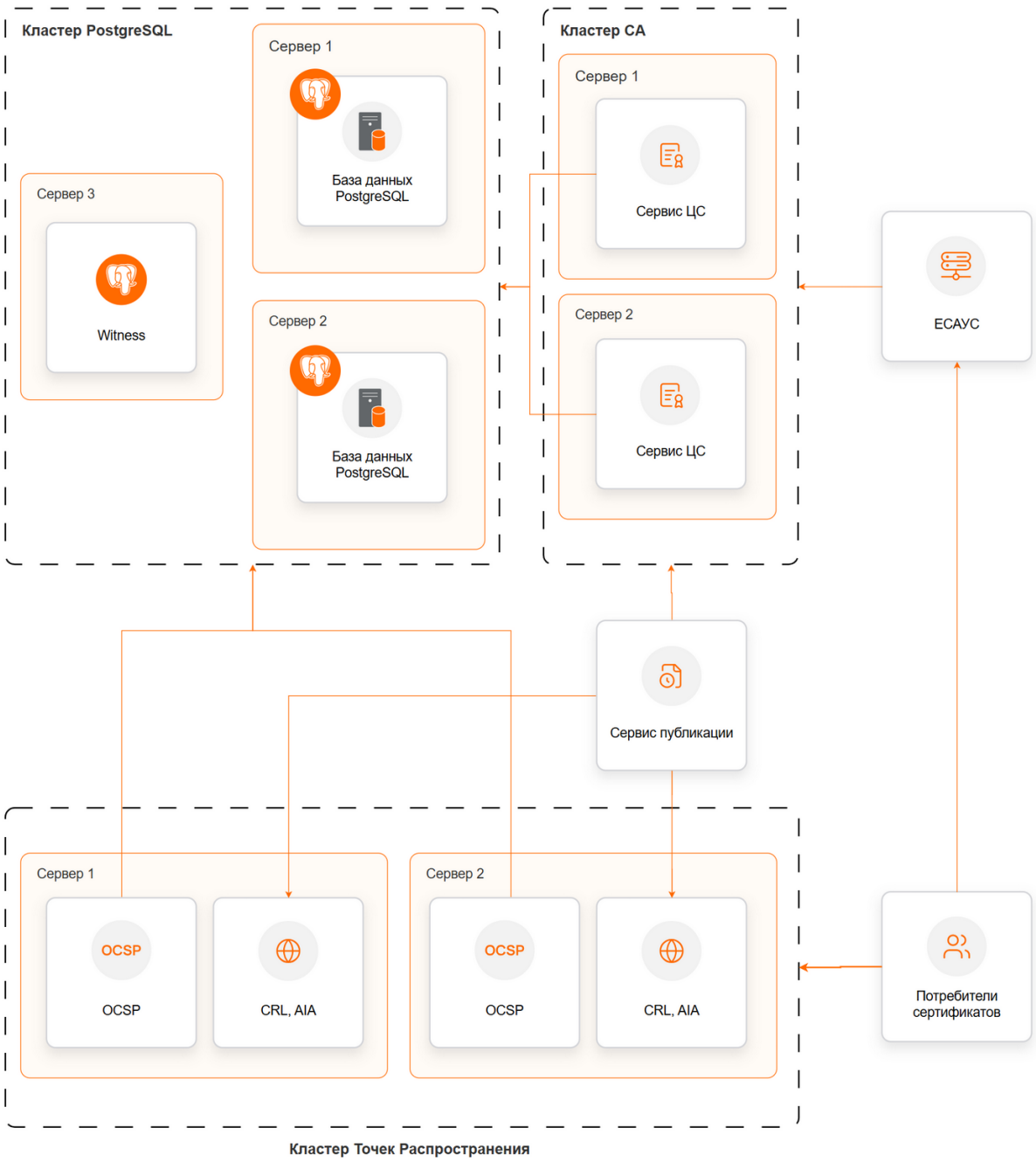

Схема «Кластер»

Кластерный вариант предусматривает резервирование сервисов центра сертификации, базы данных и точек распространения. На схеме выделены три группы узлов:

- **Кластер PostgreSQL:** два сервера базы данных и отдельный сервер Witness. Узел Witness участвует в поддержании кворума и согласовании переключения между узлами базы данных при отказах.

- **Кластер центра сертификации:** два сервера с экземплярами сервиса ЦС, работающими с кластером PostgreSQL. Экземпляры сервиса в кластере представляют один логический ЦС и используют одну ключевую пару и один сертификат ЦС. Способ доступа к ключу и переключения клиентских обращений определяется проектом.
- **Кластер точек распространения:** два сервера, на каждом из которых размещены точки распространения CDP/AIA и сервис OCSP.

Отдельно показан сервис публикации. Он получает сертификаты и списки отзыва от центра сертификации и доставляет их на обе точки распространения. Сервисы OCSP обращаются к кластеру PostgreSQL для получения сведений о статусах сертификатов. Сервис публикации на схеме не резервирован. Его отказ может остановить обновление данных на обеих точках распространения; способ восстановления или резервирования определяется проектом.

На схеме также показано взаимодействие с ЕСАУС, продуктом линейки ЦУГИ. ЕСАУС обращается к кластеру центра сертификации, балансируя нагрузку между узлами. Потребители сертификатов взаимодействуют с ЕСАУС и обращаются к точкам распространения и сервисам OCSP для получения данных, необходимых при проверке сертификатов.

Этот вариант предназначен для установок с повышенными требованиями к доступности. Работа при отказах требует настройки репликации базы данных, переключения между узлами и доступа клиентов к доступным экземплярам сервисов. Распределение запросов к CA обеспечивает ЕСАУС. Распределение запросов к точкам распространения обеспечивается сетевым балансировщиком или механизмом DNS Round Robin.

Схема показывает резервирование основных компонентов. Контрольная панель, MS-WSTEP, SCEP и сервис архивирования на ней не показаны. Они не поддерживают кластеризацию, но могут быть развернуты в горячем резерве, как и сервис публикации. Сервис OIDC также не показан; возможность его кластеризации зависит от выбранного продукта и версии. Способ резервирования и порядок переключения вспомогательных сервисов определяются проектом с учётом возможностей конкретной поставки.

2.7 Технические требования

2.7.1 Рекомендуемые технические требования к серверному оборудованию

В таблице приведены рекомендуемые стартовые конфигурации серверов для типовых вариантов установки Clearway CA. Ресурсы указаны для одного сервера или виртуальной машины. Окончательную конфигурацию определяют с учётом состава компонентов, ожидаемой нагрузки, объёма хранимых данных и требований к доступности.

Вариант установки	Сервер и размещаемые компоненты	Количество серверов	vCPU	ОЗУ, ГБ	SSD, ГБ
Один сервер	Все используемые компоненты Clearway CA и PostgreSQL	1	4	16	256

Вариант установки	Сервер и размещаемые компоненты	Количество серверов	vCPU	ОЗУ, ГБ	SSD, ГБ
Два сервера	Сервер 1: сервис ЦС и PostgreSQL	1	4	8	256
	Сервер 2: контрольная панель, OIDC, CDP/AIA и используемые опциональные сервисы	1	4	8	64
Три сервера	Сервер 1: сервис ЦС и PostgreSQL	1	4	8	256
	Сервер 2: контрольная панель, OIDC, MS-WSTEP, SCEP, архивирование и публикация[1]	1	4	8	64
	Сервер 3: CDP/AIA и OCSP[1]	1	2	4	64
Четыре сервера	Сервер 1: сервис ЦС и PostgreSQL	1	4	8	256
	Сервер 2: контрольная панель, MS-WSTEP, SCEP, архивирование и публикация[1]	1	2	4	64
	Сервер 3: CDP/AIA и OCSP[1]	1	2	4	64
	Сервер 4: сервис идентификации и аутентификации OIDC[2]	1	2	4	64
Кластер	Узел сервиса ЦС	2	4	8	64
	Узел базы данных PostgreSQL	2	4	8	256
	Узел кворума Witness	1	2	4	64
	Узел точек распространения CDP/AIA и OCSP	2	2	4	64
Кластер, дополнительные узлы[3]	Контрольная панель и используемые сервисы MS-WSTEP, SCEP и архивирования	По проекту	2	4	64
	Сервис публикации сертификатов и списков отзыва	По проекту	2	4	64
	Сервис идентификации и аутентификации OIDC[2]	По проекту	2	4	64

[1] Опциональные сервисы устанавливаются при необходимости. Наличие сервиса архивирования в строке не означает, что архивная база данных должна храниться на этом же сервере.

[2] Допускается использование существующего сервера OIDC в инфраструктуре заказчика. В этом случае отдельную виртуальную машину для него не выделяют, а доступные ресурсы проверяют с учётом дополнительной нагрузки Clearway CA. Приведённые значения относятся к выделенному узлу OIDC; ресурсы его базы данных учитывают на сервере, где она размещена.

[3] Размещение и резервирование дополнительных сервисов кластерной установки определяют проектом. При совмещении нескольких ролей ресурсы рассчитывают для их совместной работы. Эти строки дополняют семь серверов, явно показанных на кластерной схеме.

- **Дисковая ёмкость.** Указан стартовый объём для ОС, программного обеспечения и рабочих данных. Его уточняют с учётом роста базы, индексов, журналов PostgreSQL WAL, журналов приложения и опубликованных CRL. Ёмкость для архивной базы и резервных копий рассчитывают отдельно. Заполнение диска с WAL может привести к остановке PostgreSQL.
- **Дисковая производительность.** Для PostgreSQL и Witness требуется учитывать задержки записи и доступные IOPS, включая конкуренцию с другими виртуальными машинами на общем хранилище. Для узла кворума на базе etcd низкая задержка записи существенна для устойчивости кластера.
- **Кластерная установка.** Ресурсы резервируемых узлов рассчитывают так, чтобы оставшийся узел выдерживал требуемую нагрузку при отказе второго. Каждый узел PostgreSQL должен иметь место для полной копии базы. Резервируемые виртуальные машины размещают на разных физических хостах.
- **Проверка под нагрузкой.** При подборе ресурсов учитывают пиковую интенсивность выпуска сертификатов, криптографические алгоритмы и способ выполнения криптографических операций, обращения OCSP, формирование CRL, аутентификацию и фоновые задания.

Резервное копирование должно охватывать базы данных, ключи и сертификаты ЦС, а также конфигурацию компонентов. Репликация базы данных не заменяет резервное копирование. Порядок защищённого хранения копий и проверки восстановления определяется эксплуатационной документацией.

2.7.2 Требования к АРМ оператора

Оператор работает с Clearway CA через веб-интерфейс контрольной панели. Требования к рабочему месту определяются используемой операционной системой, браузером и другими приложениями оператора. Ниже приведена рекомендуемая конфигурация АРМ.

Способ работы	Доступ к контрольной панели Clearway CA через веб-браузер.
Процессор	2 процессорных ядра или 2 vCPU для виртуального рабочего места. Процессор должен соответствовать требованиям выбранной ОС и браузера.

Оперативная память	8 ГБ. При одновременной работе с несколькими ресурсоёмкими приложениями рекомендуется 16 ГБ.
Дисковая подсистема	Рекомендуется SSD. Ёмкость и свободное пространство определяются требованиями ОС, браузера, других установленных приложений и объёмом сохраняемых файлов.
Экран	Рекомендуемое разрешение 1920 × 1080 пикселей для работы с таблицами, формами и журналами контрольной панели.
Операционная система	Поддерживаемая производителем ОС, совместимая с выбранным браузером и допускающая применение установленных в организации средств защиты информации.
Браузер	Google Chrome, Microsoft Edge или Яндекс Браузер в версии, поддерживаемой производителем. Совместимость выбранной версии с используемой поставкой Clearway CA проверяется при внедрении.
Настройки браузера	Разрешены JavaScript, cookies и веб-хранилище для адресов контрольной панели и сервиса идентификации и аутентификации. Политики браузера должны позволять выполнять предусмотренные интерфейсом операции загрузки и скачивания файлов.
Сетевой доступ	Доступ по HTTPS к контрольной панели, сервису идентификации и аутентификации и адресам API, к которым обращается браузер. Имена этих узлов должны разрешаться с рабочего места оператора.
Доверие к сертификатам	Браузер должен доверять сертификатам HTTPS-сервисов, используемых при работе с Clearway CA. Сертификаты корпоративных центров сертификации устанавливаются в соответствующее хранилище доверия ОС или браузера.

- Параметры оборудования и разрешение экрана приведены как рекомендации для рабочего места, а не как минимальные требования Clearway CA.
- На АРМ не требуется выделять место для серверной базы данных Clearway CA. Локальное пространство используется ОС, браузером и файлами, которые оператор сохраняет при работе.
- Для виртуального рабочего места ресурсы выделяются с учётом одновременной работы пользователей и других приложений.

2.7.3 Требования к системному ПО

Состав системного ПО определяется компонентами Clearway CA, размещёнными на сервере, и выбранным вариантом установки. Используются версии, совместимые с конкретной поставкой Clearway CA и выбранной операционной системой. Зависимости опциональных компонентов устанавливаются при включении этих компонентов в состав установки.

Все серверы Clearway CA	Astra Linux	Версия 1.7.5 и выше. Архитектура x86-64, система управления службами systemd.	Среда выполнения компонентов и управление службами.
Серверы прикладных компонентов	Нативные библиотеки Linux	Набор библиотек, необходимый поставляемым исполняемым файлам, включая нативные зависимости .NET. Конкретные пакеты определяются дистрибутивом ОС и составом поставки.	Запуск приложений и работа с системными функциями.
Контрольная панель, архивирование, публикация, SCEP	.NET и ASP.NET Core	В релизе 26.1.0 используется версия 8.0; среда выполнения включена в пакеты компонентов.	Выполнение серверных приложений, реализованных на .NET.
Сервис ЦС	OpenSSL, включая утилиту `openssl`	Версия и криптографические модули должны соответствовать требованиям поставки Clearway CA.	Криптографические операции, обработка запросов и сертификатов, подготовка ключей и TLS-сертификатов.
Сервер или кластер баз данных	PostgreSQL	В качестве базовой версии сохраняется PostgreSQL 15.x. Применение другой версии требует подтверждения совместимости с используемой поставкой.	Хранение данных Clearway CA и размещаемых в этой СУБД баз связанных сервисов.
Узлы установки и обслуживания БД; рабочее место администратора	Клиентские утилиты PostgreSQL: `psql`, `pg_dump`, `pg_dumpall`, `pg_restore`	Рекомендуется комплект той же основной версии, что и сервер PostgreSQL. Устанавливается там, где выполняются операции создания, сопровождения, резервного копирования и восстановления БД.	Администрирование и обслуживание баз данных.
Контрольная панель, CDP/AIA и другие сервисы, использующие веб-прокси	Nginx	Поддерживаемая версия из согласованного репозитория, совместимая с конфигурационными файлами поставки.	Раздача веб-интерфейса и опубликованных файлов, обратное проксирование и обслуживание HTTPS-соединений.

Сервер идентификации и аутентификации	OpenID из поставки Clearway CA либо совместимый сервер OpenID Connect, включая Keycloak	OpenID устанавливается из соответствующей поставки. Для Keycloak или существующего сервера OIDC проверяется совместимость с настройками аутентификации Clearway CA.	Аутентификация, выдача токенов и передача сведений о пользователях и ролях.
Узлы PostgreSQL в кластерной конфигурации	Patroni и его зависимости	Версия, совместимая с выбранными PostgreSQL и etcd. Устанавливаются требуемые Python, драйвер PostgreSQL и дополнительные пакеты.	Управление состоянием узлов PostgreSQL и переключением ролей.
Узлы PostgreSQL и Witness в принятой кластерной схеме	etcd	Версия, совместимая с Patroni и конфигурацией кластера.	Хранение распределённой конфигурации и координация работы кластера.
Узлы выполнения установочных и эксплуатационных скриптов	`curl`, `jq`	Пакеты из репозитория выбранной ОС. Устанавливаются при использовании скриптов, которым они необходимы.	Вызов HTTP API и обработка JSON при настройке и сопровождении.
Узел настройки интеграции OIDC с каталогом	LDAP-клиент, включая `ldapsearch`	Пакет `ldap-utils` или `openldap-clients` в зависимости от ОС.	Проверка подключения к каталогу, параметров поиска и аутентификации.
Серверы, обслуживаемые по SSH	OpenSSH Server	Поддерживаемая версия из репозитория ОС с настройками доступа, принятыми в организации.	Удалённая установка, настройка и сопровождение серверов.
Рабочее место администратора	SSH/SFTP-клиент и средства администрирования БД	Поддерживаемые версии принятых в организации инструментов. Допускаются штатные клиенты ОС, PuTTY, WinSCP, DBeaver и другие совместимые средства.	Администрирование серверов, передача файлов и работа с БД. Конкретные продукты не являются обязательными зависимостями Clearway CA.
АРМ оператора и администратора	Веб-браузер	Согласно подразделу «Требования к АРМ оператора».	Работа с контрольной панелью.

- **Версии зависимостей.** Формулировка «самая последняя версия» не применяется. Используются поддерживаемые версии, совместимость которых подтверждена для выбранной поставки. Возможность установщика обнаружить пакет в репозитории не подтверждает совместимость всех его версий.
- **Обновление .NET.** Если среда выполнения включена в пакет компонента, её обновление выполняется вместе с обновлением этого пакета. Установка новой версии .NET в ОС не обновляет встроенную среду выполнения. Компилятор Go и .NET SDK для запуска готовых компонентов не требуются.
- **Кластерные зависимости.** Patroni и etcd требуются для соответствующего варианта кластеризации. На узле Witness установка сервера PostgreSQL не требуется, если на нём не размещается база данных. Зависимости Patroni устанавливаются в соответствии с его версией. Настройка кластера БД осуществляется самостоятельно Заказчиком.
- **Утилиты PostgreSQL.** Версию средств резервного копирования согласуют с сервером и процедурой восстановления. В частности, `pg_dump` не поддерживает выгрузку сервера с более новой основной версией, чем собственная.
- **Существующий OIDC-сервер.** При использовании инфраструктуры заказчика повторная установка сервиса не требуется. Собственные зависимости Keycloak, включая Java и СУБД, определяются требованиями выбранной версии Keycloak.
- **Опциональные компоненты.** Дополнительные зависимости подключаемых сервисов уточняются по документации соответствующих пакетов.

2.7.4 Требования к сети

Сетевые соединения определяются размещением компонентов Clearway CA и используемыми интеграциями. В таблице указаны инициатор соединения, получатель и порт на стороне получателя. Приведённые порты соответствуют типовым настройкам инфраструктурных сервисов. Фактические адреса, порты и режимы защиты соединений фиксируются при проектировании установки.

АРМ оператора и администратора	Контрольная панель	443/tcp, HTTPS	Работа с веб-интерфейсом и опубликованным API контрольной панели.
АРМ и компоненты, использующие аутентификацию OIDC	Сервис идентификации и аутентификации	443/tcp, HTTPS, либо порт настроенной точки OIDC	Вход пользователей, получение токенов и данных, необходимых для их проверки. Применяется также при использовании существующего OIDC-сервера заказчика.

Серверная часть контрольной панели	Служебный API OpenID из поставки	5559/tcp, HTTP в проверенной конфигурации	Служебное взаимодействие с OpenID. При отдельном размещении OpenID требуется межсерверное соединение. Для Keycloak и других OIDC-серверов этот порт автоматически не применяется.
Серверная часть контрольной панели, сервисы выпуска и публикации, ЕСАУС и другие разрешённые API-клиенты	Сервис ЦС	9443/tcp, HTTPS	Вызов API центра сертификации.
Контрольная панель и разрешённые клиенты API управления заданиями	Сервисы публикации и архивирования	9407/tcp и 9507/tcp соответственно	Управление настройками и заданиями соответствующих сервисов. Межсерверный доступ требуется при раздельном размещении клиента и сервиса.
Устройства и приложения, использующие SCEP	Сервис SCEP	443/tcp, HTTPS в проверенной конфигурации	Выпуск и обновление сертификатов через опубликованную точку SCEP. Только при использовании компонента.
Потребители сертификатов	Точки распространения CDP/AIA	80/tcp, HTTP	Получение CRL, Delta CRL и сертификатов центров сертификации.
Потребители сертификатов	Сервис OCSP	8888/tcp (по умолчанию, может быть изменен)	Проверка статуса сертификатов. Только при использовании OCSP.
Сервис ЦС, OCSP, OIDC, архивирование и другие компоненты с подключением к БД	Соответствующие серверы или точки доступа к PostgreSQL	5432/tcp либо настроенный порт PostgreSQL	Работа с базами данных компонентов. Для архивирования учитываются соединения с рабочей и архивной БД. При размещении на одном узле соединение может оставаться локальным.

Сервис OIDC; сервис публикации при публикации в каталог	Служба каталогов	389/tcp, LDAP, либо 636/tcp, LDAPS	Аутентификация и получение сведений о пользователях и группах; публикация сертификатов и списков отзыва в LDAP. Порт и режим защиты выбираются согласно настройкам интеграции.
Сервис публикации	Серверы точек распространения	22/tcp, SSH/SCP, либо настроенный порт SSH	Передача файлов по SCP при выборе этого способа публикации.
Компоненты, отправляющие почтовые уведомления	Корпоративный почтовый сервер	25/tcp, 587/tcp или 465/tcp согласно настройкам SMTP-подключения	Передача уведомлений почтовой системе. Используются выбранные порт, режим TLS и способ аутентификации.
Узел PostgreSQL, работающий как реплика	Текущий ведущий узел PostgreSQL	5432/tcp либо настроенный порт PostgreSQL	Репликация данных в кластере. Сетевая доступность предусматривается с учётом возможной смены ролей узлов.
Patroni на узлах PostgreSQL	Узлы etcd	2379/tcp либо настроенный клиентский порт etcd	Обращение к распределённой конфигурации и координация работы кластера. Только для соответствующей кластерной конфигурации.
Узлы etcd, включая Witness	Другие узлы того же кластера etcd	2380/tcp либо настроенный порт межузлового обмена	Обмен между участниками etcd и поддержание согласованного состояния.
Узлы Patroni, средства проверки состояния и разрешённые средства администрирования	REST API Patroni на узлах PostgreSQL	Обычно 8008/tcp; HTTP/HTTPS согласно настройкам	Проверка состояния и ролей узлов, взаимодействие Patroni и операции сопровождения. Состав разрешённых клиентов зависит от выполняемых операций.
Балансировщики и средства проверки доступности, если предусмотрены проектом	Обслуживаемые экземпляры сервисов	Порты целевых сервисов и проверок состояния согласно конфигурации	Передача клиентских запросов и проверка доступности узлов. Учитывается отдельно от доступа клиентов к общему адресу сервиса.
Разрешённые узлы установки и администрирования	Серверы Clearway CA	22/tcp, SSH, либо настроенный порт SSH	Установка, настройка, сопровождение и передача файлов.

Разрешённое рабочее место или узел администрирования БД	PostgreSQL	5432/tcp либо настроенный порт PostgreSQL	Администрирование и обслуживание БД при необходимости прямого подключения.
Серверы Clearway CA и APM	Корпоративные DNS-серверы	53/udp и 53/tcp при использовании обычного DNS	Разрешение имён сервисов, серверов БД и инфраструктурных узлов.
Серверы Clearway CA	Принятые в организации источники времени	123/udp при использовании NTP	Синхронизация времени. При другом способе синхронизации применяются требования соответствующей инфраструктуры.

- **Применимость к вариантам установки.** При размещении компонентов на одном сервере внутренние соединения выполняются локально. В вариантах на двух, трёх и четырёх серверах разрешаются соединения между фактически разнесёнными компонентами. В кластерном варианте дополнительно учитываются репликация PostgreSQL, взаимодействие Patroni и etcd, доступ ко всем резервируемым экземплярам сервисов.
- **Внутренние порты приложений.** Nginx обращается к контрольной панели на `localhost:8407`, публикации на `localhost:9407` и SCEP на `localhost:8440`. Эти порты не требуется открывать между серверами при таком размещении. Архивирование использует порт `9507`; доступ к нему определяется размещением клиентов его API. Перенос компонента на другой сервер требует изменения адреса подключения и соответствующего сетевого правила.
- **Совместное размещение OIDC и веб-интерфейса.** При совместном размещении OIDC и веб-интерфейса на одном сервере необходимо согласовать адреса и порты прослушивания.
- **Раздельное размещение OIDC.** Для варианта с четырьмя серверами учитывается доступ к пользовательской точке OIDC и, при использовании OpenID из поставки, соединение контрольной панели со служебным API на порту `5559`. Доступ к служебному API ограничивается его фактическими клиентами.
- **Доставка файлов на точки распространения.** В вариантах с отдельными серверами CDP/AIA предусматривается выбранный способ доставки файлов. Для SCP требуется соединение от сервиса публикации к серверу точки распространения. Если используется смонтированное сетевое хранилище, его соединения описываются отдельно. Порт чтения CDP/AIA не заменяет соединение для доставки файлов.
- **Адреса CDP/AIA и OCSP.** Протокол, имя узла, порт и путь должны соответствовать URL, опубликованным в сертификатах и списках отзыва. Необходимо обеспечить доступ из сетей потребителей сертификатов. Для получения этих данных не должна требоваться. OCSP использует HTTP.
- **Настройка сетевого доступа.** Указание нескольких портов означает варианты подключения или разные сервисы, а не необходимость открыть весь перечень для каждого

узла. Правила задаются для конкретных источников и получателей с разрешением ответного трафика установленных соединений. Порты БД, служебных API и кластерного обмена предоставляются только участвующим в соответствующем взаимодействии узлам.

- **Дополнительные соединения.** Доступ к репозиториям обновлений, системам мониторинга, резервного копирования и другим средствам эксплуатации определяется принятыми в организации решениями. Пропускная способность и допустимые задержки рассчитываются с учётом клиентской нагрузки, размеров CRL, объёма публикации и репликации БД.

2.8 Технологический стек

Сведения о платформе нужны при подготовке серверов и оценке внешних зависимостей.

- Сетевой обмен: HTTP(S), TLS и mTLS; на границе работает Nginx как обратный прокси (порт 443 для веб-сервисов, порт 80 для CDP).
- Криптография: в стандартном варианте ядро ЦС выполняет криптографические операции через утилиту командной строки openssl (OpenSSL).
- Криптопровайдер КриптоПро CSP: предусмотрен вариант установки Clearway CA, в котором криптографические операции выполняет КриптоПро CSP.
- Хранение ключа ЦС на аппаратных носителях и HSM: опционально, через интерфейс PKCS#11 (OpenSSL engine pkcs11); в бесплатной редакции недоступно (см. «Бесплатная и платная редакции продукта»).
- Ядро ЦС, OCSP-ответчик mOCSP и консольный клиент mclient: язык Go.
- Контрольная панель, сервис публикации, архиватор, сервис резервного копирования и адаптер SCEP: C#/.NET 8, самодостаточные сборки со встроенной средой выполнения (отдельно устанавливать .NET на серверы не нужно).
- Веб-интерфейс: одностраничное приложение (SPA) на Angular.
- Внешние компоненты: СУБД PostgreSQL версий 11–18; Nginx как веб-сервер и обратный прокси; встроенная СУБД SQLite для адаптера SCEP.

3 Требования к среде

Глава перечисляет, что нужно подготовить для установки и эксплуатации Clearway CA 26.3.0: операционную систему и ресурсы серверов, системное ПО, СУБД PostgreSQL, рабочие места администратора и оператора, сетевые порты и технологическую учётную запись. Сверьте серверы и рабочие места с этими требованиями до запуска установщика.

Операционная система

Серверные компоненты Clearway CA устанавливаются на 64-разрядные ОС семейства Linux. Поддерживаются:

ОС	Версия
Astra Linux Special Edition (РУСБ.10015-01)	очередное обновление 1.8
РЕД ОС	8
Альт 8 СП	релиз 10

Установщик автоматически определяет семейство ОС целевого сервера: Альт (altlinux), РЕД ОС (redos) или семейство Debian, к которому относится и Astra Linux. Затем он выбирает менеджер пакетов: dnf на РЕД ОС, apt на остальных ОС.

На всех узлах с компонентами Clearway CA и на сервере СУБД настройте единое системное время по протоколу NTP. ЦС проверяет метки времени в запросах с допуском `max-dt-sec` = 600 секунд, то есть ±10 минут: если часы расходятся сильнее, запрос отклоняется. Подробнее см. раздел «Синхронизация времени и защита от повторов» главы «Обеспечение безопасной среды».

Аппаратные требования

Конфигурации виртуальных ресурсов (vCPU / vRAM / vHDD) ниже рассчитаны на разную нагрузку, то есть на разное число сертификатов, которые выпускаются в сутки.

В минимальной конфигурации все компоненты Clearway CA и СУБД PostgreSQL работают на одном сервере:

Узел	Компоненты	vCPU / vRAM / vHDD	Нагрузка
Единый сервер	Все компоненты + СУБД	4 / 4 ГБ / 256 ГБ	до 100 сертификатов в сутки

В базовой конфигурации для повышенной нагрузки все компоненты и СУБД также работают на одном сервере:

Узел	Компоненты	vCPU / vRAM / vHDD	Нагрузка
Единый сервер	Все компоненты + СУБД	4 / 8 ГБ / 512 ГБ	до 5000 сертификатов в сутки

В типовой конфигурации компоненты разнесены по отдельным узлам:

Узел	Компоненты	vCPU / vRAM / vHDD	Системное ПО
Центр сертификации	Сервис управления сертификатами (ядро ЦС)	2 / 2 ГБ / 80 ГБ	PostgreSQL
Аутентификация	Сервис аутентификации OpenID	2 / 2 ГБ / 40 ГБ	PostgreSQL
Вспомогательные сервисы	Контрольная панель, сервис публикации, архиватор, OCSP, SCEP, CDP	2 / 4 ГБ / 40 ГБ	PostgreSQL, nginx, SMTP-сервер

Место под журналы

Каталог журналов ЦС `/opt/itc/minica/logs` рекомендуется размещать на разделе с запасом свободного места и контролировать его заполнение средствами ОС. То же относится к каталогам журналов OCSP-ответчика (`/opt/itc/ocsp/logs`) и .NET-сервисов (подкаталоги `/opt/itc/itc/logs`).

Место под резервные копии

Сервис резервного копирования хранит копии на своём сервере, в каталоге из параметра `StoragePath` . Каждая копия содержит полный дамп БД ЦС и файлы узлов ЦС, а хранятся копии за весь срок `RetentionDays` (по умолчанию 30 дней). Размещайте каталог на разделе с таким запасом места. Если свободно меньше `MinFreeSpaceBytes` (по умолчанию 1 ГиБ), операция не начинается.

Системное ПО

На серверных узлах устанавливается следующее ПО из репозитория ОС или из состава поставки. Большую часть пакетов установщик проверяет и при необходимости ставит сам на шаге проверки пререквизитов (см. «Пререквизиты» в главе «Установка»).

ПО	Минимальная версия	Назначение
nginx	из репозитория ОС	Обратный прокси с TLS для Контрольной панели, сервиса публикации, адаптеров SCEP и MS-WSTEP; раздача файлов точки распространения CDP по HTTP
OpenSSL	из репозитория ОС	На узлах ЦС: криптографические операции ядра ЦС, генерация ключей, запросов (CSR) и сертификатов. Нужен также на сервере адаптера MS-WSTEP
jq	из репозитория ОС	Обработка JSON при формировании токенов и настройке сервисов: запись подключения к ЦС в конфигурацию Контрольной панели, адаптеров SCEP и MS-WSTEP
curl	из репозитория ОС	Обращения к API системы из скриптов
psql (клиент PostgreSQL)	не ниже версии сервера БД	Создание и обслуживание БД и таблиц
pg_dump и pg_restore (клиент PostgreSQL)	той же основной версии, что сервер БД ЦС	На сервере сервиса резервного копирования: снятие и восстановление дампа БД ЦС. Без этих утилит сервис не запускается
Клиент LDAP (`ldapsearch`)	из репозитория ОС	Проверка подключения к каталогу LDAP/ Active Directory при интеграции OpenID с каталогом и установке адаптера MS-WSTEP. Пакет openldap-clients на Альт и РЕД ОС, ldap-utils на остальных ОС
realmd, sssd, adcli, клиент Kerberos	из репозитория ОС	Ввод сервера адаптера MS-WSTEP в домен Active Directory

Среду выполнения .NET отдельно устанавливать не нужно. Контрольная панель, сервис публикации, архиватор, сервис резервного копирования и адаптер SCEP поставляются самодостаточными сборками .NET 8 со встроенной средой выполнения. Ядро ЦС, OCSP-ответчик mOCSP и консольный клиент mclient представляют собой исполняемые файлы на языке Go.

На РЕД ОС вместе с nginx установщик разрешает ему сетевые подключения в SELinux (параметр `httpd_can_network_connect`): без этого nginx не может передавать запросы службам. При ручной установке на РЕД ОС выполните на сервере с nginx команду `sudo setsebool -P httpd_can_network_connect 1`.

СУБД PostgreSQL

В PostgreSQL хранится состояние ЦС, сервиса аутентификации OpenID и архиватора.

- Поддерживаются версии PostgreSQL 11–18; минимально допустимая версия: 11.
- При установке мастер подбирает наибольшую версию, доступную в репозитории ОС сервера, перебирая от 18 к 11; если ни одна не обнаружена, используется запасное значение 16.
- Версию можно выбрать вручную в дополнительных настройках шага «База данных».
- Та же логика применяется к БД сервиса аутентификации OpenID.

Для отказоустойчивого развертывания PostgreSQL поддерживается работа в кластере на базе etcd и patroni. Их устанавливают из репозитория ОС на узлах СУБД.

Шифрование подключений к БД

По умолчанию сервисы подключаются к PostgreSQL без TLS: у ядра ЦС это разрешает параметр `database.allow-without-tls: true`. Для защищённого состояния переведите канал между сервисом и БД на TLS; для ядра ЦС задайте `database.allow-without-tls: false` и добавьте `sslmode=verify-full` в строку подключения. Порядок действий для всех сервисов описан в разделе «Защита подключений к СУБД и каталогу LDAP» главы «Обеспечение безопасной среды».

АРМ администратора и оператора

Рабочее место: ПК под управлением любой ОС, которая поддерживает рекомендуемый браузер и допускает применение мер защиты информации. Для работы с аппаратными носителями требования строже, они приведены ниже.

Параметр	Значение
Ядер / частота	2 / не менее 2 ГГц
RAM	4 ГБ
Диск	80 ГБ SSD
Браузер	Google Chrome версии не ниже 113 или браузер на базе Chromium (Microsoft Edge, Яндекс.Браузер)

Дополнительное ПО для АРМ администратора:

ПО	Минимальная версия	Назначение

DBeaver (+ утилиты pg_dump, pg_dumpall, pg_restore, psql, драйвер PostgreSQL)	21.3.1	Настройка и обслуживание БД
WinSCP	5.13.7	Доступ к файлам серверов
PuTTY	0.70	Доступ к серверам по SSH
Google Chrome	113	Доступ к веб-интерфейсу системы

Установщик запускается на рабочем месте оператора, а его интерфейс открывается в браузере на этом же компьютере по адресу `https://localhost:5224/installer`. С рабочего места оператора нужен доступ по SSH ко всем целевым серверам (см. «Установщик» в главе «Установка»).

Рабочее место для работы с аппаратными носителями

Раздел Контрольной панели «Аппаратные токены» и конструктор CSR-запросов работают с носителями Рутокен, JaCarta и с КриптоПро CSP: проверяют установленные компоненты, показывают содержимое носителя, создают на нём ключ и запрос. Операции выполняются локально в браузере пользователя, поэтому компоненты для работы с носителями устанавливают на рабочие места, а не на серверы Clearway CA.

Работа с носителями проверена в ОС Windows в браузерах на основе Chromium: Google Chrome, Microsoft Edge, Chromium, Яндекс.Браузер. На другой платформе раздел выводит предупреждение «Платформа не входит в проверенную матрицу…», и работа не гарантируется.

Носитель	Что установить на рабочем месте	Условия и версии
Рутокен	Расширение браузера «Адаптер Рутокен Плагин» и локальный компонент Рутокен Плагин	Установленные компоненты и версию плагина показывает карточка «Рутокен» в разделе «Аппаратные токены»
JaCarta	Локальный компонент JC-WebClient. Расширение браузера не требуется	Контрольная панель обращается к JC-WebClient по адресу `https://localhost:24738`
КриптоПро CSP	КриптоПро CSP с действующей лицензией и КриптоПро Browser plug-in (расширение браузера и локальный компонент)	КриптоПро CSP версии 5.0 или новее; проверена версия 5, для более новых версий карточка выводит предупреждение. Сайт Контрольной панели добавлен в доверенные узлы КриптоПро и открыт по HTTPS

Если компонент не найден, карточка носителя в разделе «Аппаратные токены» показывает, чего не хватает, и даёт ссылку на официальную инструкцию по установке. Порядок работы с носителями описан в Руководстве пользователя, раздел «Аппаратные токены».

> **Заголовок Content-Security-Policy**
>
> Если в конфигурацию Nginx Контрольной панели добавляется заголовок Content-Security-Policy, разрешите в нём загрузку скриптов с адреса `https://localhost:24738`. Иначе браузер заблокирует обращение к JC-WebClient, и носители JaCarta будут недоступны.

Сетевые порты

Порты ниже используются в правилах межсетевого экрана. Веб-сервисы доступны клиентам через Nginx по HTTPS (порт 443). Правила по направлениям подключения собраны в разделе «Приложение Б. Сетевые порты».

Служба / узел	Порт/протокол	Назначение
Nginx на сервере служб	443/tcp	HTTPS: доступ пользователей к Контрольной панели, обращения к сервису публикации и адаптерам SCEP и MS-WSTEP
Nginx как точка распространения (CDP)	80/tcp	HTTP: загрузка CRL, Delta CRL и сертификатов ЦС приложениями, которые проверяют сертификаты
OCSP-ответчик mOCSP	8888/tcp	HTTP: запросы клиентов OCSP о статусе сертификата. Служба доступна клиентам напрямую, без Nginx
Центр сертификации (ЦС)	9443/tcp	HTTPS-интерфейс ЦС: обращения серверных частей Контрольной панели, сервиса публикации, адаптеров SCEP и MS-WSTEP и консольного клиента mclient. Взаимная аутентификация (mTLS) включается отдельно. Разрешите подключения только с серверов этих компонентов
Аутентификация (OpenID)	443/tcp	HTTPS OIDC-провайдера: вход пользователей, проверка токенов серверными частями Контрольной панели, сервиса публикации, адаптера SCEP и сервисом резервного копирования

Аутентификация (OpenID)	5559/tcp	Служебный HTTP-интерфейс администрирования (параметр `adminServer.http`). Клиентам не открывается; при необходимости разрешите подключение только с АРМ администраторов
Сервер БД	5432/tcp	Подключения к PostgreSQL. К БД ЦС подключаются ядро ЦС, mOCSP, архиватор и сервер сервиса резервного копирования (для заданий резервного копирования, порт задаётся в задании), к архивной БД архиватор, к БД OpenID сервис OpenID. Администраторы подключаются для настройки и сопровождения
Серверы продукта (ОС)	22/tcp	Администрирование по SSH (OpenSSH) с АРМ администраторов; установка компонентов с рабочего места оператора
Узлы ЦС	22/tcp	SSH-подключения с сервера сервиса резервного копирования для заданий резервного копирования; порт задаётся в задании
Серверы nginx и PostgreSQL	22/tcp	SSH-подключения с сервера Контрольной панели для типовых задач замены сертификатов
Сервис резервного копирования	Порт из параметра запуска `--urls`, в примерах 8409/tcp	HTTP: запросы серверной части Контрольной панели по адресу из `Services.BackupUrl`. Разрешите подключение только с сервера Контрольной панели
Серверы-получатели SCP	22/tcp	Публикация CRL по SCP с сервера публикации; порт задаётся в задаче публикации, по умолчанию 22
Каталог LDAP/Active Directory	389/tcp, 636/tcp	Подключение OpenID к каталогу (LDAP / LDAPS) и публикация в каталог LDAP сервисом публикации (по умолчанию порт 389). После установки OpenID подключается к каталогу по порту 389 без шифрования; рекомендуется перейти на LDAPS (636)
Контроллер домена Active Directory	389/tcp; 88/tcp и 88/udp (Kerberos)	Работа адаптера MS-WSTEP в домене: ввод сервера в домен, обращения к каталогу и проверка подлинности Kerberos

SMTP-сервер	порт из `log.secLog.notifyEmailSmtp`; порт из `Mailing.Port` сервиса резервного копирования	Уведомления журнала событий безопасности по почте от ядра ЦС и письма сервиса резервного копирования о неуспешных операциях, если они включены
Веб-интерфейс установщика	5224/tcp	HTTPS-интерфейс установщика на рабочем месте оператора. Разрешите подключение только с АРМ администратора и только на время установки

Закройте прикладные порты служб

Серверная часть Контрольной панели (8407), сервис публикации (9407), архиватор (9507) и адаптер SCEP (8440) принимают подключения по HTTP на всех сетевых интерфейсах сервера, а службы CEP и CES адаптера MS-WSTEP (8441, 8442) принимают их по HTTPS. Клиенты обращаются к ним через Nginx, поэтому закройте эти порты межсетевым экраном для подключений извне. Если сервис публикации установлен не на сервере Контрольной панели, разрешите подключение к порту 9407 только с этого сервера. Порт сервиса резервного копирования (в примерах 8409) закрывается так же: к нему обращается только серверная часть Контрольной панели. Порт 9080 (ядро ЦС без TLS) после установки не используется: установщик включает TLS. Порядок настройки и проверки описан в разделе «Ограничение сетевого доступа» главы «Обеспечение безопасной среды».

Требования к учётным записям

Прикладные службы Clearway CA работают от имени **технологической учётной записи**, владельца каталога установки (по умолчанию `itc-svc`), а не от root. Nginx и PostgreSQL работают как системные службы ОС.

По умолчанию каталогом установки служит `/opt/itc`; ЦС устанавливается в `/opt/itc/minica`, OCSP-ответчик в `/opt/itc/ocsp`, CDP в `/opt/itc/web`. Владельцем каталога установки должна быть технологическая учётная запись.

Службы запускаются как **пользовательские юниты systemd** (`systemctl --user`), поэтому для технологической учётной записи обязательны:

- режим сохранения сеанса linger (`loginctl enable-linger <учётная запись>`): без него пользовательские юниты не запускаются при отсутствии интерактивного входа и останавливаются, когда завершается последний сеанс учётной записи;
- окружение для управления юнитами: переменные XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS, прописанные в файле `~/.bashrc` учётной записи.

Проверка и включение режима linger. Команды выполняет администратор на каждом сервере с компонентами Clearway CA; `sudo` нужен только для включения.

```
# Проверка режима linger для технологической учётной записи
loginctl show-user itc-svc | grep Linger
# ожидается Linger=yes
# Включение, если выведено Linger=no
sudo loginctl enable-linger itc-svc
```

Кроме технологической учётной записи понадобятся:

- на каждом целевом сервере учётная запись с правами sudo и входом по SSH (по паролю или SSH-ключу): через неё установщик устанавливает пакеты, создаёт каталоги и технологическую учётную запись;
- на узлах ЦС учётная запись с правом sudo и входом по SSH с паролем для заданий резервного копирования (вход по ключу не поддерживается); sudo должен принимать её собственный пароль или работать без пароля, параметр `requiretty` выключен, в sshd разрешён проброс TCP (см. «Резервное копирование», подраздел «Подготовка узлов ЦС и сети»);
- на серверах nginx и PostgreSQL, где сертификаты заменяются типовыми задачами, учётная запись с правом изменять файлы сертификата и ключа и перезапускать сервис, со входом по SSH с паролем (см. «Типовые задачи замены сертификатов»).

На этапе контроля пререквизитов установщик проверяет технологическую учётную запись, каталог установки и его владельца, режим linger, наличие OpenSSL, Nginx, jq и клиента LDAP и доступность PostgreSQL. Если условие не выполнено, он предлагает автоматическое исправление. Переменные окружения XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS установщик прописывает в `~/.bashrc` сам.

4 Описание ролевой модели

Глава описывает права доступа Clearway CA, разделы Контрольной панели, которые они открывают, и стандартные роли. По ней администратор назначает пользователям роли и проверяет, что у каждого есть только нужные полномочия.

Каждое право разрешает один вид действий: суффикс `-r` обозначает право на просмотр данных определённого типа, суффикс `-w` обозначает право на их изменение. Права не назначаются пользователям поштучно. Пользователя включают в группу каталога Active Directory или LDAP, а группе в конфигурации сервиса аутентификации OpenID сопоставлен набор прав (см. Назначение ролей через группы AD/LDAP (see page 196)).

Как права доходят до пользователя:

1. Пользователь входит в Контрольную панель, вход выполняет сервис аутентификации OpenID.
2. OpenID получает из каталога группы, в которые входит учётная запись.
3. По карте `ldapGroupRolesMap` из файла `/opt/itc/openid/config.yaml` OpenID составляет перечень прав и передаёт его в поле (claim) `roles` токена, то есть подписанного набора сведений о пользователе, который OpenID выдаёт при входе.
4. Контрольная панель, ядро ЦС и сервис резервного копирования проверяют права из токена; интерфейс показывает только разделы и кнопки, разрешённые этим перечнем.

Перечень прав в токене составляется при входе. Поэтому изменение членства в группах или карты `ldapGroupRolesMap` вступает в силу, только когда пользователь выйдет из Контрольной панели и войдёт снова. После правки карты, кроме того, перезапускают сервис OpenID (см. Назначение ролей через группы AD/LDAP (see page 196)).

Свой перечень прав пользователь видит в Контрольной панели: имя пользователя в развёрнутом левом меню → «Профиль» → блок «Права».

Имя claim с правами

По умолчанию права передаются в claim `roles`. Если провайдер OpenID Connect (OIDC) передаёт их в claim с другим именем, это имя задают параметр `OpenId.JwtRoleClaim` в конфигурации серверной части Контрольной панели и параметр `rolesClaim` в файле `client_env.js` (см. «Контрольная панель» в разделе Настройка вспомогательных сервисов (see page 139)).

Служебные компоненты Clearway CA и утилита `mclient` обращаются к ЦС не с правами пользователей, а с токенами JWT сервисных ролей (`admin` , `causer` , `crluser` и другие). Эти роли настраиваются в ядре ЦС: см. «Сервисные роли JWT (ролевая модель)» в разделе Настройка ядра ЦС (see page 126).

4.1 Перечень прав

Раздел перечисляет все права Clearway CA 26.3.0 и то, что каждое из них открывает в Контрольной панели.

Определено 17 прав с идентификаторами вида `rleMiniCA<Право>`. Идентификатор передаётся значением claim `roles` и проверяется компонентом, который выполняет действие: Контрольной панелью, ядром ЦС или сервисом резервного копирования. Четыре права добавлены в версии 26.3.0: `rleMiniCABackup-r`, `rleMiniCABackup-w`, `rleMiniCABackupRestore-w` и `rleMiniCARotation-w`.

Право	Что разрешает
`rleMiniCACertList-r`	Просмотр выпущенных сертификатов: реестр и карточки сертификатов, конструктор CSR-запросов, список отозванных сертификатов на странице «Информация о CRL и Delta CRL»
`rleMiniCAEventList-r`	Просмотр журнала событий, в том числе вкладки «События» в карточке сертификата
`rleMiniCATemplList-r`	Просмотр шаблонов сертификатов: реестр и карточки шаблонов, экспорт шаблонов
`rleMiniCAReqList-r`	Просмотр запросов на выпуск: реестр и карточки запросов
`rleMiniCACRL-r`	Просмотр списков отзыва (CRL и Delta CRL), скачивание списков
`rleMiniCAConfig-r`	Просмотр конфигурации ЦС: страницы «Настройки → Конфигурация» и «Статистика ЦС», скачивание цепочки сертификатов
`rleMiniCACertReq-w`	Создание запроса на выпуск сертификата: одиночный и пакетный запрос, выпуск из конструктора CSR-запросов; типовые задачи замены сертификата nginx и postgres, если функция типовых задач включена
`rleMiniCACertRev-w`	Отзыв сертификатов
`rleMiniCACertUnRev-w`	Снятие отзыва: восстановление сертификатов, приостановленных с причиной CertificateHold
`rleMiniCATempl-w`	Создание, изменение, копирование, удаление и импорт шаблонов сертификатов

`rleMiniCACRL-w`	Формирование и публикация списков отзыва: обновление CRL; группа меню «Задачи» (списки и карточки задач публикации, запуск задач публикации вне расписания)
`rleMiniCAConfig-w`	Изменение конфигурации: добавление ЦС и изменение подключения к ЦС («Добавить новый ЦС», «Обновить конфигурацию» на странице «Настройки → Конфигурация»); добавление, изменение и удаление задач публикации; создание, изменение и удаление заданий резервного копирования (кнопки «Добавить задание», «Изменить» и «Удалить» в разделе «Задачи → Резервное копирование»)
`rleMiniCAReqApprove-w`	Одобрение и отклонение запросов на выпуск
`rleMiniCABackup-r`	Просмотр в разделе «Задачи → Резервное копирование»: список заданий, карточка задания, ход текущей операции, последний запуск, список резервных копий
`rleMiniCABackup-w`	Запуск резервного копирования вне расписания (кнопка «Запустить копирование» в карточке задания)
`rleMiniCABackupRestore-w`	Восстановление ЦС из резервной копии (кнопка «Восстановить» у пригодных копий в карточке задания и окно «Восстановление копии» с числом записей, которые будут удалены)
`rleMiniCARotation-w`	Все действия на странице «Настройки → Ротация ключей»: просмотр действующего сертификата ЦС и состояния ротации, формирование запроса на сертификат ЦС (CSR) в режиме «Продление» или «Перевыпуск», скачивание CSR, загрузка и установка нового сертификата ЦС

Права резервного копирования проверяет сервис резервного копирования: Контрольная панель передаёт ему запросы раздела «Задачи → Резервное копирование» вместе с токеном пользователя. Создание, изменение и удаление задания сервис разрешает только при праве `rleMiniCAConfig-w`: права `rleMiniCABackup-r`, `rleMiniCABackup-w` и `rleMiniCABackupRestore-w` для этого не подходят.

Отдельного права на просмотр страницы «Ротация ключей» нет: даже просмотр состояния ротации требует права `rleMiniCARotation-w`.

4.1.1 Права и пункты меню Контрольной панели

Пункт меню отображается, только если у пользователя есть указанные права. Таблица помогает понять, почему пользователь не видит нужный раздел.

Пункт меню	Необходимые права
Сертификаты	`rleMiniCACertList-r`
Шаблоны	`rleMiniCATemplList-r`
Запросы	`rleMiniCAReqList-r`
События	`rleMiniCAEventList-r`
Информация о CRL	`rleMiniCACRL-r`
Статистика ЦС	`rleMiniCAConfig-r`
Задачи → Публикация сертификатов, Публикация CRL	`rleMiniCACRL-w`
Задачи → Резервное копирование	`rleMiniCACRL-w` и `rleMiniCABackup-r`
Задачи → Типовые (замена сертификата nginx и postgres)	`rleMiniCACRL-w` и `rleMiniCACertReq-w`; функция типовых задач включена параметром `Features.TypicalTask` конфигурации Контрольной панели
Настройки → Конфигурация	`rleMiniCAConfig-r`
Настройки → Ротация ключей	`rleMiniCARotation-w`
Аппаратные токены	не требуются
Профиль (меню пользователя: имя → «Профиль»)	не требуются

Операции с аппаратными носителями выполняются в браузере пользователя, правами ЦС не разграничиваются и в журнал событий ЦС не попадают.

Правила отображения:

- Группа «Задачи» вместе со всеми вложенными пунктами, включая «Резервное копирование» и «Типовые», отображается только при праве `rleMiniCACRL-w`.

Учитывайте это при составлении ролей: право `rleMiniCACRL-w` также разрешает обновлять CRL и запускать задачи публикации.

- Группа «Настройки» отображается при наличии хотя бы одного из прав `rleMiniCAConfig-r` и `rleMiniCARotation-w`.
- Кнопка, на которую у пользователя нет права, скрыта.
- Исключение составляет меню «Действия» на страницах «Информация о CRL и Delta CRL» и «Настройки → Конфигурация»: действие без права остаётся в меню бледным и не выполняется.

4.2 Составные роли-маркеры

Кроме прав, в claim `roles` можно передать три маркера роли. Маркером называется отдельное значение claim `roles`, которым помечают учётную запись администратора ЦС.

Маркер	Назначение
`rleMiniCAOperator`	Признак оператора ЦС
`rleMiniCAAuditor`	Признак аудитора
`rleMiniCAMaster`	Признак администратора с полными полномочиями

Маркеры не дают прав

Маркеры не расширяют перечень операций: доступ определяется только правами `rleMiniCA*-r` и `rleMiniCA*-w`, интерфейс Контрольной панели маркеры не проверяет. Маркер служит признаком роли учётной записи и виден в блоке «Права» профиля пользователя.

В карте групп, которую формирует установщик, маркеров нет. Если маркеры нужны, их добавляют только в перечень `roles` группы администраторов ЦС.

4.3 Стандартные роли и матрица доступа

Раздел описывает три стандартные роли и группы каталога, которым они сопоставлены в карте `ldapGroupRolesMap`, и показывает, какие права входят в каждую роль.

Каждая роль соответствует одной группе каталога. Имена групп можно заменить на принятые в вашем каталоге, см. Назначение ролей через группы AD/LDAP (see page 196).

Роль	Группа каталога	Права
Администратор ЦС	`CWCA_Administrator`	Все 17 прав
Аудитор	`CWCA_Auditor`	6 прав на просмотр: CertList-r, EventList-r, TemplList-r, ReqList-r, CRL-r, Config-r
Пользователь	`CWCA_User`	3 права: CertList-r, TemplList-r, CertReq-w

Администратору ЦС должны быть сопоставлены все 17 прав. Если у группы администраторов в карте перечислено 13 прав, без прав резервного копирования и ротации ключей, дополните карту: см. «Добавление новых прав 26.3.0 в карту групп» в конце раздела.

Кроме карты групп, установщик создаёт в OpenID встроенную учётную запись (секция `staticPasswords` файла `config.yaml`). Её права перечислены не в карте групп, а в собственном перечне `customClaims.roles` этой записи. Сверьте этот перечень с 17 правами из раздела «Перечень прав» и при необходимости дополните его (см. «Права встроенной учётной записи» в разделе Назначение ролей через группы AD/LDAP (see page 196)).

Матрица доступа по отдельным правам (имена прав указаны без префикса `rleMiniCA`):

Право	Администратор ЦС	Аудитор	Пользователь
CertList-r	Да	Да	Да
EventList-r	Да	Да	Нет
TemplList-r	Да	Да	Да
ReqList-r	Да	Да	Нет
CRL-r	Да	Да	Нет
Config-r	Да	Да	Нет
CertReq-w	Да	Нет	Да
CertRev-w	Да	Нет	Нет
CertUnRev-w	Да	Нет	Нет

Templ-w	Да	Нет	Нет
CRL-w	Да	Нет	Нет
Config-w	Да	Нет	Нет
ReqApprove-w	Да	Нет	Нет
Backup-r	Да	Нет	Нет
Backup-w	Да	Нет	Нет
BackupRestore-w	Да	Нет	Нет
Rotation-w	Да	Нет	Нет

Отличия ролей:

- **Аудитор** только просматривает сертификаты, шаблоны, запросы, журнал событий, списки отзыва, конфигурацию и статистику ЦС. Не изменяет данные, не выпускает и не отзывает сертификаты.
- **Пользователь** просматривает реестры сертификатов и шаблонов и создаёт запросы на выпуск. Не отзывает сертификаты, не изменяет шаблоны, не видит журнал событий, запросы, списки отзыва и конфигурацию.
- **Администратор ЦС** имеет полный набор прав: дополнительно отзыв и снятие отзыва, шаблоны, списки отзыва и задачи публикации, конфигурация и подключения ЦС, одобрение запросов, резервное копирование и восстановление, ротация ключей ЦС.



Роли с другим набором прав

Если нужна роль с промежуточным набором прав, например оператор, который отзывает сертификаты и изменяет шаблоны, создайте для неё группу каталога и добавьте группу в карту `ldapGroupRolesMap` с нужным перечнем прав (см. Назначение ролей через группы AD/LDAP (see page 196)).

4.3.1 Добавление новых прав 26.3.0 в карту групп

Разделы «Задачи → Резервное копирование» и «Настройки → Ротация ключей» открываются только правами, добавленными в версии 26.3.0. Проверьте, что у группы администраторов ЦС в карте `ldapGroupRolesMap` и у встроенной учётной записи в перечне `customClaims.roles` перечислены все 17 прав. Если прав 13, дополните перечни, перезапустите сервис OpenID и попросите пользователей группы администраторов выйти из Контрольной панели и войти снова.

Порядок действий, команды и пример карты приведены в разделе Назначение ролей через группы AD/LDAP (see page 196).

Проверка результата: войдите в Контрольную панель учётной записью из группы администраторов и откройте «Профиль». В блоке «Права» должны быть четыре новых права, а в меню должны отображаться пункты «Задачи → Резервное копирование» и «Настройки → Ротация ключей».

4.4 Разграничение доступа к шаблонам

Помимо прав, доступ к отдельному шаблону сертификата можно ограничить списками ролей. Так, например, выпуск по шаблону можно разрешить только части пользователей.

У шаблона три независимых разрешения. Каждое задаётся списком ролей; пустой список означает, что доступ не ограничен.

Разрешение	Что ограничивает	Если у пользователя нет ни одной роли из списка
Просмотр	Просмотр шаблона	Шаблон скрыт в реестре, карточка не открывается
Изменение	Изменение параметров шаблона	Изменения не сохраняются, выводится сообщение об ошибке
Выпуск	Использование шаблона при выпуске сертификата	Шаблон скрыт в списке шаблонов при запросе сертификата

Правила:

- В список вносят значения claim `roles` точно так, как они назначены пользователям, например `rleMiniCATemplList-r`.
- Операция доступна, если у пользователя есть хотя бы одна роль из списка.
- Разрешения задаёт в карточке шаблона пользователь с правом `rleMiniCATempl-w`, если разрешение на изменение этого шаблона его не ограничивает.
- Экспорт шаблонов в JSON (право `rleMiniCATemplList-r`) включает только шаблоны, доступные пользователю по разрешению на просмотр.

> **Разрешение на просмотр действует и на автора**
>
> Если у пользователя, который задал разрешение на просмотр, нет ни одной из указанных ролей, после сохранения шаблон исчезнет из его реестра, а карточка станет недоступна.

Порядок настройки разрешений в интерфейсе описан в Руководстве пользователя, раздел «Разграничение прав доступа к шаблонам».

5 Установка

5.1 Общие сведения об установке

Раздел объясняет, как устроена автоматическая установка Clearway CA: как запустить установщик, из каких групп и компонентов состоит установка, в каком порядке их выполнять и что установщик проверяет на целевых серверах. Если установщик использовать нельзя, разверните сервисы вручную (см. раздел «Ручное развертывание сервисов (без инсталлятора)»).

Пошаговые инструкции

Подробные инструкции по каждому экрану мастера установки (перечень полей, значения по умолчанию, порядок действий) приведены не в этом руководстве, а в «Руководстве по установке», которое поставляется вместе с установщиком. Здесь дан обзор состава установки и общего порядка действий.

5.1.1 Установщик

Clearway CA устанавливается веб-установщиком (мастером установки). Это самодостаточное приложение: серверная часть на ASP.NET Core, веб-интерфейс на Angular. Установщик запускается на рабочем месте оператора, а его интерфейс открывается в браузере по адресу `https://localhost:5224/installer`.

На целевые серверы сам установщик не устанавливается. Он подключается к каждому серверу по SSH и разворачивает на нём компоненты. Сервер (Host) и учётная запись SSH (Username) задаются в параметрах каждого компонента, вход выполняется по паролю или по SSH-ключу. Перед установкой компонента установщик проверяет пререквизиты: условия на целевом сервере, без которых компонент не установится (подробнее см. раздел «Пререквизиты»).

Учётной записи SSH нужны права sudo: через sudo установщик устанавливает пакеты, создаёт каталоги и технологическую учётную запись. Технологическая учётная запись (по умолчанию `itc-svc`) владеет каталогом установки, от её имени работают службы Clearway CA. Сценарии установки компонентов установщик тоже выполняет от её имени.

Ход и результаты установок (введённые значения, выпущенные сертификаты, состояние компонентов) установщик сохраняет в своей базе. Поэтому начатый сценарий можно продолжить позже, а результаты одних компонентов становятся входными данными других. Например, запрос на сертификат, сформированный на одном шаге, автоматически подставляется в шаг выпуска сертификата в центре сертификации (ЦС). Выполненные установки перечислены на странице «История установок».

Версия для Linux

Установщик поставляется также в версии для ОС Linux (пакеты deb и rpm), её предоставляют по запросу заказчика. Порядок запуска этой версии описан ниже.

5.1.1.1 Запуск установщика (версия для Linux)

Что понадобится: рабочее место оператора под управлением Linux с браузером, сетевой доступ с него по SSH к целевым серверам и пакет установщика.

Шаг 1. Установите пакет установщика на рабочее место оператора. Команда выполняется с правами sudo; вместо `<версия>` подставьте версию из имени полученного файла.

```
# пакет deb (Astra Linux и другие ОС семейства Debian):
sudo apt install ./clearway-ca-installer_<версия>_amd64.deb
# пакет rpm, РЕД ОС:
sudo dnf install ./clearway-ca-installer-<версия>.x86_64.rpm
# пакет rpm, Альт:
sudo apt-get install ./clearway-ca-installer-<версия>.x86_64.rpm
```

Файлы установщика размещаются в каталоге `/opt/itc-installer`.

Шаг 2. Запустите установщик от имени обычного пользователя, без sudo. Можно также воспользоваться ярлыком «ITC Installer».

```
itc-installer
```

Установщик запускает локальный веб-сервер и открывает в браузере адрес `https://localhost:5224`; страница `/installer` открывается автоматически. На рабочем месте без графической оболочки используйте режим командной строки:

```
itc-installer cli
```

Шаг 3. При первом открытии интерфейса прочитайте лицензионное соглашение, отметьте флажок «Я принимаю условия лицензионного соглашения» и нажмите «Принимаю». После этого открывается выбор компонентов для установки.

База установщика (`installer.db`) и его журналы хранятся в каталоге `~/.local/share/itc-installer` пользователя, который запустил установщик.

Защита установщика

Разрешите подключение к порту 5224 только с рабочего места оператора. Ограничьте доступ к каталогу `~/.local/share/itc-installer`, где хранятся параметры, введённые при установке, и к учётной записи, от которой запускается установщик. Порядок действий см. «Веб-интерфейс установщика» в главе Обеспечение безопасной среды (see page 167).

5.1.2 Группы установки и компоненты

Установка разбита на компоненты. Компонент соответствует одному шагу мастера на одном сервере: создание базы данных, формирование запроса на сертификат, выпуск сертификата, установка или настройка сервиса. Запрос на сертификат (CSR) содержит открытый ключ и сведения о владельце; по нему ЦС выпускает сертификат.

Компоненты объединены в девять групп. Группа описывает законченный сценарий развертывания: в ней заданы состав компонентов и порядок их выполнения. Группу выбирают на старте, после чего мастер ведёт по её шагам. Необязательные шаги можно пропустить.

Группа	Что разворачивает	Шаги по порядку	Необязательные шаги
Корневой ЦС	Самоподписанный корневой ЦС на ядре ЦС (minica) с собственной базой данных. После установки ЦС принимает запросы по HTTPS на порту 9443	База данных ЦС → Установка корневого ЦС	Нет
Выдающий ЦС	Подчинённый (выдающий) ЦС: ключ и CSR, подписание на корневом ЦС, установка сертификата и запуск	База данных ЦС → Установка выдающего ЦС (ключ и CSR) → Выпуск сертификата на корневом ЦС → Настройка выдающего ЦС	Выпуск сертификата, если сертификат выдающего ЦС подписывается внешним корневым ЦС организации, а не корневым ЦС Clearway CA
Аутентификация (OpenID)	Сервис единого входа по протоколу OpenID Connect на основе Dex: встроенная учётная запись и при необходимости вход по учётным записям LDAP/AD	База данных OpenID → Выпуск запроса (CSR для TLS) → Выпуск сертификата → Установка OpenID → Интеграция с LDAP	Интеграция с LDAP

Контрольная панель	Веб-интерфейс управления ЦС (серверная часть и веб-приложение) за Nginx со входом через OpenID; подключение ЦС к панели. Другие ЦС можно подключить позже в интерфейсе: «Настройки → Конфигурация»	Выпуск запроса для TLS → Выпуск сертификата → Установка Контрольной панели → Выпуск JWT → Настройка (подключение ЦС)	Нет
Сервис публикации	Публикация сертификатов и списков отзыва (CRL) на точки распространения, в том числе в LDAP; за Nginx	Выпуск запроса для TLS → Выпуск сертификата → Установка сервиса публикации → Выпуск JWT	Выпуск запроса для TLS и Выпуск сертификата, если на сервере уже есть ключ и TLS-сертификат Nginx (`/opt/itc/itc/certs/nginx-tls.key` и `nginx-tls.crt`)
OCSP	OCSP-ответчик mOCSP: сообщает статус отдельного сертификата по данным базы ЦС; принимает запросы на порту 8888 без Nginx	Установка mOCSP (CSR сертификата подписи) → Выпуск сертификата подписи → Настройка OCSP	Нет
Архиватор	Сервис, который по расписанию переносит истекшие сертификаты и связанные с ними записи из рабочей базы ЦС в архивную	Архивная база данных → Установка архиватора	Нет
SCEP	Адаптер SCEP для автоматического выпуска сертификатов устройствам; за Nginx; аутентификация через OpenID; состояние хранится в SQLite	Выпуск запросов (TLS и SCEP) → Выпуск сертификатов → Установка адаптера SCEP → Выпуск JWT → Настройка SCEP	Нет
MS-WSTEP	Службы CEP (политика регистрации) и CES (выпуск сертификатов) для компьютеров и пользователей Windows в домене Active Directory; за Nginx; проверка подлинности Kerberos	Выпуск запроса для TLS → Выпуск сертификата → Установка MS-WSTEP → Выпуск JWT → Настройка MS-WSTEP	Выпуск запроса для TLS и Выпуск сертификата, если на сервере уже есть ключ и TLS-сертификат Nginx (`/opt/itc/itc/certs/nginx-tls.key` и `nginx-tls.crt`)

Протокольные адаптеры

Установщик разворачивает адаптеры SCEP и MS-WSTEP (группы «SCEP» и «MS-WSTEP»). Для MS-WSTEP нужен домен Active Directory: сервер адаптера вводится в домен на этапе проверки пререквизитов. Адаптеры ACMEv2, CMP, EST и MS-RPC в состав установщика не входят и поставляются отдельно.

Встроенная учётная запись OpenID

Группа «Аутентификация (OpenID)» создаёт встроенную учётную запись с правами администратора Clearway CA. По умолчанию используются логин `admin@company.ru` и отображаемое имя `admin`. Пароль задаётся при установке, в конфигурацию записывается только его хэш (bcrypt). С этой учётной записью в Контрольную панель можно войти и без каталога LDAP.
Шаг «Интеграция с LDAP» добавляет подключение к LDAP/AD, но встроенную учётную запись не удаляет. Как отключить её после проверки входа через каталог, см. «Встроенная учётная запись OpenID» в главе Обеспечение безопасной среды (see page 167).

5.1.2.1 Шаги, общие для нескольких групп

Шаг	Где выполняется	Что делает
База данных	Сервер СУБД	При необходимости устанавливает PostgreSQL, создаёт учётную запись и базу данных, выдаёт права. Для базы ЦС и архивной базы загружает схему. Имена по умолчанию: база ЦС `cwica`, архивная база `cwica_archive` (для обеих учётная запись `cwica`), база OpenID `openid` (учётная запись `cwi`)
Выпуск запроса для TLS	Сервер устанавливаемого сервиса	Создаёт ключ и CSR для TLS-сертификата сервиса
Выпуск сертификата	По умолчанию сервер выдающего ЦС, если группа «Выдающий ЦС» уже установлена, иначе сервер корневого ЦС	Выпускает сертификат по CSR утилитой командной строки ЦС `mclient`. По умолчанию используется шаблон `tls`, для выдающего ЦС `subca`, для OCSP `ocsp`

Выпуск JWT	По умолчанию тот же сервер ЦС, что и для «Выпуска сертификата»	Выпускает JWT (токен, по которому ЦС проверяет права подключающегося сервиса) и ключ к нему. Ключ шифруется, если задана парольная фраза. По умолчанию токен выпускается с ролью `admin` (полные права в ЦС); другую роль выбирают в расширенных настройках шага, см. «Сервисные роли JWT (ролевая модель)»
Настройка	Сервер устанавливаемого сервиса	Завершает установку: устанавливает выпущенные сертификаты, подключает сервис к ЦС или запускает службу

5.1.2.2 Точка распространения CRL и сертификатов (CDP)

Компонент «Точка распространения CRL и сертификатов (web-сервер)» не входит в группы: в мастере он находится в разделе «Остальные компоненты». Он настраивает сайт Nginx на порту 80, который раздаёт по HTTP файлы из каталога `/opt/itc/web` (подкаталоги `cdp/` и `aia/`). Имя сервера CDP указывается при установке ЦС (параметр «FQDN сервера точек распространения») и попадает в расширения CDP и AIA выпускаемых сертификатов. Списки отзыва в каталог `cdp/` записывает сервис публикации по заданиям публикации.

5.1.2.3 Как работают установленные службы

Все службы Clearway CA работают как пользовательские юниты systemd технологической учётной записи (`systemctl --user`); файлы юнитов лежат в `/home/<технологическая учётная запись>/.config/systemd/user/` (по умолчанию `/home/itc-svc/.config/systemd/user/`). Чтобы службы работали без входа этой учётной записи в систему, для неё включается режим linger. Кроме того, в её `~/.bashrc` прописываются переменные XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS: без них не работает `systemctl --user`. Nginx и PostgreSQL работают как системные службы ОС.

Контрольная панель, сервис публикации и адаптеры SCEP и MS-WSTEP публикуются через Nginx: он принимает запросы по HTTPS на порту 443 и передаёт их приложению на том же сервере. Собственные порты .NET-сервисов: Контрольная панель (8407), сервис публикации (9407), архиватор (9507), SCEP (8440), службы CEP и CES адаптера MS-WSTEP (8441 и 8442).

Сервис OpenID сам принимает HTTPS-подключения на порту 443, без Nginx. Ответчик mOCSP принимает HTTP-запросы на порту 8888 (путь `/ocsp`), тоже без Nginx. Полный перечень портов приведён в главе Требования к среде (see page 38) и в разделе «Приложение Б. Сетевые порты».

Установщик поддерживает PostgreSQL версий 11–18. По умолчанию выбирается наибольшая версия, доступная в репозиториях ОС сервера СУБД; если определить её не удалось, выбирается 16.

База данных на отдельном сервере

Шаг «База данных» создаёт учётную запись и базу через локальное подключение к PostgreSQL и не меняет параметры удалённого доступа: `listen_addresses` в `postgresql.conf` и правила `pg_hba.conf`. Если сервис и его база данных находятся на разных серверах, до установки сервиса разрешите в PostgreSQL подключение с сервера сервиса. Иначе не пройдёт проверка «Проверка подключения к базе данных». Порядок настройки описан в шаге 4 раздела Развертывание базы данных ЦС (see page 71).

5.1.3 Порядок установки

Перед первой установкой запустите установщик (см. «Руководство по установке»; для версии для Linux см. «Запуск установщика (версия для Linux)») и примите лицензионное соглашение. Затем устанавливайте группы с учётом их зависимостей, а шаги внутри группы выполняйте по порядку.

5.1.3.1 Этапы установки компонента

Каждый компонент мастер проводит через одни и те же этапы.

№	Этап	Что делает администратор	Что делает мастер
1	Выбор компонента	Выбирает группу и её очередной шаг или компонент вне групп. Необязательный шаг можно пропустить	Показывает шаги группы и отмечает выполненные
2	Сервер и параметры	На первом экране указывает имя сервера, учётную запись SSH и пароль или SSH-ключ. На следующих экранах заполняет параметры компонента: реквизиты БД, сертификаты, параметры LDAP и т. п. Технологическая учётная запись и каталог установки обычно находятся в расширенных настройках	Подключается к серверу по SSH. Подставляет значения, полученные на предыдущих шагах и в ранее установленных группах
3	Проверка пререквизитов	Изучает результаты проверок. Если есть непройденные, нажимает «Выполнить настройку пререквизитов» и при запросе вводит пароль администратора для sudo	Проверяет готовность сервера. По кнопке исправляет то, что можно исправить автоматически: создаёт учётную запись и каталог, выставляет владельца, включает linger, устанавливает недостающие пакеты

4	Установка	Нажимает «Выполнить установку»	Загружает на сервер исполняемые файлы и конфигурации, выполняет сценарии установки от имени технологической учётной записи, запускает пользовательские юниты systemd
5	Результаты	Скачивает нужные выходные файлы	Сохраняет результаты в своей базе и подставляет их в следующие шаги и группы: сертификаты (ЦС, TLS, цепочка), CSR, JWT с ключом. Для MS-WSTEP формирует LDF-файлы и сценарий `add_enrollment_policy.ps1`: их скачивают, а сценарий запускают от имени администратора на компьютере Windows с доступом к контроллеру домена

Проверка результата. Шаг выполнен, если все сценарии этапа «Установка» завершились без ошибок. Чтобы убедиться, что служба работает, войдите на сервер компонента и проверьте её юнит от имени технологической учётной записи. Для ЦС, кроме того, проверьте ответ службы командой `mclient ping`. Команды выполняются на сервере ЦС от учётной записи SSH с правами sudo; первая переключает сеанс на технологическую учётную запись (по умолчанию `itc-svc`):

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

Успех: служба в состоянии `active (running)`, `mclient ping` выводит версию ЦС и SKI его ключа. Имена юнитов других компонентов: `mocsp.service` (OCSP), `openid.service` (OpenID), `controlPanel.service` (Контрольная панель), `publication.service` (сервис публикации), `archiver.service` (архиватор), `scepAdapter.service` (SCEP), `wstepCep.service` и `wstepCesAdapter.service` (MS-WSTEP).

5.1.3.2 Зависимости между группами

Многие параметры мастер берёт из групп, установленных раньше. Устанавливайте группу, когда готово всё, что указано во втором столбце.

Группа	Что должно быть установлено раньше	Что мастер берёт из ранее установленных групп
Корневой ЦС	Нет	Нет
Выдающий ЦС	Корневой ЦС, если сертификат выдающего ЦС подписывается на нём	Сертификат корневого ЦС; шаг «Выпуск сертификата» выполняется на сервере корневого ЦС
Аутентификация (OpenID)	ЦС (корневой или выдающий)	TLS-сертификат OpenID выпускается на ЦС
Контрольная панель	ЦС, Аутентификация (OpenID)	Адрес OpenID и идентификатор клиента; имя сервера и TLS-сертификат ЦС для подключения; JWT выпускается на ЦС
Сервис публикации	ЦС, Аутентификация (OpenID)	Адрес OpenID; TLS-сертификат и JWT выпускаются на ЦС
OCSP	ЦС	Параметры базы ЦС и сертификат ЦС; сертификат подписи выпускается на ЦС
Архиватор	ЦС	Параметры рабочей базы ЦС
SCEP	ЦС, Аутентификация (OpenID)	Адрес OpenID, цепочка сертификатов ЦС, имя сервера и TLS-сертификат ЦС
MS-WSTEP	ЦС; домен Active Directory	Имя сервера и TLS-сертификат ЦС
CDP (вне групп)	Нет	Нет

Если отдельный выдающий ЦС не нужен и сертификаты выпускает сам корневой ЦС, группу «Выдающий ЦС» не устанавливайте. Контрольная панель по умолчанию обращается к сервису публикации по адресу `http://localhost:9407` (параметр `Services.PublicationCrlUrl`), поэтому при настройках по умолчанию сервис публикации должен находиться на сервере Контрольной панели. Если он установлен на другой сервер, измените этот адрес в конфигурации Контрольной панели.

Все зависимости соблюдены, например, при такой последовательности: Корневой ЦС → Выдающий ЦС → Аутентификация (OpenID) → Контрольная панель → Сервис публикации → OCSP → Архиватор → SCEP и MS-WSTEP. Компонент CDP можно установить в любой момент; имя его сервера указывают заранее, при установке ЦС (см. «Точка распространения CRL и сертификатов (CDP)»).

Размещение на серверах

Установщик не поддерживает установку на один сервер следующих пар: корневой и выдающий ЦС; Контрольная панель и Аутентификация (OpenID); SCEP и Аутентификация (OpenID). Для групп каждой пары указывайте разные серверы на первом экране компонента. Сервис OpenID сам принимает подключения на порту 443, а на серверах Контрольной панели и SCEP этот порт занимает Nginx.

5.1.4 Пререквизиты

Раздел перечисляет проверки, которые мастер выполняет на целевом сервере перед установкой компонента. Для большинства из них предусмотрено автоматическое исправление: его запускает кнопка «Выполнить настройку пререквизитов».

Пререквизит	Какие шаги проверяют	Проверка	Автоматическое исправление
Технологическая учётная запись	Выпуск запроса (запросов); установка ЦС, OCSP, OpenID, Контрольной панели, сервиса публикации, архиватора, SCEP, MS-WSTEP и CDP; Настройка выдающего ЦС и OCSP	Учётная запись существует (по умолчанию `itc-svc`)	Создание учётной записи
Каталог установки	Выпуск запроса (запросов); установка ЦС, OCSP, OpenID, Контрольной панели, сервиса публикации, архиватора, SCEP, MS-WSTEP и CDP; Настройка выдающего ЦС и OCSP	Каталог существует. По умолчанию `/opt/itc`; для ЦС `/opt/itc/minica`, для OCSP `/opt/itc/ocsp`, для CDP `/opt/itc/web`	Создание каталога
Владелец каталога	Выпуск запроса (запросов); установка ЦС, OCSP, OpenID, Контрольной панели, сервиса публикации, архиватора, SCEP, MS-WSTEP и CDP; Настройка выдающего ЦС и OCSP	Владелец и группа каталога соответствуют технологической учётной записи	Смена владельца (`chown`)

Режим linger	Установка ЦС, OCSP, OpenID, Контрольной панели, сервиса публикации, архиватора, SCEP, MS-WSTEP	Для технологической учётной записи включён linger (`Linger=yes`)	`loginctl enable-linger`
Nginx	Установка Контрольной панели, сервиса публикации, SCEP, MS-WSTEP, CDP	Служба `nginx` активна	Установка пакета nginx, включение и запуск службы; на РЕД ОС также разрешение сетевых подключений для nginx в SELinux (`httpd_can_network_connect`)
OpenSSL	Установка корневого и выдающего ЦС, MS-WSTEP	Команда `openssl` доступна	Установка пакета openssl
jq	Настройка Контрольной панели, SCEP, MS-WSTEP	Команда `jq` доступна	Установка пакета jq
Клиент LDAP	Интеграция с LDAP, установка MS-WSTEP	Команда `ldapsearch` доступна	Установка пакета openldap-clients или ldap-utils
PostgreSQL	База данных (ЦС, архивная, OpenID)	PostgreSQL принимает подключения (`pg_isready`)	Установка PostgreSQL выбранной версии, инициализация и запуск службы
Учётная запись и база в PostgreSQL	База данных	Учётная запись и база существуют	Создание учётной записи и базы; эта учётная запись становится владельцем базы
Пустая база	База данных	В схеме `public` нет таблиц	Удаление и пересоздание схемы `public` : **все данные в базе уничтожаются**
Права в PostgreSQL	База данных	Права CONNECT и CREATE на базу, USAGE и CREATE на схему `public`	Выдача прав, в том числе прав по умолчанию на таблицы и последовательности схемы `public`

Подключение к базе данных	Установка ЦС, OCSP, OpenID, архиватора (к рабочей и к архивной базе)	Подключение с заданными реквизитами	Нет
Работающий сервис	Выпуск сертификата и Выпуск JWT (ЦС); Настройка Контрольной панели, SCEP, MS-WSTEP (служба своей группы); Интеграция с LDAP (OpenID)	ЦС отвечает на `mclient ping`; нужная служба активна	Нет
Каталог LDAP	Интеграция с LDAP, установка MS-WSTEP	Сервер LDAP доступен; для «Интеграции с LDAP» также успешный вход (bind) с заданными DN и паролем	Нет
Ввод в домен Active Directory	Установка MS-WSTEP	Сервер введён в домен: домен есть в выводе `realm list`, файл `/etc/krb5.keytab` не пустой	Ввод в домен: установка realmd и sssd, присоединение к домену с учётными данными администратора домена
Обнаружение домена Active Directory	Установка MS-WSTEP	Домен обнаруживается командой `realm discover`	Нет

Повторная установка на работающий ЦС

Автоматическое исправление проверки «Проверка отсутствия таблиц в БД» удаляет и заново создаёт схему `public` (`DROP SCHEMA public CASCADE`): все данные в базе уничтожаются. Не нажимайте «Выполнить настройку пререквизитов» на шаге «База данных», если указанная база уже содержит данные ЦС или OpenID.

Повторная установка корневого или выдающего ЦС создаёт новый ключ ЦС (`ca.key`) поверх существующего. Перед повторным запуском этих шагов сделайте резервную копию (см. «Резервное копирование»).

5.1.4.1 Типовые проблемы

Что видно	Причина	Что сделать

Не проходит «Проверка подключения к базе данных»	Неверные реквизиты БД или PostgreSQL не принимает подключения с этого сервера	Проверьте адрес и порт сервера СУБД, имя базы, учётную запись и пароль; при необходимости разрешите удалённое подключение в PostgreSQL (шаг 4 раздела Развертывание базы данных ЦС (see page 71))
На шагах «Выпуск сертификата» и «Выпуск JWT» не проходит «Проверка установки ЦС»	ЦС не установлен или его служба не отвечает на `mclient ping`	Установите группу «Корневой ЦС» или «Выдающий ЦС» либо проверьте службу `minica.service`
Не проходит «Проверка отсутствия таблиц в БД»	В базе уже есть данные	Укажите новую пустую базу; не запускайте автоматическое исправление для базы с данными
На шаге «Настройка» не проходит проверка установки службы (Контрольной панели, SCEP или CES)	Служба, установленная на предыдущем шаге группы, не запущена	Проверьте службу командой `systemctl --user status <юнит>` от имени технологической учётной записи
На шаге «Интеграция с LDAP» не проходит «Проверка подключения к серверу LDAP»	Неверные адрес сервера, DN или пароль учётной записи для подключения к каталогу	Проверьте адрес сервера, DN и пароль
При установке MS-WSTEP не проходят проверки домена или LDAP-сервера	Сервер не находит домен через DNS или контроллер домена недоступен	Проверьте DNS-имя домена и сетевой доступ к контроллеру домена

5.1.5 После установки

После установки приведите среду в безопасное состояние (см. главу Обеспечение безопасной среды (see page 167)). В первую очередь:

- ограничьте права на конфигурационные файлы с паролями и ключами (см. Ограничение прав доступа к файлам (see page 178));
- закройте для внешних подключений порты .NET-сервисов 8407, 9407, 9507 и 8440, при установленном MS-WSTEP также 8441 и 8442, а при работающем сервисе резервного копирования его порт (в примерах 8409); если сервис публикации или сервис резервного копирования установлен отдельно, оставьте его порт открытым только для сервера Контрольной панели (см. «Ограничение сетевого доступа» в главе Обеспечение безопасной среды (see page 167));
- проверьте права встроенной учётной записи OpenID и групп каталога, включая права резервного копирования и ротации ключей (см. Описание ролевой модели (see page 47) и Назначение ролей через группы AD/LDAP (see page 196));

- храните пароль встроенной учётной записи OpenID в тайне, а после настройки входа через LDAP отключите её (см. «Встроенная учётная запись OpenID» в главе Обеспечение безопасной среды (see page 167));
- для сервисов, которым не нужны полные права в ЦС, используйте JWT с ролью меньших прав (см. «Сервисные роли JWT (ролевая модель)»).

5.2 Ручное развертывание сервисов (без инсталлятора)

В этом разделе описано, как развернуть и настроить сервисы Clearway CA вручную, без веб-установщика. Такой способ применяют, когда использовать установщик невозможно или нежелательно.

Все прикладные сервисы работают как пользовательские юниты systemd технологической учётной записи (по умолчанию itc-svc). Параметры конфигураций подробно описаны в разделах Настройка ядра ЦС (see page 126) и Настройка вспомогательных сервисов (see page 139).

Каталоги, файлы и службы при ручном развертывании те же, что при автоматической установке, и порядок действий тот же:

1. Общая подготовка каждого сервера, на котором будут работать сервисы Clearway CA.
2. Развертывание базы данных ЦС на сервере СУБД.
3. Развертывание ядра ЦС: сначала корневого, затем, если он нужен, выдающего.
4. Развертывание вспомогательных сервисов (описано в следующих разделах).

> **Соглашения в командах**
>
> Команды используют значения по умолчанию: учётная запись itc-svc, каталог /opt/itc (для ЦС /opt/itc/minica). Если вы выбрали другие значения, подставьте свои. Заполнители в угловых скобках, например `<ПАРОЛЬ>` или `<host>`, замените своими значениями. Перед началом ознакомьтесь с разделами Требования к среде (see page 38) и Обеспечение безопасной среды (see page 167).

5.2.1 Общая подготовка сервера

Подготовка создаёт на сервере учётную запись и каталог для сервисов Clearway CA и настраивает запуск пользовательских служб. Выполните её на каждом сервере, где будут работать сервисы Clearway CA: корневого и выдающего ЦС, OpenID, вспомогательных сервисов. Серверу, на котором работает только СУБД, эта подготовка не нужна.

Используемые понятия:

- **Технологическая учётная запись** (по умолчанию itc-svc) владеет каталогом установки; службы Clearway CA работают от её имени, а не от root.
- **Пользовательский юнит systemd**: служба, которой учётная запись управляет сама командой `systemctl --user`, без прав администратора.

- **Режим linger** разрешает пользовательским службам работать без открытого сеанса учётной записи, в том числе после перезагрузки сервера.

Шаги 1–3, 5 и 6 выполняются под учётной записью администратора сервера (через sudo), а шаг 4 в сеансе itc-svc.

Шаг 1. Создайте технологическую учётную запись.

```
# Astra Linux и другие ОС семейства Debian:
sudo adduser --disabled-password --gecos "" itc-svc
# Альт (ALT Linux), РЕД ОС:
sudo adduser itc-svc
# проверка: команда выводит числовой идентификатор учётной записи
id -u itc-svc
```

Шаг 2. Создайте каталог установки и передайте его технологической учётной записи. Для ЦС это /opt/itc/minica; на серверах OpenID и вспомогательных сервисов укажите /opt/itc.

```
sudo mkdir -p /opt/itc/minica
sudo chown itc-svc:itc-svc /opt/itc/minica
# проверка, ожидается itc-svc:itc-svc
stat -c "%U:%G" /opt/itc/minica
```

Шаг 3. Включите режим linger для учётной записи. Без него пользовательские службы не запускаются, пока учётная запись не вошла в систему.

```
sudo loginctl enable-linger itc-svc
# проверка, ожидается Linger=yes
loginctl show-user itc-svc | grep --color=never Linger
```

Шаг 4. Настройте окружение для `systemctl --user`. Войдите в сеанс учётной записи командой `sudo su - itc-svc` и выполните команды ниже. Они записывают в ~/.bashrc переменные XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS (адрес шины D-Bus): без них `systemctl --user` не может подключиться к диспетчеру служб учётной записи. Команды можно выполнять повторно: прежние строки удаляются перед записью новых.

```
sed -i "/export XDG_RUNTIME_DIR=/d" ~/.bashrc
echo "export XDG_RUNTIME_DIR=/run/user/$UID" >> ~/.bashrc
export XDG_RUNTIME_DIR=/run/user/$UID
sed -i "/export DBUS_SESSION_BUS_ADDRESS=/d" ~/.bashrc
echo "export DBUS_SESSION_BUS_ADDRESS=unix:path=${XDG_RUNTIME_DIR}/bus" >> ~/.bashrc
export DBUS_SESSION_BUS_ADDRESS="unix:path=${XDG_RUNTIME_DIR}/bus"
# проверка: выводится список служб, без ошибки «Failed to connect to bus»
systemctl --user list-units --type=service --no-pager
```

Завершите сеанс itc-svc командой `exit`.

Шаг 5. Установите OpenSSL. На серверах ЦС он нужен ядру ЦС для работы с ключами, запросами и сертификатами, а на остальных серверах нужен для формирования запросов на TLS-сертификаты.

```
sudo apt-get install -y openssl    # Astra Linux, Альт
sudo dnf install -y openssl        # РЕД ОС
# проверка: команда выводит путь к openssl
command -v openssl
```

Шаг 6. Установите Nginx, если на сервере будут работать компоненты, которые публикуются по HTTP/HTTPS: Контрольная панель, сервис публикации, адаптеры SCEP и MS-WSTEP, точка распространения CRL и сертификатов (CDP/AIA). Ядру ЦС Nginx не нужен. На РЕД ОС дополнительно разрешите Nginx сетевые подключения в политике SELinux, иначе он не сможет передавать запросы сервисам.

```
sudo apt-get install -y nginx      # Astra Linux, Альт
sudo dnf install -y nginx          # РЕД ОС
sudo setsebool -P httpd_can_network_connect 1   # только РЕД ОС
sudo systemctl enable --now nginx
# проверка, ожидается active
systemctl is-active nginx
```

5.2.2 Подготовка СУБД PostgreSQL

Поддерживается PostgreSQL 13 или новее. Миграции ЦС создают расширение `pg_trgm` (`CREATE EXTENSION IF NOT EXISTS pg_trgm;`): начиная с версии 13 расширение доверенное, его устанавливает владелец базы без прав суперпользователя.

> **PostgreSQL 11 и 12**
>
> Расширение `pg_trgm` на этих версиях устанавливает суперпользователь вручную (см. «Подготовка внешней базы»).

Инсталлятор определяет установленную версию по пакетной базе или устанавливает доступную из репозитория ОС. Поддерживаются версии от 11 до 18.

Компоненты, работающие с базой

Компонент	База данных

`minica`	Основная БД ЦС; объекты в схеме `public` создаёт сам сервис при старте (миграции)
`mocsp`	БД OCSP-сервиса; подключение настраивается так же, как у `minica`
`scep`	БД SCEP-сервиса
`archiver`	Копирует данные из исходной БД ЦС в архивную; обе строки подключения в конфиге `archiver`

Для заданий резервного копирования на сервере Контрольной панели должны быть утилиты `pg_dump` и `pg_restore` (пакет `postgresql-client`).

Установка PostgreSQL

Способ установки зависит от размещения базы: на сервере компонента или на отдельном сервере.

Локальная база

Если база размещается на сервере компонента, инсталлятор выполняет подготовку сам:

- устанавливает пакеты PostgreSQL и инициализирует кластер (пароли `scram-sha-256`, локальный доступ `peer`);
- запускает сервис и включает автозапуск;
- создаёт роль и базу, выдаёт права (см. «Подготовка внешней базы»).

Установка на отдельном сервере

Если база вынесена на отдельный сервер, инсталлятор при установке компонента сам настраивает прослушивание сети и правила `pg_hba.conf` для сервера компонента.

Базу устанавливает администратор. Установите пакет `contrib`: расширение `pg_trgm` входит в него. Возьмите версию, доступную в репозитории ОС, 13 или новее.

Debian, Astra Linux

Кластер создаётся автоматически при установке пакета:

```
sudo apt-get install -y postgresql-15 postgresql-contrib-15
sudo systemctl enable --now postgresql
```

РЕД ОС

```
sudo dnf install -y postgresql16-server postgresql16-contrib
sudo /usr/pgsql-16/bin/postgresql-16-setup initdb
sudo systemctl enable --now postgresql-16
```

ALT Linux

```
sudo apt-get install -y postgresql16 postgresql16-contrib
sudo -u postgres /usr/bin/initdb -D /var/lib/pgsql/data -A scram-sha-256 --auth-
local=peer
sudo systemctl enable --now postgresql
```

Проверьте установку:

```
sudo -u postgres psql -c "select version();"
```

Затем подготовьте роли, базы и доступ по сети (см. «Подготовка внешней базы»).

Подготовка внешней базы

Для каждого компонента создайте свою роль и базу: `minica`, `mocsp`, `scep`; для архиватора дополнительно архивную базу. Выполняйте команды на сервере PostgreSQL от имени `postgres`. Примеры даны для компонента `minica`, для остальных подставьте его имя.

Шаг 1. Создайте роль и базу:

```
CREATE ROLE minica LOGIN PASSWORD '<ПАРОЛЬ>';
CREATE DATABASE minica OWNER minica;
```

Шаг 2. Выдайте права, необходимые для миграций:

```
GRANT CONNECT, CREATE ON DATABASE minica TO minica;
\c minica
GRANT USAGE, CREATE ON SCHEMA public TO minica;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON TABLES TO minica;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON SEQUENCES TO minica;
```

Шаг 3. Проверьте владельца объектов. При повторной установке на уже используемой базе таблицы могут принадлежать суперпользователю, тогда сервис падает с ошибкой SQLSTATE 42501 (permission denied). Владельцем всех таблиц и последовательностей в схеме `public` должен быть пользователь ЦС. Запрос должен вернуть 0:

```
SELECT count(*) FROM pg_class c
JOIN pg_namespace n ON n.oid = c.relnamespace
WHERE n.nspname = 'public'
  AND c.relkind IN ('r', 'S')
  AND c.relowner <> (SELECT oid FROM pg_roles WHERE rolname = 'minica');
```

Шаг 4. Создайте расширение `pg_trgm`. На PostgreSQL 13 и новее ничего делать не нужно: миграции создают расширение сами. На более старых версиях выполните `CREATE EXTENSION pg_trgm;` в базе компонента от имени суперпользователя.

Шаг 5. Настройте сеть, если база размещена на отдельном сервере:

- `postgresql.conf`: `listen_addresses = '*'` или конкретный адрес;
- `pg_hba.conf`: правило `host all all <IP-адрес сервера компонента>/32 md5`;
- откройте порт 5432 на межсетевом экране для серверов компонентов.

> **Метод md5 и SCRAM**
>
> Метод `md5` принимает и SCRAM-верификаторы (начиная с PostgreSQL 10), поэтому пароль роли может храниться в `scram-sha-256`.

После правки конфигурации перезапустите PostgreSQL.

TLS между компонентами ЦС и базой

TLS к PostgreSQL поддерживают `minica` и `mocsp`. Параметры задаются в блоке `database.tls`; он имеет приоритет над TLS-параметрами URL подключения:

```
database:
  url: postgres://minica:<ПАРОЛЬ>@db.example.local:5432/minica
  tls:
    enable: true                  # только TLS, без перехода на открытый канал
    verify: true                  # проверять цепочку сертификатов и имя сервера
    # ca-cert: conf/db-ca.crt     # пусто: системные корневые сертификаты
    # server-name: db.example.local
    # cert: conf/db-client.crt    # клиентский сертификат для mTLS
    # key: conf/db-client.key
```

На стороне PostgreSQL:

- `ssl = on`, настроены `ssl_cert_file` и `ssl_key_file`;
- сертификат сервера должен проверяться цепочкой, которой доверяет компонент (свой `ca-cert` или системные корневые), и содержать имя из `server-name` или из URL подключения;

- чтобы разрешить только TLS-подключения, используйте правила `hostssl` в `pg_hba.conf`.

> **TLS в конфигурации инсталлятора**
>
> В конфигурации, создаваемой инсталлятором, TLS к базе выключен (`enable: false`). Включайте его после настройки серверной части.

Archiver подключается строкой формата Npgsql; TLS включается параметрами строки подключения, например `sslmode=Require` или `VerifyFull`.

Проверка

Подключение с сервера компонента:

```
psql "postgres://minica:<ПАРОЛЬ>@db.example.local:5432/minica" -c "select version();"
```

От имени пользователя ЦС без ошибок выполняется `CREATE EXTENSION IF NOT EXISTS pg_trgm;` (на PostgreSQL 13 и новее).

После установки инсталлятором стартует сервис ЦС. Ошибки SQLSTATE 42501 в журнале означают проблему прав или владельца объектов (шаг 3 раздела «Подготовка внешней базы»).

5.2.3 Развертывание базы данных ЦС

База данных ЦС хранит запросы на сертификаты, выпущенные сертификаты, списки отзыва (CRL), шаблоны, выданные JWT с открытыми ключами их владельцев и журнал событий ЦС. Раздел описывает установку PostgreSQL, создание БД и загрузку её схемы. Все команды выполняются на сервере СУБД под учётной записью администратора сервера, кроме проверки из шага 4.

К БД ЦС подключаются ядро ЦС, OCSP-ответчик mOCSP, архиватор, а также сервис резервного копирования, который выполняет задания резервного копирования.

Параметр	Назначение	Значение по умолчанию
Версия PostgreSQL	Версия СУБД из поддерживаемых 11–18	Наибольшая доступная в репозитории ОС; в примерах используется 16
Учётная запись в БД	Владелец БД ЦС; от её имени ядро ЦС подключается к БД	cwica

Имя БД	База данных ЦС	cwica; для архивной БД cwica_archive
Порт	Сетевой порт PostgreSQL	5432

Архивная БД для архиватора создаётся так же, но с именем cwica_archive (см. «Развертывание Архиватора»).

Шаг 1. Установите PostgreSQL, если он ещё не установлен. Наличие нужной версии в репозитории проверяется командой `apt-cache show postgresql-16` (Astra Linux), `apt-cache show postgresql16-server` (Альт) или `dnf list --available postgresql16-server` (РЕД ОС). В примерах используется версия 16; для другой версии замените номер.

В Astra Linux и других ОС семейства Debian кластер БД создаётся при установке пакета:

```
sudo apt-get update
sudo apt-get install -y postgresql-16 postgresql-contrib-16
sudo systemctl enable --now postgresql
```

В ОС Альт кластер создаётся командой initdb. Параметр `-A scram-sha-256` включает проверку пароля для подключений по сети, `--auth-local=peer` оставляет вход через локальный сокет по имени системного пользователя (так работает `sudo -u postgres psql`).

```
sudo apt-get update
sudo apt-get install -y postgresql16 postgresql16-server postgresql16-contrib
sudo -u postgres /usr/bin/initdb -D /var/lib/pgsql/data -A scram-sha-256 --auth-
local=peer
sudo systemctl enable --now postgresql
```

В РЕД ОС кластер создаётся сценарием postgresql-16-setup, а каталог утилит PostgreSQL добавляется в PATH:

```
sudo dnf install -y postgresql16 postgresql16-server postgresql16-contrib
sudo /usr/pgsql-16/bin/postgresql-16-setup initdb
sudo systemctl enable --now postgresql-16
echo 'export PATH=/usr/pgsql-16/bin:$PATH' >> ~/.bashrc
export PATH=/usr/pgsql-16/bin:$PATH
```

Проверьте, что СУБД принимает подключения. Ожидается `active`.

```
pg_isready -q && echo active || echo inactive
```

> ⚠ **Пароль в командной строке**
>
> Команды, в которых пароль указан в командной строке (`CREATE USER … PASSWORD` , переменная PGPASSWORD), сохраняют его в истории оболочки. После развертывания удалите эти строки из истории (`history -d <номер>`) или вместо PGPASSWORD используйте файл ~/.pgpass с правами 0600.

Шаг 2. Создайте учётную запись-владельца и БД и выдайте права. Команды выполняются от имени системного пользователя postgres и создают объекты в СУБД. Замените `<ПАРОЛЬ>` паролем учётной записи cwica: этот же пароль указывается потом в конфигурации ядра ЦС. Переход в /tmp нужен, чтобы psql не выводил предупреждение о недоступном текущем каталоге.

```
cd /tmp
sudo -u postgres psql -c "CREATE USER \"cwica\" WITH PASSWORD '<ПАРОЛЬ>';"
sudo -u postgres psql -c "CREATE DATABASE \"cwica\" OWNER \"cwica\";"
sudo -u postgres psql -c "GRANT CONNECT, CREATE ON DATABASE \"cwica\" TO \"cwica\";"
sudo -u postgres psql -d cwica -c "GRANT USAGE, CREATE ON SCHEMA public TO \"cwica\";"
sudo -u postgres psql -d cwica -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL
ON TABLES TO \"cwica\";"
sudo -u postgres psql -d cwica -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL
ON SEQUENCES TO \"cwica\";"
```

Проверьте права. Команда подключается к БД по паролю и должна вывести `t` .

```
PGPASSWORD='<ПАРОЛЬ>' psql -U cwica -h localhost -d cwica -tAc "SELECT
has_database_privilege('cwica', 'cwica', 'CONNECT') AND has_database_privilege('cwica',
'cwica', 'CREATE') AND has_schema_privilege('cwica', 'public', 'USAGE') AND
has_schema_privilege('cwica', 'public', 'CREATE')"
```

Шаг 3. Загрузите схему БД. Скопируйте файл схемы minica.up.sql из дистрибутива на сервер СУБД, например в /tmp. Схему загружают в пустую БД и от имени владельца cwica: тогда таблицы принадлежат ему, а не postgres.

Сначала убедитесь, что БД пуста: ожидается `0` . Если таблицы уже есть, выясните их происхождение, прежде чем продолжать, и не удаляйте данные действующего ЦС.

```
PGPASSWORD='<ПАРОЛЬ>' psql -U cwica -h localhost -d cwica -tAc "SELECT COUNT(*) FROM
information_schema.tables WHERE table_schema='public'"
```

Загрузите схему и проверьте результат. Вторая команда должна вывести таблицы cert, crl, csr, evt, jwt и templ.

```
PGPASSWORD='<ПАРОЛЬ>' psql -U cwica -h localhost -d cwica -f /tmp/minica.up.sql
PGPASSWORD='<ПАРОЛЬ>' psql -U cwica -h localhost -d cwica -tAc "SELECT table_name FROM
information_schema.tables WHERE table_schema='public' ORDER BY 1"
```

Шаг 4. Откройте доступ к БД по сети, если к ней подключаются с других серверов: ядро ЦС на отдельном сервере, mOCSP, архиватор, Контрольная панель. PostgreSQL принимает сетевые подключения только на адресах из параметра listen_addresses (файл postgresql.conf) и только от клиентов, разрешённых в файле pg_hba.conf. Узнайте расположение этих файлов:

```
sudo -u postgres psql -tAc "SHOW config_file"
sudo -u postgres psql -tAc "SHOW hba_file"
```

В postgresql.conf добавьте в listen_addresses адрес сервера СУБД. В pg_hba.conf добавьте строку для каждого сервера, которому нужен доступ к БД ЦС. Метод проверки пароля укажите тот же, что в имеющейся строке для 127.0.0.1 (как правило, scram-sha-256).

```
# postgresql.conf
listen_addresses = 'localhost,<адрес сервера БД>'

# pg_hba.conf
host    cwica    cwica    <адрес сервера ЦС>/32    scram-sha-256
```

Перезапустите PostgreSQL, чтобы применить изменения. Перезапуск разрывает текущие подключения к БД. Затем разрешите в межсетевом экране сервера СУБД входящие подключения на порт 5432/tcp только с этих серверов.

```
sudo systemctl restart postgresql        # Astra Linux, Альт
sudo systemctl restart postgresql-16     # РЕД ОС
```

Проверьте доступ с сервера ЦС. Для проверки на нём нужен клиент PostgreSQL (psql): общая подготовка сервера его не устанавливает, поэтому при необходимости установите пакет клиента PostgreSQL из репозитория ОС. Ожидается `1`.

```
PGPASSWORD='<ПАРОЛЬ>' psql -U cwica -h <сервер БД> -p 5432 -d cwica -tAc "SELECT 1"
```

5.2.4 Развертывание ядра ЦС (minica)

Ядро ЦС работает как служба minica.service: она выпускает и отзывает сертификаты, формирует CRL и хранит свое состояние в БД ЦС. Вместе с ней на сервер устанавливаются консольный клиент mclient для управления ЦС и утилита mjwt для выпуска JWT. В результате на сервере

работает ЦС с доступом по HTTPS на порту 9443, созданы базовые шаблоны и мастер-JWT для mclient.

Разворачивайте ядро после разделов Общая подготовка сервера (see page 69) и Развертывание базы данных ЦС (see page 71). Сначала полностью разверните корневой ЦС: выдающий ЦС получает свой сертификат от корневого по шаблону subca. Все команды выполняются на сервере ЦС в сеансе технологической учётной записи (`sudo su - itc-svc`), если не указано иное. Каталог установки: /opt/itc/minica.

5.2.4.1 Подготовка файлов и конфигурации

Файлы ядра входят в дистрибутив Clearway CA. В составе веб-установщика они находятся в каталоге components/minica, а файл схемы БД minica.up.sql лежит в каталоге components/database. Разместите файлы на сервере ЦС так:

Файл дистрибутива	Размещение на сервере	Назначение
minica	/opt/itc/minica/minica	Исполняемый файл службы ЦС
mclient	/opt/itc/minica/mclient	Консольный клиент управления ЦС
mjwt	/opt/itc/minica/bin/mjwt	Утилита выпуска JWT
minica.yml	/opt/itc/minica/conf/minica.yml	Конфигурация ЦС
mclient.yml	/opt/itc/minica/conf/mclient.yml	Конфигурация mclient
openssl.cnf	/opt/itc/minica/conf/openssl.cnf	Базовая конфигурация OpenSSL
minica.env	/opt/itc/minica/minica.env	Переменные окружения службы: путь к конфигурации (MINICA_CONF) и параметры запуска (MINICA_OPTIONS)
config-jwt.env	/opt/itc/minica/config-jwt.env	Параметры ключа подписи JWT (тип ключа, пути к файлам) для скриптов
config-mclient.env	/opt/itc/minica/config-mclient.env	Параметры ключевой пары mclient и мастер-JWT (тип ключа, пути к файлам, имя, роль, срок) для скриптов
minica.service	/home/itc-svc/.config/systemd/user/minica.service	Пользовательский юнит systemd

make-jwt-key.sh	/opt/itc/minica/script/make-jwt-key.sh	Генерация ключа подписи JWT
make-mclient-key.sh	/opt/itc/minica/script/make-mclient-key.sh	Генерация ключевой пары mclient
make-mclient-jwt.sh	/opt/itc/minica/script/make-mclient-jwt.sh	Выпуск мастер-JWT

Шаг 1. Под учётной записью администратора скопируйте файлы дистрибутива во временный каталог на сервере ЦС, например /tmp/minica-dist. Учётная запись itc-svc должна иметь право читать эти файлы.

Шаг 2. В сеансе itc-svc создайте каталоги установки и каталог пользовательских юнитов.

```
mkdir -p /opt/itc/minica/{conf,bin,script,logs,ca/certs,ca/crl}
mkdir -p ~/.config/systemd/user
```

Шаг 3. Разложите файлы по таблице и дайте программам и скриптам право на исполнение.

```
cd /tmp/minica-dist
cp minica mclient minica.env config-jwt.env config-mclient.env /opt/itc/minica/
cp mjwt /opt/itc/minica/bin/
cp minica.yml mclient.yml openssl.cnf /opt/itc/minica/conf/
cp make-jwt-key.sh make-mclient-key.sh make-mclient-jwt.sh /opt/itc/minica/script/
cp minica.service ~/.config/systemd/user/
chmod +x /opt/itc/minica/minica /opt/itc/minica/mclient /opt/itc/minica/bin/mjwt \
  /opt/itc/minica/script/make-jwt-key.sh \
  /opt/itc/minica/script/make-mclient-key.sh \
  /opt/itc/minica/script/make-mclient-jwt.sh
# проверка: файлы принадлежат itc-svc, у программ и скриптов есть право x
ls -l /opt/itc/minica /opt/itc/minica/bin /opt/itc/minica/script
```

Шаг 4. Замените служебный заполнитель $PATH на каталог установки. В файлах дистрибутива им обозначен путь установки, который установщик подставляет автоматически. Проверочная команда просматривает все текстовые файлы каталога установки и юнит службы и не должна ничего выводить. Если она нашла заполнитель в другом файле, выполните для этого файла ту же команду sed.

```
sed -i 's#\$PATH#/opt/itc/minica#g' \
  /opt/itc/minica/minica.env /opt/itc/minica/conf/minica.yml \
  /opt/itc/minica/conf/mclient.yml \
  /home/itc-svc/.config/systemd/user/minica.service
# проверка: вывод должен быть пустым
grep -rIn '\$PATH' /opt/itc/minica /home/itc-svc/.config/systemd/user/minica.service
```

Шаг 5. Заполните в /opt/itc/minica/conf/minica.yml значения, которые установщик вычисляет при установке. В файле дистрибутива на их месте стоят конструкции шаблонизатора установщика вида `{% … %}` и `{{ … }}`. Удалите все такие конструкции, в том числе первую строку файла `{%- assign cn_value … -%}`, и задайте значения вручную:

Параметр	Назначение	Значение
ca.cdp	Точка распространения CRL (CDP), которую ЦС записывает в выпускаемые сертификаты	`["http://<сервер CDP>/cdp/<CN>.crl"]`
ca.aia	Адрес сертификата ЦС-издателя (AIA) в выпускаемых сертификатах	`["caIssuers;URI:http://<сервер CDP>/aia/<CN>.crt"]`
ca.freshest-crl	Точка распространения Delta CRL	`["http://<сервер CDP>/cdp/<CN>+.crl"]`; в поставляемом шаблоне задаётся только для выдающего ЦС (см. ниже)
database.url	Строка подключения к БД ЦС	`postgres://cwica:<ПАРОЛЬ>@<сервер БД>:5432/cwica`

Заполнители в значениях:

- `<сервер CDP>`: FQDN сервера точек распространения; установщик по умолчанию подставляет имя сервера ЦС. Если сервер CDP не используется, оставьте списки пустыми: `[]`.
- `<CN>`: значение CN из Subject сертификата ЦС, в котором пробелы заменены на «_»: для CN=RootCA подставляется RootCA, для CN=Issuing CA подставляется Issuing_CA.
- `<ПАРОЛЬ>`: пароль учётной записи cwica; специальные символы укажите в процентной кодировке (например, @ как %40).

Строка freshest-crl в файле дистрибутива обрамлена строками `{% if IS_SubCA %}` и `{% endif %}`. Для выдающего ЦС удалите только эти две строки, а значение freshest-crl заполните. Для корневого ЦС удалите строку freshest-crl вместе с ними.

Пример заполненных фрагментов для выдающего ЦС:

```
ca:
  # остальные ключи секции ca оставьте без изменений
  cdp: ["http://<сервер CDP>/cdp/Issuing_CA.crl"]
```

```
  aia: ["caIssuers;URI:http://<сервер CDP>/aia/Issuing_CA.crt"]
  freshest-crl: ["http://<сервер CDP>/cdp/Issuing_CA+.crl"]

database:
  type: postgres
  url: postgres://cwica:<ПАРОЛЬ>@<сервер БД>:5432/cwica
  allow-without-tls: true
```

В поставляемом файле `database.allow-without-tls: true`: ЦС подключается к БД и без TLS. Чтобы запретить подключение без TLS, укажите false и добавьте в url параметр `sslmode=verify-full` (см. «Подключение ядра ЦС и mOCSP к СУБД» в главе Обеспечение безопасной среды (see page 167) и Настройка ядра ЦС (see page 126)).

Проверьте, что ни в одном текстовом файле каталога установки и в юните службы не осталось конструкций шаблонизатора и заполнителя $PATH. Вывод должен быть пустым. Если команда нашла строки в другом файле, исправьте его так же, как minica.yml на шагах 4 и 5.

```
grep -rIn -E '\{%|\{\{|\$PATH' /opt/itc/minica /home/itc-svc/.config/systemd/user/
minica.service
```

5.2.4.2 Ключ и сертификат корневого ЦС

Только для корневого ЦС; для выдающего ЦС вместо этого выполните подраздел «Ключ и сертификат выдающего ЦС».

Корневой ЦС подписывает свой сертификат сам (самоподписанный сертификат). Для ЦС создаётся ключ RSA длиной 4096 бит, сертификат действует 2920 дней (8 лет). Subject указывается в формате OpenSSL, через «/»: значение установщика по умолчанию CN=RootCA,O=Company,C=RU записывается как `/CN=RootCA/O=Company/C=RU`. Замените его своим.

Шаг 1. Создайте ключ и сертификат корневого ЦС и цепочку сертификатов. Для корневого ЦС цепочка chain.pem состоит из одного его сертификата.

```
openssl genrsa -out /opt/itc/minica/ca/ca.key 4096
openssl req -x509 -new -nodes -key /opt/itc/minica/ca/ca.key -sha256 -days 2920 \
  -out /opt/itc/minica/ca/ca.crt -subj "/CN=RootCA/O=Company/C=RU"
cp /opt/itc/minica/ca/ca.crt /opt/itc/minica/ca/chain.pem
# проверка: Subject и срок действия сертификата
openssl x509 -in /opt/itc/minica/ca/ca.crt -noout -subject -dates
```

5.2.4.3 Ключ и сертификат выдающего ЦС

Только для выдающего ЦС, вместо самоподписанного сертификата из предыдущего подраздела.

Выдающий (подчинённый) ЦС получает сертификат от корневого. На сервере выдающего ЦС формируется запрос на сертификат (CSR), корневой ЦС подписывает его по шаблону subca, затем на выдающем ЦС собирается цепочка «корневой ЦС + выдающий ЦС». К этому моменту корневой ЦС должен быть полностью развёрнут, включая шаблон subca (см. «Базовые шаблоны»).

Шаг 1. На сервере выдающего ЦС в сеансе itc-svc создайте ключ и запрос. Расширение basicConstraints=critical,CA:TRUE помечает запрос как запрос сертификата ЦС. Subject замените своим.

```
openssl genrsa -out /opt/itc/minica/ca/ca.key 4096
openssl req -new -key /opt/itc/minica/ca/ca.key -out /opt/itc/minica/ca/ca.csr \
  -subj "/CN=Issuing CA/O=Company/C=RU" -addext "basicConstraints=critical,CA:TRUE"
# проверка: Subject запроса
openssl req -in /opt/itc/minica/ca/ca.csr -noout -subject
```

Шаг 2. Передайте файл /opt/itc/minica/ca/ca.csr на сервер корневого ЦС в /tmp/ca.csr защищённым способом, например через scp под учётной записью администратора. Учётная запись itc-svc корневого ЦС должна иметь право читать этот файл. На сервере корневого ЦС в сеансе itc-svc подпишите запрос: команда выпускает сертификат выдающего ЦС в файл /tmp/sub-ca.crt.

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ca -csr /tmp/ca.csr -out /tmp/sub-ca.crt -template subca
```

Шаг 3. Передайте на сервер выдающего ЦС в /tmp два файла: подписанный сертификат sub-ca.crt и сертификат корневого ЦС /opt/itc/minica/ca/ca.crt под именем root-ca.crt. Учётная запись itc-svc выдающего ЦС должна иметь право читать эти файлы. В сеансе itc-svc выдающего ЦС установите сертификат и соберите цепочку: сначала сертификат корневого ЦС, затем выдающего. Проверка должна вывести `/opt/itc/minica/ca/ca.crt: OK`.

```
cp /tmp/sub-ca.crt /opt/itc/minica/ca/ca.crt
cat /tmp/root-ca.crt /opt/itc/minica/ca/ca.crt > /opt/itc/minica/ca/chain.pem
# проверка: сертификат выдающего ЦС подписан корневым ЦС
openssl verify -CAfile /tmp/root-ca.crt /opt/itc/minica/ca/ca.crt
```

Шаг 4. Под учётной записью администратора удалите временные файлы из /tmp на обоих серверах. Часть файлов создана itc-svc, часть скопирована через scp и принадлежит администратору, а в каталоге /tmp учётная запись не может удалить чужие файлы. Поэтому команды выполняются через sudo.

```
sudo rm -f /tmp/ca.csr /tmp/sub-ca.crt       # на сервере корневого ЦС
sudo rm -f /tmp/sub-ca.crt /tmp/root-ca.crt  # на сервере выдающего ЦС
```

5.2.4.4 Ключи mclient, запуск и проверка

mclient подписывает каждый запрос к ЦС своим ключом и предъявляет JWT (токен, в котором указаны его роль и права). ЦС проверяет подпись запроса по открытому ключу mclient (conf/mclient.pem, параметр master-key), а JWT проверяет по ключам из каталога conf/jwt-keys. Первичный токен mclient называется мастер-JWT.

Шаг 1. Создайте ключ подписи JWT, ключевую пару mclient и мастер-JWT. Ключи имеют тип ED25519; мастер-JWT выпускается с ролью admin (полный доступ к API) и бессрочно. Запускайте скрипты строго в указанном порядке: каждый следующий использует файлы, созданные предыдущим. Скрипты сами переходят в каталог установки и читают параметры из config-jwt.env и config-mclient.env, поэтому их можно запускать из любого каталога.

```
/opt/itc/minica/script/make-jwt-key.sh       # -> conf/jwt.key, conf/jwt-keys/<SKI>.pem
/opt/itc/minica/script/make-mclient-key.sh   # -> conf/mclient.key, conf/mclient.pem
/opt/itc/minica/script/make-mclient-jwt.sh   # -> conf/mclient.jwt (name=master,
role=admin)
```

Скрипты в конце выводят использованные параметры. make-jwt-key.sh и make-mclient-key.sh выводят тип ключа, пути к созданным файлам и SKI ключа. make-mclient-jwt.sh выводит результат проверки выпущенного токена, пути к открытому ключу mclient и мастер-JWT, имя, роль и срок действия токена. Закрытым ключам jwt.key и mclient.key скрипты назначают права 0600, а открытым ключам права 0640.

Шаг 2. Запустите службу ЦС и включите её автозапуск. На выдающем ЦС делайте это только после установки сертификата и цепочки (подраздел «Ключ и сертификат выдающего ЦС»).

```
systemctl --user daemon-reload
systemctl --user enable minica.service
systemctl --user start minica.service
```

Шаг 3. Проверьте работу службы. Перед вызовом mclient укажите путь к его конфигурации в переменной MCLIENT_CONF. Служба должна быть в состоянии active (running), а `mclient ping` должен вывести версию ЦС и SKI. На запуск службе нужно несколько секунд: если ping сразу не проходит, повторите его.

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

5.2.4.5 Базовые шаблоны

Шаблон задаёт правила выпуска сертификата: срок действия, тип и размер ключа, расширения. Для корневого ЦС создаются шаблоны tls, subca и ocsp, для выдающего ЦС создаются tls и ocsp.

Шаблон	Для чего	Где создаётся
tls	TLS-сертификаты ЦС и сервисов: аутентификация сервера и клиента, срок 365 дней	Корневой и выдающий ЦС
subca	Сертификат выдающего ЦС, срок 1460 дней	Только корневой ЦС
ocsp	Сертификат подписи ответов OCSP-ответчика mOCSP, срок 365 дней	Корневой и выдающий ЦС

Команды выполняются в сеансе itc-svc с заданной переменной MCLIENT_CONF (см. предыдущий подраздел). Шаблон tls нужен до включения TLS: по нему выпускается TLS-сертификат ЦС.

Шаг 1. Создайте шаблон tls.

```
/opt/itc/minica/mclient addtempl -requester mclient -name "tls" -desc "tls
server&client" \
  -days "365" -key-type "RSA" -key-size "2048" -exts "tls_client_cert" -bc "" \
  -ku "critical,digitalSignature,keyEncipherment" -eku "serverAuth,clientAuth" \
  -ski "hash" -aki "keyid,issuer" -ext ""
```

Шаг 2. Только на корневом ЦС создайте шаблон subca.

```
/opt/itc/minica/mclient addtempl -requester mclient -name "subca" -desc "Subordinate
CA" \
  -days "1460" -key-type "RSA" -key-size "2048" -exts "tls_client_cert" \
  -bc "critical,CA:true,pathlen:0" -ku "critical,cRLSign,digitalSignature,keyCertSign"
 \
  -eku "serverAuth,clientAuth" -ski "hash" -aki "keyid,issuer" -ext ""
```

Шаг 3. Создайте шаблон ocsp. Параметр `-ext "noCheck=ignored"` добавляет в сертификат расширение OCSP No Check: клиенты не проверяют статус сертификата самого OCSP-ответчика.

```
/opt/itc/minica/mclient addtempl -requester mclient -name "ocsp" -desc "OCSP Signing" \
  -days "365" -key-type "RSA" -key-size "2048" -bc "CA:FALSE" \
  -ku "critical,digitalSignature" -eku "OCSPSigning" -ski "hash" -aki "keyid,issuer" \
  -ext "noCheck=ignored"
```

Проверьте, что шаблоны созданы: в списке должны быть tls, ocsp и, на корневом ЦС, subca.

```
/opt/itc/minica/mclient templates
```

5.2.4.6 Включение TLS

До этого момента ЦС принимает запросы по HTTP на порту 9080, без шифрования. После включения TLS ЦС работает только по HTTPS на порту 9443, а порт 9080 больше не используется. Выполните этот шаг сразу после создания шаблонов. Взаимная аутентификация (mTLS, параметр tls.verify) остаётся выключенной; как её включить, см. «Защита соединений центра сертификации (TLS/mTLS)».

Заполнитель `<host>` обозначает имя, по которому к ЦС обращаются другие компоненты; на это имя выпускается TLS-сертификат ЦС. Установщик по умолчанию использует имя сервера, которое выводит команда `hostname`. Имя должно разрешаться в DNS на серверах, которые подключаются к ЦС.

Команды выполняются в сеансе itc-svc с заданной переменной MCLIENT_CONF.

Шаг 1. Выпустите TLS-сертификат ЦС по шаблону tls. Первая команда создаёт ключ conf/tls.key и запрос, а вторая выпускает сертификат conf/tls.crt.

```
openssl req -new -keyout /opt/itc/minica/conf/tls.key -out /opt/itc/minica/conf/tls.req
\
  -sha256 -nodes -subj "/CN=<host>" -addext "subjectAltName = DNS:<host>" \
  -addext "extendedKeyUsage = serverAuth, clientAuth"
/opt/itc/minica/mclient ca -csr /opt/itc/minica/conf/tls.req \
  -out /opt/itc/minica/conf/tls.crt -template tls
```

Шаг 2. Включите TLS в конфигурациях ЦС и mclient. В обоих файлах в секции tls параметр enable меняется с false на true. Адрес ЦС (server-url) задаётся только в mclient.yml: в minica.yml такого параметра нет. Команды можно заменить правкой файлов в текстовом редакторе.

```
sed -i "/tls:/,/^[[:space:]]*[^[:space:]:]/ s/enable: false/enable: true/" /opt/itc/
minica/conf/minica.yml
sed -i "/tls:/,/^[[:space:]]*[^[:space:]:]/ s/enable: false/enable: true/" /opt/itc/
minica/conf/mclient.yml
sed -i "s|server-url: http://localhost:9080|server-url: https://<host>:9443|" /opt/itc/
minica/conf/mclient.yml
# проверка: под строкой tls: в обоих файлах стоит enable: true
grep -n -A1 '^tls:' /opt/itc/minica/conf/minica.yml /opt/itc/minica/conf/mclient.yml
# проверка: ожидается server-url: https://<host>:9443
grep -n '^server-url' /opt/itc/minica/conf/mclient.yml
```

Шаг 3. Перезапустите службу ЦС и проверьте подключение по HTTPS. Команда перезапуска кратковременно останавливает ЦС. `mclient ping` должен, как и при первой проверке, вывести версию ЦС и SKI, а в списке прослушиваемых портов должен остаться только 9443.

```
systemctl --user restart minica.service
/opt/itc/minica/mclient ping
ss -ltn | grep -E ':9443|:9080'
```

Файлы, которые понадобятся в следующих разделах:

- /opt/itc/minica/ca/ca.crt: сертификат ЦС. Сертификат корневого ЦС добавьте в доверенные на серверах и рабочих местах (см. «Добавление сертификата ЦС в доверенные»); сертификат выдающего ЦС в доверенные корневые не добавляют.
- /opt/itc/minica/ca/chain.pem: цепочка сертификатов ЦС.
- /opt/itc/minica/conf/tls.crt: TLS-сертификат ЦС; указывается при подключении к ЦС Контрольной панели, сервиса публикации и SCEP-адаптера.

5.2.4.7 Защита файлов ЦС

Закрытые ключи (ca/ca.key, conf/tls.key, conf/jwt.key, conf/mclient.key) и мастер-JWT (conf/mclient.jwt) дают полный доступ к ЦС. Файл minica.yml содержит пароль БД, а config-jwt.env и config-mclient.env могут содержать парольные фразы ключей (JWT_KEY_PASS, MCLIENT_KEY_PASS). Эти файлы, а также mclient.yml и minica.env должны быть доступны только технологической учётной записи (см. Ограничение прав доступа к файлам (see page 178)).

Шаг 1. В сеансе itc-svc назначьте права 0600 и проверьте результат: для каждого файла ожидается `600 itc-svc:itc-svc`.

```
chmod 600 /opt/itc/minica/ca/ca.key /opt/itc/minica/conf/tls.key \
  /opt/itc/minica/conf/jwt.key /opt/itc/minica/conf/mclient.key \
  /opt/itc/minica/conf/mclient.jwt /opt/itc/minica/conf/minica.yml \
  /opt/itc/minica/conf/mclient.yml /opt/itc/minica/minica.env \
  /opt/itc/minica/config-jwt.env /opt/itc/minica/config-mclient.env
stat -c '%a %U:%G %n' /opt/itc/minica/ca/ca.key /opt/itc/minica/conf/*.key \
  /opt/itc/minica/conf/mclient.jwt /opt/itc/minica/conf/*.yml /opt/itc/minica/*.env
```

Шаг 2. Под учётной записью администратора удалите временный каталог с файлами дистрибутива: `sudo rm -rf /tmp/minica-dist`.

> **Передача ключей и токенов**
>
> Не передавайте закрытые ключи и мастер-JWT по незащищённым каналам и не оставляйте их копии в общедоступных каталогах.

5.2.4.8 Подготовка к резервному копированию

Если ЦС будет копироваться заданиями резервного копирования, подготовьте узел ЦС и сетевой доступ к нему и к БД ЦС с сервера сервиса резервного копирования по разделу «Резервное копирование» (подраздел «Подготовка узлов ЦС и сети»).

5.2.4.9 Типовые проблемы

Что видно	Причина	Что сделать
`systemctl --user` выводит «Failed to connect to bus»	Не включён режим linger или в сеансе не заданы XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS	Выполните шаги 3 и 4 раздела Общая подготовка сервера (see page 69), затем войдите заново командой `sudo su - itc-svc`
Скрипт `make-*.sh` завершается ошибкой о файле config-jwt.env или config-mclient.env	Файлы параметров не скопированы в /opt/itc/minica	Скопируйте их (шаг 3 подраздела «Подготовка файлов и конфигурации») и запустите скрипты заново по порядку
Служба minica.service не запускается или перезапускается каждые 10 секунд	Ошибка в minica.yml (остались конструкции `{% … %}`, `{{ … }}` или $PATH), неверный database.url, БД недоступна, либо allow-without-tls: false при БД без TLS	Посмотрите журнал: `journalctl --user -u minica.service -n 50` и /opt/itc/minica/logs/minica.log; повторите проверки шагов 4 и 5 подготовки файлов
`mclient ping` сразу после запуска не отвечает	Служба ещё запускается	Повторите команду через несколько секунд
Проверка подключения к БД с сервера ЦС завершается ошибкой	Сервер СУБД не принимает сетевые подключения или порт закрыт	Проверьте listen_addresses, pg_hba.conf и правила межсетевого экрана (шаг 4 раздела Развертывание базы данных ЦС (see page 71))
После включения TLS `mclient ping` завершается ошибкой	В mclient.yml не изменён server-url, или имя `<host>` не разрешается либо не совпадает с именем в TLS-сертификате	Проверьте server-url в /opt/itc/minica/conf/mclient.yml и разрешение имени; при другом имени выпустите TLS-сертификат заново

5.2.5 Общие операции для вспомогательных сервисов

Несколько операций повторяются при развертывании разных вспомогательных сервисов: подготовка сервера, установка юнитов .NET-сервисов, выпуск JWT, подключение к ЦС и общий серверный блок Nginx. Они описаны здесь один раз, а инструкции по отдельным сервисам ссылаются на них.

Пути приведены для значений по умолчанию: каталог установки /opt/itc, технологическая учётная запись itc-svc. При иных значениях подставьте свои. Команды без sudo выполняются от

имени itc-svc (`su - itc-svc`), а команды с sudo выполняются от учётной записи с административными правами.

5.2.5.1 Подготовка сервера

Сервер вспомогательных сервисов готовится так же, как сервер ЦС (см. Общая подготовка сервера (see page 69)), но каталогом установки служит /opt/itc:

- создайте технологическую учётную запись itc-svc;
- создайте каталог /opt/itc и сделайте itc-svc его владельцем;
- включите linger для itc-svc;
- настройте окружение `systemctl --user` (XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS в ~/.bashrc);
- на серверах Контрольной панели, сервиса публикации и адаптера SCEP установите Nginx.

Каталог установки и каталог пользовательских юнитов:

```
sudo mkdir -p /opt/itc
sudo chown itc-svc:itc-svc /opt/itc
# от имени itc-svc:
mkdir -p /home/itc-svc/.config/systemd/user
```

5.2.5.2 Юниты и окружение .NET-сервисов

Контрольная панель, сервис публикации, архиватор и адаптер SCEP написаны на .NET. Они поставляются самодостаточными сборками, поэтому устанавливать среду выполнения .NET на сервер не нужно.

Сервисы читают файлы конфигурации из каталога `<ITCRoot>/itc/config` . Переменная ITCRoot задаётся в файле /opt/itc/itc_svc_env.conf, который юниты всех четырех сервисов подключают параметром EnvironmentFile. Без этой переменной Контрольная панель, сервис публикации и архиватор не запускаются.

Создайте файл окружения один раз на каждом сервере с .NET-сервисами (от имени itc-svc):

```
echo 'ITCRoot="/opt/itc"' > /opt/itc/itc_svc_env.conf
```

Юниты берутся из дистрибутива и копируются в каталог /home/itc-svc/.config/systemd/user/. В поставляемых файлах каталог установки обозначен заполнителем `$PATH` . Замените его на /opt/itc. Пример юнита Контрольной панели:

```
[Unit]
Description=itc.minica.controlPanel
```

```
[Service]
EnvironmentFile=/opt/itc/itc_svc_env.conf
WorkingDirectory=/opt/itc/itc.minica.controlPanel
ExecStart=/opt/itc/itc.minica.controlPanel/ITC.Api.MiniCA.ControlPanel --urls="http://
0.0.0.0:8407"
Restart=always
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=itc_api_minica_controlPanel

[Install]
WantedBy=default.target
```

Юниты остальных сервисов устроены так же и отличаются значениями из таблицы:

Сервис и юнит	Каталог приложения (WorkingDirectory)	Исполняемый файл	Порт	SyslogIdentifier	Каталог журналов
Контрольная панель controlPanel.service	/opt/itc/itc.minica.controlPanel	ITC.Api.MiniCA.ControlPanel	8407	itc_api_minica_controlPanel	/opt/itc/itc/logs/itc_api_miniCA_controlPanel
Сервис публикации publication.service	/opt/itc/itc.minica.publication	ITC.Api.MiniCA.Publication	9407	itc_api_minica_publication	/opt/itc/itc/logs/itc_api_miniCA_publication
Архиватор archiver.service	/opt/itc/itc.minica.archiver	ITC.Api.MiniCA.Archiver	9507	itc_api_minica_archiver	/opt/itc/itc/logs/itc_api_miniCA_archiver
Адаптер SCEP scepAdapter.service	/opt/itc/itc.acm.scepAdapter.clearwayCa	ITC.Api.Acm.Scep.Adapter.ClearwayCa	8440	itc_scep	/opt/itc/itc/logs/itc_acm_scepAdapter_clearwayCa

Прикладные порты доступны из сети

Параметр `--urls="http://0.0.0.0:<порт>"` означает, что сервис принимает подключения на всех сетевых интерфейсах. Клиенты должны обращаться к Контрольной панели, сервису публикации и адаптеру SCEP через Nginx по HTTPS, поэтому закройте порты 8407, 9407, 9507 и 8440 для внешних подключений на межсетевом экране. Исключение составляет подключение Контрольной панели к сервису публикации на другом сервере (см. «Развертывание Контрольной панели»).

5.2.5.3 Выпуск JWT для сервиса

Контрольная панель, сервис публикации и адаптер SCEP обращаются к ЦС с JWT. Токен подтверждает, что запрос пришёл от зарегистрированного сервиса, а записанная в нём роль определяет, какие функции API ЦС сервису доступны. JWT выпускается на сервере ЦС и привязывается к ключевой паре сервиса; имя сервиса сохраняется в БД ЦС.

Сервис	Имя сервиса (NAME)	Роль (ROLE)
Контрольная панель	control-panel	admin
Сервис публикации	publication-service	crluser (получение CRL); exporter (выгрузка сертификатов)
Адаптер SCEP	scep	causer (центр регистрации)

Назначайте сервису минимально необходимую роль. Состав ролей описан в подразделе «Сервисные роли JWT (ролевая модель)» раздела Настройка ядра ЦС (see page 126). При установке установщиком роль задаётся в дополнительных настройках шага «Выпуск JWT»; по умолчанию там выбрана admin.

Шаг 1. На сервере ЦС от имени itc-svc выпустите ключ, сертификат и JWT сервиса. Замените `<имя сервиса>` и `<роль>` значениями из таблицы. Вместо `<ПАРОЛЬ>` укажите парольную фразу, которой шифруется закрытый ключ на время передачи.

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
NAME=<имя сервиса>
ROLE=<роль>
openssl genpkey -algorithm RSA > /tmp/$NAME.key
openssl req -new -x509 -key /tmp/$NAME.key -out /tmp/$NAME.crt -subj "/CN=$NAME"
/opt/itc/minica/mclient mkjwt -requester "mclient" -jwt-name "$NAME" -jwt-role "$ROLE"
 \
  -jwt-key "/tmp/$NAME.crt" -out "/tmp/$NAME.jwt"
openssl pkcs8 -topk8 -in /tmp/$NAME.key -out /tmp/$NAME.encrypted.key -v2 aes-256-cbc
-passout pass:"<ПАРОЛЬ>"
```

В результате появятся файлы `/tmp/<имя сервиса>.jwt` (JWT) и `/tmp/<имя сервиса>.encrypted.key` (зашифрованный закрытый ключ).

Шаг 2. Скопируйте оба файла на сервер сервиса. Затем удалите временные файлы на сервере ЦС (команда удаляет файлы; выполняйте её в том же сеансе, где задана переменная NAME):

```
rm /tmp/$NAME.key /tmp/$NAME.crt /tmp/$NAME.jwt /tmp/$NAME.encrypted.key
```

5.2.5.4 Учётные данные для подключения к ЦС

Контрольная панель, сервис публикации и адаптер SCEP описывают подключение к ЦС блоком MiniCa. Для него на сервере сервиса в каталоге /opt/itc/itc/certs/ размещаются три файла, названных по имени подключения к ЦС:

Файл	Что это	Откуда взять
`<имя ЦС>.crt`	TLS-сертификат ЦС	С сервера ЦС: /opt/itc/minica/conf/tls.crt
`<имя ЦС>.jwt`	JWT сервиса	Файл `<имя сервиса>.jwt` (см. «Выпуск JWT для сервиса»)
`<имя ЦС>.encrypted.key`	Зашифрованный закрытый ключ сервиса	Файл `<имя сервиса>.encrypted.key` оттуда же

`<имя ЦС>` означает имя подключения, оно состоит из латинских букв и дефиса. По умолчанию используется sub-ca для выдающего ЦС и root-ca для корневого. В примерах ниже указано sub-ca.

Расшифруйте ключ на сервере сервиса от имени itc-svc и удалите зашифрованную копию. Вместо `<ПАРОЛЬ>` укажите парольную фразу, заданную при выпуске JWT.

```
openssl pkcs8 -in /opt/itc/itc/certs/sub-ca.encrypted.key -out /opt/itc/itc/certs/sub-ca.key -passin pass:"<ПАРОЛЬ>"
rm /opt/itc/itc/certs/sub-ca.encrypted.key
chmod 600 /opt/itc/itc/certs/sub-ca.key
```

Блок MiniCa ссылается на эти файлы:

```
"MiniCa": {
  "sub-ca": [
    {
      "Url": "https://ca.example.local:9443",
      "Auth": {
        "Tls": "/opt/itc/itc/certs/sub-ca.crt",
        "Sign": "/opt/itc/itc/certs/sub-ca.key",
        "Jwt": "/opt/itc/itc/certs/sub-ca.jwt"
      }
    }
  ]
}
```

Параметр	Назначение
MiniCa.`<имя ЦС>`	Имя подключения к ЦС. Значение задаётся массивом узлов ЦС: для кластера ЦС в нём перечисляется каждый узел
Url	Адрес ЦС: `https://<fqdn ЦС>:9443`
Auth.Tls	TLS-сертификат ЦС
Auth.Sign	Расшифрованный закрытый ключ сервиса, парный JWT
Auth.Jwt	JWT сервиса

Куда именно добавляется блок, указано в инструкции каждого сервиса.

5.2.5.5 Общий серверный блок Nginx

Контрольная панель, сервис публикации и адаптер SCEP публикуются через один серверный блок Nginx на порту 443. Блок подключает все файлы `/opt/itc/itc/nginx/*.conf`, а каждый сервис кладёт туда свой файл с блоками location. Если на сервере уже развёрнут один из этих сервисов, сертификат и серверный блок уже есть: пропустите эту процедуру.

В командах `<fqdn>` означает полное имя сервера служб, по которому к нему обращаются клиенты, например cp.example.local.

Шаг 1. На сервере служб от имени itc-svc сформируйте ключ и запрос на TLS-сертификат Nginx:

```
mkdir -p /opt/itc/itc/certs /opt/itc/itc/nginx
openssl req -new -keyout /opt/itc/itc/certs/nginx-tls.key -out /opt/itc/itc/certs/
nginx-tls.csr \
  -sha256 -nodes -subj "/CN=<fqdn>" -addext "subjectAltName = DNS:<fqdn>" \
  -addext "extendedKeyUsage = serverAuth, clientAuth"
chmod 600 /opt/itc/itc/certs/nginx-tls.key
```

Шаг 2. Передайте nginx-tls.csr на сервер ЦС в каталог /tmp и выпустите сертификат по шаблону tls (на сервере ЦС, от имени itc-svc):

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ca -csr /tmp/nginx-tls.csr -out /tmp/nginx-tls.crt -template
tls
```

Шаг 3. Верните сертификат на сервер служб в /opt/itc/itc/certs/nginx-tls.crt и проверьте его. В выводе subject должен содержать `CN = <fqdn>`, а issuer должен указывать на ваш ЦС.

```
openssl x509 -in /opt/itc/itc/certs/nginx-tls.crt -noout -subject -issuer -dates
```

Шаг 4. Создайте файл /opt/itc/itc/nginx/default. В дистрибутиве он поставляется с заполнителями `$HOSTNAME` и `$PATH`; итоговое содержимое:

```
server {
    listen 443 default_server ssl;
    server_name <fqdn>;

    ssl_certificate /opt/itc/itc/certs/nginx-tls.crt;
    ssl_certificate_key /opt/itc/itc/certs/nginx-tls.key;

    include /opt/itc/itc/nginx/*.conf;
}
```

Шаг 5. Подключите блок к Nginx символической ссылкой:

```
# ALT Linux:
sudo ln -sf /opt/itc/itc/nginx/default /etc/nginx/sites-enabled.d/itc.conf
# Astra Linux, Debian, РЕД ОС:
sudo ln -sf /opt/itc/itc/nginx/default /etc/nginx/conf.d/itc.conf
```

Перезагружайте Nginx после того, как в /opt/itc/itc/nginx/ добавлен файл сервиса; это указано в инструкции каждого сервиса. Перед перезагрузкой проверяйте синтаксис: команда `sudo nginx -t` должна сообщить `syntax is ok` и `test is successful`.

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

5.2.6 Развертывание сервиса аутентификации OpenID

Сервис OpenID служит провайдером OpenID Connect (форк Dex). Он обеспечивает единый вход пользователей во все компоненты Clearway CA: проверяет учётную запись в каталоге LDAP или Active Directory и сопоставляет группы каталога с правами Clearway CA. Права передаются в токене, который сервис выдаёт при входе.

Состояние сервиса (сессии, refresh-токены, ключи подписи) хранится в собственной БД PostgreSQL. Сервис слушает HTTPS на порту 443 и служебный HTTP-интерфейс администрирования на порту 5559. Параметры конфигурации описаны в подразделе «Сервис аутентификации OpenID» раздела Настройка вспомогательных сервисов (see page 139).

Что понадобится:

- сервер, подготовленный по «Подготовка сервера»;

- LDAP-клиент для проверки каталога: openldap-clients (ALT Linux, РЕД ОС) или ldap-utils (Astra Linux, Debian);
- доступ к серверу PostgreSQL с правами postgres;
- учётная запись для чтения каталога (bindDN и пароль), базовые DN пользователей и групп;
- группы каталога для ролей Clearway CA (см. Описание ролевой модели (see page 47));
- адрес Контрольной панели, например `https://cp.example.local`;
- из дистрибутива: исполняемый файл openid, юнит openid.service, шаблон config.yaml.

Шаг 1. Создайте базу данных OpenID. Схему сервис создаёт сам при первом запуске, поэтому нужны только пользователь-владелец и пустая БД (по умолчанию пользователь cwi, БД openid). Выполните команды на сервере PostgreSQL, заменив `<ПАРОЛЬ>`:

```
sudo -u postgres psql -c "CREATE USER \"cwi\" WITH PASSWORD '<ПАРОЛЬ>';"
sudo -u postgres psql -c "CREATE DATABASE openid OWNER \"cwi\";"
sudo -u postgres psql -c "GRANT CONNECT, CREATE ON DATABASE \"openid\" TO \"cwi\";"
sudo -u postgres psql -d openid -c "GRANT USAGE, CREATE ON SCHEMA public TO \"cwi\";"
sudo -u postgres psql -d openid -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL
ON TABLES TO \"cwi\";"
sudo -u postgres psql -d openid -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL
ON SEQUENCES TO \"cwi\";"
```

Шаг 2. На сервере OpenID от имени itc-svc сформируйте ключ и запрос на TLS-сертификат сервиса:

```
mkdir -p /opt/itc/itc/certs
openssl req -new -keyout /opt/itc/itc/certs/openid-tls.key -out /opt/itc/itc/certs/
openid-tls.csr \
  -sha256 -nodes -subj "/CN=openid.example.local" \
  -addext "subjectAltName = DNS:openid.example.local" \
  -addext "extendedKeyUsage = serverAuth, clientAuth"
chmod 600 /opt/itc/itc/certs/openid-tls.key
```

Шаг 3. Передайте openid-tls.csr на сервер ЦС в каталог /tmp и выпустите сертификат по шаблону tls (на сервере ЦС, от имени itc-svc). Верните сертификат на сервер OpenID в /opt/itc/itc/certs/openid-tls.crt.

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ca -csr /tmp/openid-tls.csr -out /tmp/openid-tls.crt -template
tls
```

Шаг 4. Создайте каталог сервиса (`mkdir -p /opt/itc/openid`) и разложите файлы по каталогам:

Файл из дистрибутива	Куда поместить
openid	/opt/itc/openid/openid
openid.service	/home/itc-svc/.config/systemd/user/openid.service
config.yaml	/opt/itc/openid/config.yaml

В юните замените заполнитель `$PATH` на /opt/itc. Итоговое содержимое:

```
[Unit]
Description=itc.openid
After=syslog.target network.target

[Service]
WorkingDirectory=/opt/itc/openid
ExecStart=/opt/itc/openid/openid serve /opt/itc/openid/config.yaml
Restart=always
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=itc_api_openid

[Install]
WantedBy=default.target
```

Шаг 5. Проверьте доступ к каталогу с сервера OpenID. Команда выполняет вход (bind) учётной записью сервиса и запрашивает её пароль. Замените адрес каталога, bindDN и базовый DN своими значениями. Если команда завершилась без ошибок, адрес каталога, учётная запись и базовый DN указаны верно.

```
ldapsearch -x -H ldap://dc.example.local:389 -D 'CN=svc-
openid,OU=Users,DC=example,DC=local' -W \
  -b 'ou=Users,dc=example,dc=local' -s base '1.1'
```

Шаг 6. Заполните /opt/itc/openid/config.yaml, заменив примеры и заполнители своими значениями. Карта ldapGroupRolesMap в примере соответствует стандартным ролям: администратор ЦС получает все 17 прав.

```
issuer: https://openid.example.local          # канонический адрес сервиса (claim
"iss")

storage:                                       # состояние сервиса хранится в своей БД
 PostgreSQL
  type: postgres
```

```
  config:
    host: "db.example.local"
    port: 5432
    database: "openid"
    user: "cwi"
    password: "<ПАРОЛЬ>"
    ssl:
      mode: disable                          # подключение к БД без TLS

web:
  https: "openid.example.local:443"          # 443: привилегированный порт (см.
setcap ниже)
  tlsCert: "/opt/itc/itc/certs/openid-tls.crt"
  tlsKey: "/opt/itc/itc/certs/openid-tls.key"
  allowedOrigins: ['*']                      # источники (origin), из которых браузер
может обращаться к сервису

adminServer:
  http: "openid.example.local:5559"          # служебный порт администрирования

connectors:
- type: ldap
  id: ldap
  name: LDAP
  config:
    host: "dc.example.local:389"             # сервер каталога (Active Directory)
    bindDN: "CN=svc-openid,OU=Users,DC=example,DC=local"
    bindPW: "<ПАРОЛЬ>"
    insecureNoSSL: true                      # подключение к каталогу по LDAP без TLS
    import:                                  # периодический импорт из каталога
      active: true
      timer: 1h
      objects:
        - person
        - groups
    userSearch:
      onDemand: true
      baseDN: "ou=Users,dc=example,dc=local"
      filter: "(objectClass=person)"
      username: "sAMAccountName"
      idAttr: "sAMAccountName"
      emailAttr: "userPrincipalName"
      nameAttr: "displayName"
      preferredUsernameAttr: "sAMAccountName"
    groupSearch:
      onDemand: true
      baseDN: "ou=Groups,dc=example,dc=local"
      filter: "(objectClass=group)"
      userMatchers:
      - userAttr: "DN"
        groupAttr: "member"
      nameAttr: "cn"
    ldapGroupRolesMap:                       # сопоставление групп каталога правам
 Clearway CA
      - ldapGroup: CWCA_Administrator        # администратор ЦС: все 17 прав
        roles:
```

```
          - rleMiniCACertList-r
          - rleMiniCATemplList-r
          - rleMiniCAReqList-r
          - rleMiniCAEventList-r
          - rleMiniCACRL-r
          - rleMiniCAConfig-r
          - rleMiniCACertReq-w
          - rleMiniCACertRev-w
          - rleMiniCACertUnRev-w
          - rleMiniCATempl-w
          - rleMiniCACRL-w
          - rleMiniCAConfig-w
          - rleMiniCAReqApprove-w
          - rleMiniCABackup-r
          - rleMiniCABackup-w
          - rleMiniCABackupRestore-w
          - rleMiniCARotation-w
      - ldapGroup: CWCA_Auditor                # аудитор: только просмотр
        roles:
          - rleMiniCACertList-r
          - rleMiniCAEventList-r
          - rleMiniCATemplList-r
          - rleMiniCACRL-r
          - rleMiniCAConfig-r
          - rleMiniCAReqList-r
      - ldapGroup: CWCA_User                   # пользователь: просмотр и запрос
сертификатов
        roles:
          - rleMiniCACertList-r
          - rleMiniCATemplList-r
          - rleMiniCACertReq-w

expiry:
  idTokens: "2m"                               # время жизни access-токена
  authRequests: "10m"                          # время на прохождение авторизации
  signingKeysAlgorithm: "RS256"                # алгоритм подписи токенов (по
умолчанию)
  signingKeys: "6h"                            # период ротации ключей подписи
  refreshTokens:
    validIfNotUsedFor: "3m"                    # refresh-токен истекает, если не
используется
    sessionLimit: 1                            # ограничение числа одновременных сессий
    sessionLimitPolicy: "reject"               # reject: отклонять новую сессию;
terminate_oldest: завершать самую старую
    absoluteLifetime: "24h"                    # максимальный срок жизни refresh-токена

staticClients:                                 # клиенты, которым OpenID выдает токены
  - name: clearway
    id: itc-clearway                           # этот id указывается в client_env.js
панели
    secret: <СЕКРЕТ>
    redirectURIs:
      - https://cp.example.local               # адрес Контрольной панели
```

Что указать в основных параметрах:

Параметр	Откуда взять
issuer	Адрес сервиса вида `https://<fqdn>`. Записывается в токены и должен быть доступен всем клиентам: браузерам пользователей, Контрольной панели, адаптеру SCEP
storage.config	Сервер PostgreSQL, БД и пароль пользователя cwi из шага 1
web.https	Имя сервера OpenID и порт 443. Сервис слушает порт на адресе, в который разрешается это имя
connectors[].config.host	Сервер каталога в формате `<сервер>:<порт>`
bindDN, bindPW	Учётная запись для чтения каталога, проверенная в шаге 5
userSearch.baseDN, groupSearch.baseDN	Контейнеры каталога с учётными записями пользователей и с группами
ldapGroupRolesMap	Группы каталога и права Clearway CA. Состав прав стандартных ролей приведён в разделе Описание ролевой модели (see page 47)
staticClients	Клиент Контрольной панели: id (по умолчанию itc-clearway), секрет и адрес панели в redirectURIs



Имена групп и изменение прав

Если имена групп каталога отличаются от CWCA_Administrator, CWCA_Auditor и CWCA_User, измените значения ldapGroup, оставив перечни прав без изменения. Карта применяется после перезапуска openid.service. Права передаются в токене при входе, поэтому после изменения карты пользователям нужно выйти из Контрольной панели и войти снова.

Конфигурация, которую создаёт установщик, дополнительно включает встроенную учётную запись для входа без каталога (параметры enablePasswordDB и staticPasswords, права задаются в customClaims.roles). В примере выше её нет: пользователи входят через каталог.

Шаг 7. Ограничьте доступ к файлу конфигурации: он содержит пароли БД и каталога и секрет клиента.

```
chmod 600 /opt/itc/openid/config.yaml
```

Шаг 8. Запустите сервис. Порт 443 привилегированный (меньше 1024), поэтому право на привязку к нему выдаётся исполняемому файлу через setcap:

```
chmod +x /opt/itc/openid/openid
sudo setcap 'cap_net_bind_service=+ep' /opt/itc/openid/openid   # право слушать 443 без
 root
systemctl --user daemon-reload
systemctl --user enable openid.service
systemctl --user start openid.service
```

Шаг 9. На межсетевом экране откройте входящий порт 443 (вход клиентов) и исходящие подключения к 5432 (БД) и 389 (LDAP). Порт 5559 (служебный интерфейс администрирования) клиентам не открывайте; при необходимости разрешите подключение к нему только с АРМ администраторов.

Проверка. От имени itc-svc проверьте состояние службы и ответ сервиса. Служба должна быть в состоянии `active (running)`, а сервис должен вернуть JSON, в котором поле issuer совпадает с issuer из config.yaml.

```
systemctl --user status openid.service
curl --cacert <корневой сертификат ЦС> https://openid.example.local/.well-known/openid-
configuration
```

Типовые проблемы:

Что видно	Причина	Что сделать
Служба не запускается, в журнале (`journalctl --user -u openid.service`) ошибка привязки к порту 443	У исполняемого файла нет права cap_net_bind_service. Право теряется при замене файла, например при обновлении	Повторите команду setcap из шага 8 и перезапустите службу
Пользователь входит, но не видит разделов Контрольной панели	Группы пользователя в каталоге не совпадают с ldapGroup в карте ldapGroupRolesMap	Сверьте имена групп, исправьте карту, перезапустите openid.service; пользователю нужно выйти и войти снова

5.2.7 Развертывание Keycloak (вместо OpenID)

Инструкция описывает настройки Keycloak, которые нужны Контрольной панели и веб-интерфейсу ЦС релиза 26.3. Вместо Keycloak может использоваться собственный провайдер Clearway Open ID; выбор делается при установке, параметр «Тип провайдера аутентификации». При выборе Keycloak шаги установки Clearway Open ID пропускаются.

Установка

Пример для Keycloak 26.x на Linux-сервере (Astra Linux, РЕД ОС, ALT Linux, Debian). Нужны JDK 21 и дистрибутив Keycloak (zip); в закрытый контур он доставляется отдельно.

Шаг 1. Установите JDK 21:

```
sudo apt-get install -y openjdk-21-jdk   # Debian, Astra Linux
sudo dnf install -y java-21-openjdk      # РЕД ОС
sudo apt-get install -y java-21-openjdk  # ALT Linux
```

Шаг 2. Создайте пользователя и распакуйте дистрибутив:

```
sudo useradd -r -d /opt/keycloak -s /usr/sbin/nologin keycloak
sudo unzip keycloak-26.3.x.zip -d /opt
sudo chown -R keycloak:keycloak /opt/keycloak
```

Шаг 3. Подготовьте базу в PostgreSQL. Встроенная база Keycloak (H2) пригодна только для тестов, в производственном режиме нужна внешняя СУБД. Схему Keycloak создаст сам при первом старте:

```
CREATE ROLE keycloak LOGIN PASSWORD '<ПАРОЛЬ>';
CREATE DATABASE keycloak OWNER keycloak;
```

Шаг 4. Заполните `/opt/keycloak/conf/keycloak.conf`:

```
hostname=https://kc.example.local
https-certificate-file=/opt/keycloak/conf/tls.crt
https-certificate-key-file=/opt/keycloak/conf/tls.key
db=postgres
db-url=jdbc:postgresql://localhost:5432/keycloak
db-username=keycloak
db-password=<ПАРОЛЬ>
```

Параметр `hostname` задаёт внешний адрес, по которому Keycloak доступен браузерам и Контрольной панели; от него зависит issuer.

> ⚠ **TLS на реверс-прокси**
>
> Если TLS терминируется на реверс-прокси, сертификаты в `keycloak.conf` не указываются. Keycloak настраивается за прокси (параметр `proxy`), а `hostname` остаётся внешним адресом.

Шаг 5. Создайте администратора при первом старте (переменные `KC_BOOTSTRAP_ADMIN_USERNAME` и `KC_BOOTSTRAP_ADMIN_PASSWORD`) и установите автозапуск:

```
[Unit]
Description=Keycloak
After=network.target

[Service]
User=keycloak
Environment=KC_BOOTSTRAP_ADMIN_USERNAME=admin
Environment=KC_BOOTSTRAP_ADMIN_PASSWORD=<ПАРОЛЬ>
ExecStart=/opt/keycloak/bin/kc.sh start
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

После первого старта строки `Environment` можно убрать.

Шаг 6. Проверьте установку: состояние службы показывает `systemctl status keycloak`, консоль администратора открывается по адресу сервера, документ `https://kc.example.local/realms/master/.well-known/openid-configuration` отвечает.

Дальше realm, клиент, роли и пользователи настраиваются по разделам ниже.

Взаимодействие ЦС с Keycloak

- Веб-интерфейс выполняет вход по Authorization Code Flow: браузер уходит на страницу Keycloak и возвращается на адрес Контрольной панели.
- Контрольная панель проверяет access token в каждом запросе: подпись по ключам издателя (JWKS), совпадение issuer, срок действия, роли пользователя.
- Административный API SCEP проверяет токены того же издателя.

Значения из мастера установки:

Параметр в инсталляторе	Назначение
Тип провайдера аутентификации = Keycloak	Режим Keycloak

Адрес издателя (issuer)	Формат `https://<fqdn>/realms/<realm>`. Записывается в конфиг Контрольной панели (`OpenId.Authority`) и в `client_env.js` веб-интерфейса
Идентификатор клиента	Client id, под которым веб-интерфейс зарегистрирован в Keycloak
Claim для ролей JWT	Имя claim с ролями в токене, по умолчанию `roles`

Требования к серверу

- HTTPS и сертификат, которому доверяют браузеры пользователей. Контрольная панель проверку сертификата Keycloak не выполняет (`IgnoreCertificateErrors = true`), браузер проверяет всегда.
- Адрес издателя должен отвечать по одному и тому же URL из браузера и с сервера Контрольной панели: при проверке токенов панель читает `https://<issuer>/.well-known/openid-configuration`. Если Keycloak стоит за реверс-прокси, настройте `hostname` так, чтобы `iss` в токенах совпадал с внешним адресом.
- Синхронизация времени (NTP): допустимое расхождение при проверке срока действия токена равно нулю.

Импорт конфигурации

Файл ниже создаёт всё, что нужно ЦС: realm, клиент `itc-minica` с mapper'ом ролей, все роли и композитную роль `CWCA_Administrator` (даёт все роли разом).

Импорт: для нового realm боковое меню → Manage realms → Create realm → загрузить файл. Если realm уже существует: Realm settings → Partial import → файл → Import (режим Overwrite).

```
{
  "realm": "clearway",
  "displayName": "Clearway CA",
  "enabled": true,
  "sslRequired": "external",
  "registrationAllowed": false,
  "roles": {
    "realm": [
      { "name": "rleMiniCACertReq-w", "description": "Выпуск сертификатов" },
      { "name": "rleMiniCACertRev-w", "description": "Отзыв сертификатов" },
      { "name": "rleMiniCACertUnRev-w", "description": "Возврат отозванных
сертификатов" },
      { "name": "rleMiniCAReqApprove-w", "description": "Подтверждение запросов на
сертификат" },
      { "name": "rleMiniCATempl-w", "description": "Редактирование шаблонов" },
      { "name": "rleMiniCATemplList-r", "description": "Просмотр шаблонов" },
```

```
      { "name": "rleMiniCAConfig-w", "description": "Изменение конфигурации ЦС" },
      { "name": "rleMiniCAConfig-r", "description": "Просмотр конфигурации" },
      { "name": "rleMiniCACRL-w", "description": "Управление публикацией CRL" },
      { "name": "rleMiniCACRL-r", "description": "Просмотр конфигурации CRL" },
      { "name": "rleMiniCACertList-r", "description": "Просмотр сертификатов" },
      { "name": "rleMiniCAReqList-r", "description": "Просмотр запросов" },
      { "name": "rleMiniCAEventList-r", "description": "Просмотр журнала событий" },
      { "name": "rleOpenId-r", "description": "Просмотр сессий OpenID" },
      { "name": "rleOpenId-w", "description": "Завершение сессий OpenID" },
      {
        "name": "CWCA_Administrator",
        "description": "Все роли ЦС",
        "composite": true,
        "composites": {
          "realm": [
            "rleMiniCACertReq-w",
            "rleMiniCACertRev-w",
            "rleMiniCACertUnRev-w",
            "rleMiniCAReqApprove-w",
            "rleMiniCATempl-w",
            "rleMiniCATemplList-r",
            "rleMiniCAConfig-w",
            "rleMiniCAConfig-r",
            "rleMiniCACRL-w",
            "rleMiniCACRL-r",
            "rleMiniCACertList-r",
            "rleMiniCAReqList-r",
            "rleMiniCAEventList-r",
            "rleOpenId-r",
            "rleOpenId-w"
          ]
        }
      }
    ]
  },
  "clients": [
    {
      "clientId": "itc-minica",
      "name": "Clearway CA",
      "enabled": true,
      "protocol": "openid-connect",
      "publicClient": true,
      "standardFlowEnabled": true,
      "directAccessGrantsEnabled": false,
      "redirectUris": [ "https://ca.example.local/*" ],
      "webOrigins": [ "https://ca.example.local" ],
      "attributes": {
        "post.logout.redirect.uris": "https://ca.example.local/*"
      },
      "defaultClientScopes": [ "profile", "email", "roles", "web-origins", "acr",
"basic" ],
      "optionalClientScopes": [ "offline_access" ],
      "protocolMappers": [
        {
          "name": "realm roles",
          "protocol": "openid-connect",
```

```
            "protocolMapper": "oidc-usermodel-realm-role-mapper",
            "consentRequired": false,
            "config": {
              "claim.name": "roles",
              "jsonType.label": "String",
              "multivalued": "true",
              "id.token.claim": "true",
              "access.token.claim": "true",
              "userinfo.token.claim": "false"
            }
          }
        ]
      }
    ]
}
```

После импорта поправьте под свой стенд:

- имя realm: оно входит в issuer. В инсталляторе параметр «Адрес издателя (issuer)» = `https://<fqdn>/realms/clearway`;
- `redirectUris` и `webOrigins`: адрес Контрольной панели вместо `ca.example.local`;
- пользователи создаются отдельно (см. «Пользователи»); администратору достаточно роли `CWCA_Administrator`;
- если параметр «Claim для ролей JWT» в инсталляторе отличается от `roles`, переименуйте `claim.name`[1] в mapper'е.

Разделы «Клиент» и «Роли» описывают те же настройки вручную; при импорте они уже выполнены.

Клиент
Раздел описывает ручную настройку; при импорте конфигурации (см. «Импорт конфигурации») клиент уже создан.

Clients → Create client:

- Client type: OpenID Connect; Client ID: то же значение, что указано в параметре «Идентификатор клиента» инсталлятора (например `itc-minica`);
- Client authentication: Off. Клиент публичный, секрет не используется;
- Authentication flow: Standard flow; Direct access grants: Off;
- Valid redirect URIs: адрес Контрольной панели, например `https://ca.example.local/*`;
- Web origins: адрес Контрольной панели.

> ⚠ **Offline access**

1 http://claim.name

Веб-интерфейс обновляет access token через refresh token. Стандартный scope запроса: `openid profile email offline_access`. Если вход падает с ошибкой `invalid_scope` по `offline_access`, назначьте пользователям встроенную роль Offline Access или уберите `offline_access` из scope в `client_env.js` (параметр `scope`).

Роли

Раздел описывает ручную настройку; при импорте конфигурации роли уже созданы.

Контрольная панель и веб-интерфейс читают роли из claim, имя которого задано параметром «Claim для ролей JWT» (по умолчанию `roles`). Keycloak по умолчанию пишет realm-роли в `realm_access.roles`, поэтому нужен protocol mapper.

Clients → <клиент> → Client scopes → <клиент>-dedicated → Add mapper → By configuration → User Realm Role:

- Token Claim Name: `roles` (должно совпадать с параметром «Claim для ролей JWT»);
- Multivalued: On;
- Add to access token: On;
- Claim JSON Type: String.

Роли уровня realm и их назначение:

Роль	Доступ
`rleMiniCACertReq-w`	Выпуск сертификатов
`rleMiniCACertRev-w`	Отзыв сертификатов
`rleMiniCACertUnRev-w`	Возврат отозванных сертификатов
`rleMiniCAReqApprove-w`	Подтверждение запросов на сертификат
`rleMiniCATempl-w`	Редактирование шаблонов
`rleMiniCATemplList-r`	Просмотр шаблонов
`rleMiniCAConfig-w`	Изменение конфигурации ЦС
`rleMiniCAConfig-r`	Просмотр конфигурации

`rleMiniCACRL-w`	Управление публикацией CRL (период, DeltaCRL, ручная публикация)
`rleMiniCACRL-r`	Просмотр конфигурации CRL
`rleMiniCACertList-r`	Просмотр сертификатов
`rleMiniCAReqList-r`	Просмотр запросов
`rleMiniCAEventList-r`	Просмотр журнала событий
`rleOpenId-r`	Просмотр сессий OpenID
`rleOpenId-w`	Завершение сессий OpenID (работает только с Clearway Open ID)

Администратору ЦС назначьте все роли (или композитную `CWCA_Administrator`), остальным пользователям по потребности.

Пользователи

Создайте пользователей realm и назначьте роли напрямую или через группы. Для входа используются логин и пароль realm.

Ограничения

Раздел «Сессии OpenID» в Контрольной панели обращается к административному API Clearway Open ID. При Keycloak он не работает; на остальную функциональность это не влияет.

Проверка

Подключение с сервера Контрольной панели:

```
curl -f https://<fqdn>/realms/<realm>/.well-known/openid-configuration
```

Документ должен открываться без ошибок TLS.

Выполните вход в веб-интерфейс и проверьте access token: роли должны быть в claim с настроенным именем. Состав токена виден в Keycloak или на jwt.io[2].

Разделы веб-интерфейса открываются согласно назначенным ролям.

2 http://jwt.io

5.2.8 Развертывание Контрольной панели

Контрольная панель служит веб-интерфейсом управления ЦС. Технически это два приложения за Nginx: серверная часть ITC.Api.MiniCA.ControlPanel (порт 8407) и статическое одностраничное веб-приложение (SPA) ITC.MiniCA.Spa. Nginx публикует панель по HTTPS на порту 443: раздаёт SPA и файлы настроек браузера, а запросы `/api/minica` передаёт серверной части.

Пользователи входят в панель через сервис OpenID. К ЦС серверная часть обращается с JWT, роль admin. Параметры конфигурации описаны в подразделе «Контрольная панель» раздела Настройка вспомогательных сервисов (see page 139).

Что понадобится:

- сервер, подготовленный по «Подготовка сервера», с установленным Nginx;
- файл окружения /opt/itc/itc_svc_env.conf и юнит controlPanel.service (см. «Юниты и окружение .NET-сервисов»);
- TLS-сертификат Nginx и серверный блок (см. «Общий серверный блок Nginx»);
- развёрнутый сервис OpenID, в redirectURIs клиента itc-clearway которого указан адрес панели;
- JWT и ключ панели: имя control-panel, роль admin (см. «Выпуск JWT для сервиса»);
- утилита jq, если ЦС подключается из командной строки;
- из дистрибутива: архивы ITC.Api.MiniCA.ControlPanel.tar.gz и ITC.MiniCA.Spa.tar.gz, файлы client_env.js, spa.config.json, brand.svg, logo.svg, favicon.ico и блок Nginx панели (nginx.conf).

Шаг 1. Создайте каталоги и распакуйте серверную часть и SPA (от имени itc-svc):

```
mkdir -p /opt/itc/itc.minica.controlPanel /opt/itc/itc.minica.spa /opt/itc/itc/
{config,env,nginx,certs} \
  /opt/itc/itc/logs/itc_api_miniCA_controlPanel
tar -xzf /tmp/ITC.Api.MiniCA.ControlPanel.tar.gz -C /opt/itc/itc.minica.controlPanel/
tar -xzf /tmp/ITC.MiniCA.Spa.tar.gz              -C /opt/itc/itc.minica.spa/
```

Шаг 2. Настройте серверную часть в файле /opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json:

```
{
  "OpenId": {
    "Authority": "https://openid.example.local",
    "IgnoreCertificateErrors": true,
    "JwtRoleClaim": null
  },
  "Services": {
    "OpenId": "https://openid.example.local",
    "PublicationCrlUrl": "http://localhost:9407"
  },
  "Features": {
    "TypicalTask": true
```

```
  }
}
```

В `OpenId.Authority` и `Services.OpenId` укажите адрес сервиса OpenID (issuer) вида `https://<fqdn OpenID>`. Назначение всех параметров и значения по умолчанию приведены в подразделе «Параметры серверной части» раздела Настройка вспомогательных сервисов (see page 139).

Если сервис публикации развёрнут на другом сервере, укажите в Services.PublicationCrlUrl адрес `http://<сервер публикации>:9407` и разрешите подключение к порту 9407 только с сервера панели.

Раздел «Задачи → Резервное копирование» работает, только если в секции Services задан параметр `BackupUrl` с адресом сервиса резервного копирования, например `"BackupUrl": "http://localhost:8409"`, если сервис работает на сервере панели. Задания копирования выполняет этот сервис, а панель передаёт ему запросы (см. «Сервис резервного копирования» в разделе Настройка вспомогательных сервисов (see page 139)).

Шаг 3. Настройте клиент (SPA). Создайте файл /opt/itc/itc/env/client_env.js; clientId должен совпадать с id клиента в staticClients сервиса OpenID:

```
(function () {
  window.itcenv = {
    oid: {
      issuer: 'https://openid.example.local',
      clientId: 'itc-clearway',
      secret: null,
      rolesClaim: 'roles',
      useSilentRefresh: false
    },
  };
})();
```

Параметр	Назначение
issuer	Адрес сервиса OpenID, как issuer в config.yaml
clientId	id клиента Контрольной панели в OpenID (по умолчанию itc-clearway)
secret	Оставьте null. Файл раздаётся браузерам по адресу `https://<fqdn>/assets/env/client_env.js`, и указанное в нём значение доступно любому

rolesClaim	Claim ID-токена, из которого панель читает права пользователя; по умолчанию roles
useSilentRefresh	Тихое обновление токена. Для OpenID задайте false; для Keycloak задайте true, если есть проблемы с обновлением access-токена. При true адрес `<адрес панели>/silent-refresh.html` должен быть разрешён у клиента OIDC
scope	Необязательный параметр. Если не задан, запрашиваются области openid profile email offline_access

В тот же каталог /opt/itc/itc/env/ скопируйте из дистрибутива файлы brand.svg, logo.svg, favicon.ico и файл настроек интерфейса spa.config.json под именем config.json.

Шаг 4. Настройте Nginx. Создайте блок панели /opt/itc/itc/nginx/controlPanel.conf (в дистрибутиве он называется nginx.conf; заполнитель `$PATH` замените на /opt/itc):

```
location / {
    root /opt/itc/itc.minica.spa;
    index index.html;
    try_files $uri $uri/ /index.html?$args;
}

location =/favicon.ico {
    alias /opt/itc/itc/env/favicon.ico;
}

location /assets/env {
    alias /opt/itc/itc/env;
}

location /api/minica {
    proxy_pass http://localhost:8407/api;
    proxy_http_version 1.1;
    proxy_cache_bypass $http_upgrade;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Connection keep-alive;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

Проверьте конфигурацию и перезагрузите Nginx:

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

Шаг 5. Поместите юнит controlPanel.service в /home/itc-svc/.config/systemd/user/ (см. «Юниты и окружение .NET-сервисов») и запустите сервис:

```
systemctl --user daemon-reload
systemctl --user enable controlPanel.service
systemctl --user start controlPanel.service
```

После запуска панель доступна по адресу `https://cp.example.local`.

5.2.8.1 Подключение ЦС к панели

Панель управляет одним или несколькими ЦС. Каждый ЦС подключается отдельно: через интерфейс или из командной строки. Для подключения нужны TLS-сертификат ЦС, зашифрованный закрытый ключ и JWT панели (см. «Учётные данные для подключения к ЦС»). Повторите подключение для каждого ЦС, которым управляет панель.

Способ 1. Через интерфейс. Войдите в панель под учётной записью с правом rleMiniCAConfig-w и откройте «Настройки → Конфигурация → Действия → Добавить новый ЦС». Заполните поля и нажмите «Сохранить»:

Поле	**Что указать**
Наименование	Имя подключения. После сохранения не меняется
Url	Адрес ЦС, например `https://ca.example.local:9443`
Пароль	Парольная фраза зашифрованного ключа
Сертификат Tls	TLS-сертификат ЦС (.crt, .cer или .pem): файл /opt/itc/minica/conf/tls.crt с сервера ЦС
Зашифрованный приватный ключ	Файл control-panel.encrypted.key, созданный при выпуске JWT
Jwt	Файл control-panel.jwt

Ключ загружается зашифрованным, расшифровывать его не нужно. После сохранения ЦС появляется в карточке «Выбор Clearway CA» у всех пользователей. Подключения, добавленные через интерфейс, панель хранит в файле /opt/itc/itc/config/itc_api_miniCA_config.runtime.json. Подробно работа со страницей описана в Руководстве пользователя, глава «Настройки», раздел «Конфигурация».

Способ 2. Из командной строки. Разместите файлы и расшифруйте ключ по «Учётные данные для подключения к ЦС». Затем от имени itc-svc добавьте блок MiniCa в конфигурацию серверной части и перезапустите сервис. Команда перезаписывает файл конфигурации; в примере имя подключения sub-ca.

```
jq '.MiniCa["sub-ca"] += [{
  "Url": "https://ca.example.local:9443",
  "Auth": {
    "Tls": "/opt/itc/itc/certs/sub-ca.crt",
    "Sign": "/opt/itc/itc/certs/sub-ca.key",
    "Jwt": "/opt/itc/itc/certs/sub-ca.jwt"
  }
}]' /opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json > /tmp/
itc_api_miniCA_controlPanel_config.json \
  && mv /tmp/itc_api_miniCA_controlPanel_config.json /opt/itc/itc/config/
itc_api_miniCA_controlPanel_config.json
systemctl --user restart controlPanel.service
```

Серверная часть читает файл itc_api_miniCA_controlPanel_config.json после файла itc_api_miniCA_config.runtime.json. Поэтому если один и тот же ЦС описан в обоих файлах, действуют значения из itc_api_miniCA_controlPanel_config.json.

Проверка.

- `systemctl --user status controlPanel.service` показывает `active (running)`.
- По адресу `https://cp.example.local` открывается панель, и пользователь из группы администраторов каталога входит в неё.
- На странице «Настройки → Конфигурация» в карточке «Выбор Clearway CA» есть подключённый ЦС.
- У карточек «Конфигурация Clearway CA», «Конфигурация контрольной панели» и «Конфигурация сервиса публикации CRL» зелёный индикатор и указана версия; карточка сервиса публикации зелёная после его развертывания.

Сетевой доступ:

Направление	Порт	Назначение
Браузеры пользователей → сервер панели	443/tcp	Интерфейс Контрольной панели
Браузеры пользователей → сервер OpenID	443/tcp	Вход в панель
Сервер панели → сервер OpenID	443/tcp	Проверка токенов

Сервер панели → ЦС	9443/tcp	Операции с ЦС
Сервер панели → сервис публикации	9407/tcp	Если сервис публикации на другом сервере
Сервер панели → сервис резервного копирования	Порт сервиса, в примерах 8409/tcp	Раздел «Задачи → Резервное копирование», если сервис на другом сервере
Сервер панели → серверы с Nginx и PostgreSQL	22/tcp	Типовые задачи замены сертификатов

Порт 8407 закройте для внешних подключений. Подключения к узлам ЦС (22/tcp) и к серверу БД ЦС (5432/tcp) для резервного копирования и восстановления выполняет сервис резервного копирования со своего сервера, а не серверная часть панели (см. «Резервное копирование»).

5.2.9 Развертывание сервиса публикации

Сервис публикации (ITC.Api.MiniCA.Publication) работает как фоновый .NET-сервис (пользовательский юнит systemd, порт 9407). По расписанию cron он публикует списки отзыва (CRL и Delta CRL) в локальный каталог, на удалённый сервер по SCP или в каталог LDAP, а выпущенные сертификаты публикует в каталог LDAP (Active Directory или Samba AD).

Сервис публикуется через Nginx (TLS 443, путь `/api/publication`); входящие запросы авторизуются через OpenID/JWT. Контрольная панель обращается к сервису по адресу из Services.PublicationCrlUrl (по умолчанию http://localhost:9407), поэтому обычно его разворачивают на сервере панели. Параметры конфигурации описаны в подразделе «Сервис публикации» раздела Настройка вспомогательных сервисов (see page 139).

Что понадобится:

- сервер, подготовленный по «Подготовка сервера», с установленным Nginx (см. Общая подготовка сервера (see page 69));
- файл окружения /opt/itc/itc_svc_env.conf и юнит publication.service (см. «Юниты и окружение .NET-сервисов»);
- TLS-сертификат Nginx (nginx-tls.crt и nginx-tls.key в /opt/itc/itc/certs/) и серверный блок (см. «Общий серверный блок Nginx»);
- JWT и ключ сервиса: имя publication-service, роль crluser для получения CRL либо exporter для выгрузки сертификатов (см. «Выпуск JWT для сервиса»);
- из дистрибутива: архив ITC.Api.MiniCA.Publication.tar.gz и блок Nginx сервиса (nginx.conf).

Шаг 1. Создайте каталоги и распакуйте дистрибутив сервиса (от имени itc-svc):

```
mkdir -p /opt/itc/itc.minica.publication /opt/itc/itc/{config,nginx,certs} /opt/itc/
itc/logs/itc_api_miniCA_publication
```

```
tar -xzf /tmp/ITC.Api.MiniCA.Publication.tar.gz -C /opt/itc/itc.minica.publication/
```

Шаг 2. Разместите файлы для подключения к ЦС и расшифруйте ключ (см. «Учётные данные для подключения к ЦС»).

Шаг 3. Создайте файл конфигурации /opt/itc/itc/config/itc_api_miniCA_publication_config.json. Он содержит адрес OpenID, подключение к ЦС (блок MiniCa) и списки заданий публикации CRL и сертификатов. По умолчанию списки пусты.

```
{
  "OpenId": {
    "authority": "https://openid.example.local",
    "ignoreCertificateErrors": true,
    "jwtRoleClaim": null
  },
  "MiniCa": {
    "sub-ca": [
      {
        "Url": "https://ca.example.local:9443",
        "Auth": {
          "Tls": "/opt/itc/itc/certs/sub-ca.crt",
          "Sign": "/opt/itc/itc/certs/sub-ca.key",
          "Jwt": "/opt/itc/itc/certs/sub-ca.jwt"
        }
      }
    ]
  },
  "PublishingCrlTasks": [],
  "PublishingCertificationTasks": []
}
```

Задания могут содержать пароли учётных записей SCP и LDAP, поэтому ограничьте доступ к файлу:

```
chmod 600 /opt/itc/itc/config/itc_api_miniCA_publication_config.json
```

Шаг 4. Настройте Nginx. Создайте блок /opt/itc/itc/nginx/publication.conf (в дистрибутиве он называется nginx.conf); увеличенные тайм-ауты нужны для длительных операций публикации:

```
location /api/publication {
    proxy_pass http://localhost:9407/api;
    proxy_http_version 1.1;
    proxy_cache_bypass $http_upgrade;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Connection keep-alive;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

```
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_connect_timeout 600s;
    proxy_read_timeout 600s;
    proxy_send_timeout 600s;
    send_timeout 600s;
}
```

Если серверный блок default ещё не подключён, подключите его ссылкой (см. «Общий серверный блок Nginx», шаг 5). Затем проверьте конфигурацию и перезагрузите Nginx:

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

Шаг 5. Поместите юнит publication.service в /home/itc-svc/.config/systemd/user/ и запустите сервис:

```
systemctl --user daemon-reload && systemctl --user enable --now publication.service
```

5.2.9.1 Задания публикации

Задания удобнее создавать в Контрольной панели: «Задачи → Публикация CRL» и «Задачи → Публикация сертификатов». Добавлять, изменять и удалять задания может пользователь с правом rleMiniCAConfig-w, а запускать их вне расписания может пользователь с доступом к разделу «Задачи» (право rleMiniCACRL-w). Задание создаётся для ЦС, выбранного в панели, а после нажатия «Сохранить» панель записывает его в конфигурацию сервиса. Порядок работы описан в Руководстве пользователя, раздел «Задачи публикации».

Задания можно задать и в файле конфигурации, например при развертывании. Такие задания изменяйте в файле. Пример задания публикации полного CRL в каталог точки распространения CDP:

```
"PublishingCrlTasks": [
  {
    "Key": "subca-crl-cdp",
    "Description": "CRL выдающего ЦС в CDP",
    "MiniCa": "sub-ca",
    "IsDeltaCrl": false,
    "LocalCopy": {
      "Path": "/opt/itc/web/cdp/<CN>.crl"
    },
    "Format": "Der",
    "Cron": "0 * * * *"
  }
]
```

Параметр	Назначение
Key	Уникальное название задания
Description	Комментарий к заданию
MiniCa	Имя подключения к ЦС из блока MiniCa
IsDeltaCrl	false для полного CRL, true для Delta CRL
Format	Формат файла: DER (значение Der) или PEM
Cron	Расписание cron из пяти полей, например `0 * * * *` (каждый час); время считается в UTC
LocalCopy.Path	Путь к файлу на сервере публикации
Scp	Передача по SCP: Host, Port (по умолчанию 22), Username, Password, Path, Ess
Ldap	Запись в каталог LDAP: Host, Port (по умолчанию 389), Username, Password, BaseDn, Filter, AttributeName (например, certificateRevocationList), SecureSocketLayer (LDAPS), ConnectionTimeout (мс), Ess

В задании должен быть задан хотя бы один способ публикации: LocalCopy, Scp или Ldap. Флаг Ess означает, что в Password указана ссылка на секрет во внешнем хранилище секретов (EXC), а не сам пароль.

`<CN>` означает общее имя ЦС, в котором пробелы заменены на «_». Файл полного CRL называется `<CN>.crl`, файл Delta CRL называется `<CN>+.crl` (задание с `"IsDeltaCrl": true`): именно эти адреса ЦС записывает в выпускаемые сертификаты. Каталог /opt/itc/web/cdp служит каталогом списков отзыва на сервере точки распространения CDP: файлы из него доступны по адресу `http://<сервер CDP>/cdp/<CN>.crl` (см. подраздел «Точка распространения CRL и сертификатов (CDP)» раздела Настройка вспомогательных сервисов (see page 139)); локальное копирование в него работает, если сервис публикации развёрнут на сервере CDP.

Задания PublishingCertificationTasks публикуют сертификаты, выпущенные по заданному шаблону, в каталог LDAP (атрибут записи, например userCertificate). Их параметры: Key, Description, MiniCa, TemplateName (шаблон), CertificateLastId (ID последнего опубликованного сертификата), Chunk (размер порции), Format (по умолчанию PEM), Cron и блок Ldap.

После правки файла перезапустите сервис (команда перезапускает службу):

```
systemctl --user restart publication.service
```

Проверка.

- `systemctl --user status publication.service` показывает `active (running)`.
- В Контрольной панели на странице «Настройки → Конфигурация» у карточки «Конфигурация сервиса публикации CRL» зелёный индикатор и указана версия.
- В карточке задания кнопка «Выполнить» завершается уведомлением «Задача по публикации выполнена», а файл появляется по указанному пути.

Проверить опубликованный CRL можно командой openssl; поля lastUpdate и nextUpdate показывают время выпуска и срок действия списка:

```
openssl crl -inform DER -in /opt/itc/web/cdp/<CN>.crl -noout -issuer -lastupdate
-nextupdate
```

Сетевой доступ: сервис публикации → ЦС, 9443/tcp; → серверы SCP, 22/tcp; → каталог LDAP, 389/tcp (или порт LDAPS). Порт 9407 закройте для внешних подключений, кроме сервера Контрольной панели, если она на другом сервере.

5.2.10 Развертывание OCSP-ответчика (mOCSP)

Ответчик mOCSP (исполняемый файл mocsp) отвечает на OCSP-запросы о статусе сертификатов в реальном времени. В зависимости от параметра `status-source` сервис определяет статус сертификата по данным БД ЦС, по актуальному файлу CRL либо сначала обращается к БД и при ошибке запроса использует CRL. Ответы подписываются отдельным сертификатом подписи OCSP.

Сервис слушает порт 8888 и принимает запросы по пути `/ocsp`. Клиенты обращаются к нему напрямую, без Nginx: `http://<fqdn-ocsp>:8888/ocsp`. Параметры конфигурации и настройка источника статуса описаны в подразделе «OCSP-респондер mOCSP» раздела Настройка вспомогательных сервисов (see page 139).

Что понадобится:

- сервер, подготовленный по «Подготовка сервера», с каталогом /opt/itc/ocsp, принадлежащим itc-svc;
- сетевой доступ к БД ЦС (5432/tcp) для режимов db-only и db-first;
- сертификат ЦС, статус сертификатов которого проверяет ответчик: на сервере ЦС это /opt/itc/minica/ca/ca.crt;
- из дистрибутива: исполняемый файл mocsp, файлы mocsp.env, mocsp.yml и юнит mocsp.service.

Шаг 1. От имени itc-svc создайте каталог конфигурации:

```
mkdir -p /opt/itc/ocsp/conf
```

Разложите файлы сервиса по каталогам:

Файл	Куда поместить
mocsp	/opt/itc/ocsp/mocsp
mocsp.env	/opt/itc/ocsp/mocsp.env
mocsp.yml	/opt/itc/ocsp/conf/mocsp.yml
mocsp.service	/home/itc-svc/.config/systemd/user/mocsp.service
Сертификат ЦС	/opt/itc/ocsp/conf/ca.crt

Сделайте файл mocsp исполняемым:

```
chmod +x /opt/itc/ocsp/mocsp
```

Шаг 2. Создайте ключ и запрос на сертификат, которым ответчик будет подписывать ответы. Вместо `<fqdn-ocsp>` укажите имя сервера mOCSP.

```
openssl genrsa -out /opt/itc/ocsp/conf/mocsp.key 4096
chmod 600 /opt/itc/ocsp/conf/mocsp.key
openssl req -new \
  -key /opt/itc/ocsp/conf/mocsp.key \
  -out /opt/itc/ocsp/conf/mocsp.csr \
  -sha256 -nodes \
  -subj "/CN=<fqdn-ocsp>" \
  -addext "subjectAltName = DNS:<fqdn-ocsp>"
```

Шаг 3. Передайте mocsp.csr на сервер ЦС в каталог /tmp и выпустите сертификат по шаблону `ocsp` (не `tls`): шаблон задаёт сертификату назначение «Подпись OCSP» (OCSP Signing). На сервере ЦС от имени itc-svc выполните команду, затем верните сертификат в /opt/itc/ocsp/conf/mocsp.crt.

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ca \
```

```
  -csr /tmp/mocsp.csr \
  -out /tmp/mocsp.crt \
  -template ocsp
```

Шаг 4. Настройте переменные окружения сервиса в /opt/itc/ocsp/mocsp.env. Юнит подключает этот файл параметром EnvironmentFile.

```
MOCSP_CONF="/opt/itc/ocsp/conf/mocsp.yml"
MOCSP_OPTIONS="-log-level=info -log-format=tint"
```

MOCSP_CONF задаёт путь к файлу конфигурации. MOCSP_OPTIONS содержит параметры командной строки, с которыми юнит запускает mocsp. Ключ `-log-level` задаёт уровень журнала: для рабочей эксплуатации подходит info, уровень debug включает подробный отладочный журнал.

Шаг 5. Укажите параметры в файле конфигурации /opt/itc/ocsp/conf/mocsp.yml: источник сведений о статусе, пути к файлам ключа и сертификатов, параметры БД и (для режимов crl-only и db-first) CRL. В поставляемом файле замените заполнитель `$PATH` на /opt/itc/ocsp. Значение database.url в нём записано конструкцией шаблона вида `{{ PG_USERNAME }}`. Замените его целиком.

```
listen-port: 8888
url-path: /ocsp
key-file: /opt/itc/ocsp/conf/mocsp.key      # ключ подписи OCSP-ответов
cert-file: /opt/itc/ocsp/conf/mocsp.crt     # сертификат подписи OCSP-ответов
ca-file: /opt/itc/ocsp/conf/ca.crt          # сертификат ЦС
status-source: db-only                      # для crl-only и db-first настройте также
секцию crl (см. «OCSP-респондер mOCSP»)
database:
  type: postgres
  url: postgres://<пользователь>:<пароль>@<хост-БД>:5432/<база-ЦС>
  allow-without-tls: true
```

В database.url укажите учётную запись и БД ЦС (по умолчанию cwica и cwica). Специальные символы в пароле закодируйте для URL. Файл содержит пароль БД, поэтому ограничьте доступ к нему:

```
chmod 600 /opt/itc/ocsp/conf/mocsp.yml
```

Шаг 6. Запустите сервис:

```
systemctl --user daemon-reload && systemctl --user enable --now mocsp.service
```

Шаг 7. На межсетевом экране откройте входящий порт 8888/tcp для клиентов OCSP и исходящие подключения к БД ЦС (5432/tcp).

Проверка. Запросите статус любого выпущенного сертификата. Команду можно выполнить на сервере mOCSP или на любом сервере с OpenSSL, которому доступен порт 8888. В `-CAfile` укажите корневой сертификат ЦС; если ответчик обслуживает корневой ЦС, это /opt/itc/ocsp/conf/ca.crt.

```
openssl ocsp -issuer /opt/itc/ocsp/conf/ca.crt -cert <проверяемый>.crt \
  -url http://<fqdn-ocsp>:8888/ocsp -CAfile <корневой сертификат ЦС> -resp_text
-no_nonce
```

Ответчик работает, если в выводе есть строка `Response verify OK` и статус сертификата: `good`, `revoked` или `unknown`. Значения ошибок openssl приведены в Руководстве пользователя, раздел «Проверка статуса по протоколу OCSP».

Клиенты находят ответчик по адресу из расширения AIA проверяемого сертификата. Адрес ответчика попадает в выпускаемые сертификаты, только если параметр ca.aia ядра ЦС содержит элемент `OCSP;URI:http://<fqdn-ocsp>:8888/ocsp` (см. «Адрес OCSP-ответчика в AIA» в подразделе «Шаблоны, поля CDP/AIA и сроки действия» раздела Настройка ядра ЦС (see page 126)).

5.2.11 Развертывание Архиватора

Архиватор (.NET-сервис ITC.Api.MiniCA.Archiver, порт 9507) по расписанию (по умолчанию @daily, раз в сутки в 00:00 UTC) переносит из рабочей БД ЦС в отдельную архивную БД истекшие сертификаты и связанные с ними записи: запросы и события. Параметры конфигурации описаны в подразделе «Архиватор» раздела Настройка вспомогательных сервисов (see page 139).

Что понадобится:

- сервер, подготовленный по «Подготовка сервера»;
- файл окружения /opt/itc/itc_svc_env.conf и юнит archiver.service (см. «Юниты и окружение .NET-сервисов»);
- сервер PostgreSQL для архивной БД и файл схемы minica.up.sql из дистрибутива (тот же файл схемы, что и для БД ЦС, см. Развертывание базы данных ЦС (see page 71));
- сетевой доступ с сервера архиватора к рабочей и архивной БД (5432/tcp);
- из дистрибутива: архив ITC.Api.MiniCA.Archiver.tar.gz.

Шаг 1. Создайте архивную БД на сервере PostgreSQL. По умолчанию БД называется cwica_archive и принадлежит учётной записи cwica. Схема та же, что у рабочей БД. Первую команду выполняйте, только если учётной записи cwica на этом сервере ещё нет.

```
sudo -u postgres psql -c "CREATE USER \"cwica\" WITH PASSWORD '<ПАРОЛЬ>';"
sudo -u postgres psql -c "CREATE DATABASE cwica_archive OWNER \"cwica\";"
```

```
sudo -u postgres psql -c "GRANT CONNECT, CREATE ON DATABASE \"cwica_archive\" TO
\"cwica\";"
sudo -u postgres psql -d cwica_archive -c "GRANT USAGE, CREATE ON SCHEMA public TO
\"cwica\";"
sudo -u postgres psql -d cwica_archive -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL ON TABLES TO \"cwica\";"
sudo -u postgres psql -d cwica_archive -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL ON SEQUENCES TO \"cwica\";"
```

Шаг 2. Загрузите схему от имени владельца БД, чтобы таблицы принадлежали cwica. Скопируйте minica.up.sql на сервер PostgreSQL (например, в /tmp) и выполните:

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

Шаг 3. На сервере архиватора создайте каталоги и распакуйте дистрибутив сервиса ITC.Api.MiniCA.Archiver.tar.gz в /opt/itc/itc.minica.archiver/ (от имени itc-svc):

```
mkdir -p /opt/itc/itc.minica.archiver /opt/itc/itc/config /opt/itc/itc/logs/
itc_api_miniCA_archiver
tar -xzf /tmp/ITC.Api.MiniCA.Archiver.tar.gz -C /opt/itc/itc.minica.archiver/
```

Шаг 4. Создайте файл конфигурации /opt/itc/itc/config/itc_api_miniCA_archiver_config.json, указав свои данные вместо заполнителей:

```
{
  "Tasks": [
    {
      "Key": "ClearwayCA",
      "Cron": "@daily",
      "Source": "Host=<хост-рабочей-
БД>;Port=5432;Database=cwica;Username=cwica;Password=<пароль>",
      "Target": "Host=<хост-архивной-
БД>;Port=5432;Database=cwica_archive;Username=cwica;Password=<пароль>"
    }
  ]
}
```

В `Source` укажите подключение к рабочей БД ЦС, в `Target` укажите подключение к архивной БД (по умолчанию для обеих используется учётная запись cwica). Назначение и значения по умолчанию всех параметров задания приведены в подразделе «Архиватор» раздела Настройка вспомогательных сервисов (see page 139).

Файл содержит пароли двух БД, поэтому ограничьте доступ к нему:

```
chmod 600 /opt/itc/itc/config/itc_api_miniCA_archiver_config.json
```

Шаг 5. Поместите юнит archiver.service в /home/itc-svc/.config/systemd/user/ и запустите сервис:

```
systemctl --user daemon-reload && systemctl --user enable --now archiver.service
```

Проверка. `systemctl --user status archiver.service` показывает `active (running)`. В назначенное расписанием время в журнале службы (`journalctl --user -u archiver.service` или каталог /opt/itc/itc/logs/itc_api_miniCA_archiver) появляются записи «Начало архивации…».

Сетевой доступ: архиватор → рабочая и архивная БД, 5432/tcp. Порт 9507 закройте для внешних подключений.

5.2.12 Развертывание адаптера SCEP

Через адаптер SCEP (.NET-сервис itc.acm.scepAdapter.clearwayCa, порт 8440) выпускаются и обновляются сертификаты сетевых устройств и серверов по протоколу SCEP. Адаптер публикуется через Nginx (TLS 443, путь `/api/v1/scep`); клиенты обращаются к нему по внешнему адресу `https://<fqdn>/api/v1/scep`. Путь чувствителен к регистру букв.

Право на выпуск клиент подтверждает одноразовым паролем (challenge), вложенным в запрос. Пароль выдаётся по адресу `https://<fqdn>/api/v1/scep/challenge-password` пользователю, вошедшему через сервис OpenID: запрос выполняется с токеном доступа OpenID (аутентификация через OpenID, не Kerberos). Пароль подходит для одного выпуска и по умолчанию действует 60 минут; для повторного выпуска нужен новый пароль. Сами операции SCEP токена не требуют.

Сертификаты выпускаются по шаблону ЦС из параметра CaTemplate (по умолчанию tls). Локальное состояние адаптера, в том числе выданные пароли, хранится в SQLite (/opt/itc/itc/sqlite/scep.db), не в PostgreSQL. Параметры конфигурации описаны в подразделе «SCEP-адаптер» раздела Настройка вспомогательных сервисов (see page 139).

Что понадобится:

- сервер, подготовленный по «Подготовка сервера», с установленным Nginx;
- файл окружения /opt/itc/itc_svc_env.conf и юнит scepAdapter.service (см. «Юниты и окружение .NET-сервисов»);
- TLS-сертификат Nginx и серверный блок (см. «Общий серверный блок Nginx»);
- JWT и ключ адаптера: имя scep, роль центра регистрации causer (см. «Выпуск JWT для сервиса»);
- развёрнутый сервис OpenID;
- из дистрибутива: архив itc.acm.scepAdapter.clearwayCa.tar.gz и файл конфигурации адаптера.

Шаг 1. Выпустите сертификат адаптера (RA): им адаптер подписывает ответы клиентам. На сервере адаптера от имени itc-svc сформируйте ключ и запрос. Вместо `<fqdn>` укажите имя сервера адаптера.

```
mkdir -p /opt/itc/itc/certs
openssl req -new -keyout /opt/itc/itc/certs/scep.key -out /opt/itc/itc/certs/scep.csr \
  -sha256 -nodes -subj "/CN=<fqdn>" -addext "subjectAltName = DNS:<fqdn>" \
  -addext "extendedKeyUsage = serverAuth, clientAuth"
chmod 600 /opt/itc/itc/certs/scep.key
```

Передайте scep.csr на сервер ЦС в каталог /tmp, выпустите сертификат по шаблону tls и верните его в /opt/itc/itc/certs/scep.crt:

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ca -csr /tmp/scep.csr -out /tmp/scep.crt -template tls
```

Шаг 2. Подготовьте цепочку сертификатов ЦС /opt/itc/itc/certs/scep-chain.pem. Если сертификаты выпускает выдающий ЦС, скопируйте с его сервера файл цепочки /opt/itc/minica/ca/chain.pem (корневой и выдающий ЦС). Если сертификаты выпускает корневой ЦС, скопируйте его сертификат /opt/itc/minica/ca/ca.crt. Сертификат адаптера scep.crt в цепочку не входит.

Проверьте состав цепочки. В выводе должны быть сертификаты ЦС:

```
openssl crl2pkcs7 -nocrl -certfile /opt/itc/itc/certs/scep-chain.pem | openssl pkcs7
-print_certs -noout
```

Шаг 3. Выпустите JWT адаптера (имя scep, роль causer), разместите файлы для подключения к ЦС и расшифруйте ключ (см. «Выпуск JWT для сервиса» и «Учётные данные для подключения к ЦС»).

Шаг 4. Создайте каталоги и распакуйте дистрибутив сервиса (от имени itc-svc):

```
mkdir -p /opt/itc/itc/sqlite /opt/itc/itc.acm.scepAdapter.clearwayCa /opt/itc/itc/
{config,nginx} \
  /opt/itc/itc/logs/itc_acm_scepAdapter_clearwayCa
tar -xzf /tmp/itc.acm.scepAdapter.clearwayCa.tar.gz -C /opt/itc/
itc.acm.scepAdapter.clearwayCa/
```

Шаг 5. Создайте файл конфигурации /opt/itc/itc/config/itc_acm_scepAdapter_clearwayCa_config.json: провайдер SQLite, пути к сертификатам, аутентификация OpenID и подключение к ЦС.

```
{
  "Database": { "Provider": "sqlite", "ConnectionName": "DefaultConnection" },
  "ConnectionStrings": { "DefaultConnection": "Data Source=/opt/itc/itc/sqlite/scep.db"
},
  "CaChainPath": "/opt/itc/itc/certs/scep-chain.pem",
  "ScepCertPath": "/opt/itc/itc/certs/scep.crt",
  "ScepKeyPath": "/opt/itc/itc/certs/scep.key",
  "SupportedAlgorithms": [ "SHA256withRSA", "SHA384withECDSA" ],
  "SupportsRenewal": true,
  "AuthenticationConfig": {
    "AuthenticationType": "openId",
    "OpenId": { "Authority": "https://openid.example.local", "IgnoreCertificateErrors":
true }
  },
  "ClearwayCaClientConfig": {
    "CaTemplate": "tls",
    "CaCluster": "sub-ca",
    "MiniCa": {
      "sub-ca": [
        {
          "Url": "https://ca.example.local:9443",
          "Auth": {
            "Tls": "/opt/itc/itc/certs/sub-ca.crt",
            "Sign": "/opt/itc/itc/certs/sub-ca.key",
            "Jwt": "/opt/itc/itc/certs/sub-ca.jwt"
          }
        }
      ]
    }
  }
}
```

Параметр	Что указать
AuthenticationConfig.OpenId.Authority	Адрес сервиса OpenID (issuer)
ClearwayCaClientConfig.CaTemplate	Шаблон ЦС, по которому выпускаются сертификаты
ClearwayCaClientConfig.CaCluster	Имя подключения к ЦС; должно совпадать с ключом в блоке MiniCa
ClearwayCaClientConfig.MiniCa	Подключение к ЦС (см. «Учётные данные для подключения к ЦС»)

Шаг 6. Поместите юнит scepAdapter.service в /home/itc-svc/.config/systemd/user/ (см. «Юниты и окружение .NET-сервисов»). Юнит принимает HTTP-запросы на порту 8440 и использует окружение из itc_svc_env.conf, без переменных `KRB5_*`. Строка запуска:

```
ExecStart=/opt/itc/itc.acm.scepAdapter.clearwayCa/ITC.Api.Acm.Scep.Adapter.ClearwayCa
--urls="http://0.0.0.0:8440"
```

Запустите юнит адаптера:

```
systemctl --user daemon-reload && systemctl --user enable --now scepAdapter.service
```

Шаг 7. Настройте Nginx. Создайте блок /opt/itc/itc/nginx/scep.conf, проксирующий `/api/v1/scep` на `http://localhost:8440/api/v1/scep`:

```
location /api/v1/scep {
    proxy_pass http://localhost:8440/api/v1/scep;
    proxy_http_version 1.1;
    proxy_cache_bypass $http_upgrade;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection keep-alive;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

Если серверный блок default ещё не подключён, подключите его ссылкой (см. «Общий серверный блок Nginx», шаг 5). Затем проверьте конфигурацию и перезагрузите Nginx:

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

После этого адаптер доступен клиентам по адресу `https://<fqdn>/api/v1/scep`.

Проверка. Выполните запросы с любого сервера, которому доступен адрес адаптера. Вместо `<корневой сертификат ЦС>` укажите файл сертификата корневого ЦС, по которому проверяется HTTPS-соединение.

```
curl -sSf --cacert <корневой сертификат ЦС> "https://<fqdn>/api/v1/scep?
operation=GetCACaps"
curl -sSf --cacert <корневой сертификат ЦС> "https://<fqdn>/api/v1/scep?
operation=GetCACert" -o scep-ca.p7b
openssl pkcs7 -inform DER -in scep-ca.p7b -print_certs
curl -s -o /dev/null -w "%{http_code}\n" --cacert <корневой сертификат ЦС> "https://
<fqdn>/api/v1/scep/challenge-password"
```

Адаптер работает, если:

- GetCACaps возвращает список возможностей адаптера (например, POSTPKIOperation, SHA-256, Renewal);
- ответ GetCACert содержит сертификат адаптера (в subject указано имя сервера адаптера) и сертификаты ЦС;
- запрос одноразового пароля без токена возвращает код 401: адрес выдачи паролей доступен и требует входа через OpenID.

Порядок получения одноразового пароля и выпуска сертификата клиентом (например, службой certmonger) описан в Руководстве пользователя, раздел «SCEP».

Сетевой доступ: клиенты SCEP → сервер адаптера, 443/tcp; адаптер → ЦС, 9443/tcp; адаптер → сервис OpenID, 443/tcp. Порт 8440 закройте для внешних подключений.

5.3 Настройка ядра ЦС (minica.yml)

Ядро ЦС (minica) работает как сетевой сервис: выпускает и отзывает сертификаты, формирует списки отзыва (CRL) и ведёт реестры в базе данных. Служба ядра называется minica.service. Раздел описывает параметры ядра: сеть и TLS, ключ ЦС, расширения выпускаемых сертификатов, подключение к БД и сервисные роли.

Всё поведение ядра ЦС задаётся одним конфигурационным файлом:

```
/opt/itc/minica/conf/minica.yml
```

Путь к файлу задаёт переменная MINICA_CONF в файле окружения службы /opt/itc/minica/minica.env; по умолчанию это /opt/itc/minica/conf/minica.yml. Установщик создаёт файл из шаблона и подставляет в него вычисляемые значения: каталог установки, имя ЦС из субъекта сертификата, имя сервера точек распространения (CDP_SERVER_NAME) и параметры подключения к БД. Поэтому после установки большинство параметров уже заполнено.

В таблицах раздела пути указаны относительно каталога ЦС /opt/itc/minica. В файле они записаны полностью: например, `conf/tls.key` означает /opt/itc/minica/conf/tls.key.

5.3.1 Порядок изменения параметров

Все действия выполняются на сервере ЦС. Права на файл проверяет администратор с правом sudo, остальные шаги выполняются в сеансе технологической учётной записи itc-svc. Изменения вступают в силу только после перезапуска службы.

Шаг 1. Под учётной записью администратора с правом sudo проверьте владельца и права файла. Файл содержит пароль БД и пути к закрытым ключам ЦС, поэтому он должен принадлежать itc-svc и иметь права 0600.

```
sudo stat -c '%U:%G %a' /opt/itc/minica/conf/minica.yml      # ожидается: itc-svc:itc-
svc 600
```

```
sudo chown itc-svc:itc-svc /opt/itc/minica/conf/minica.yml   # если владелец другой
sudo chmod 600 /opt/itc/minica/conf/minica.yml               # если права шире 600
```

Шаг 2. Войдите в сеанс itc-svc командой `sudo su - itc-svc` и измените нужные параметры. Редактируйте файл от имени itc-svc, а не root. Не оставляйте резервные копии файла в общедоступных каталогах.

Шаг 3. Перезапустите службу командой `systemctl --user restart minica.service` (см. Управление службой (see page 138)). На время перезапуска ЦС не принимает запросы.

Шаг 4. Проверьте, что служба активна и ЦС отвечает на запросы (см. Управление службой (see page 138)). Если служба не запустилась, просмотрите сообщения службы командой `journalctl --user -u minica.service -n 50` и журнал /opt/itc/minica/logs/minica.log. Типовые причины описаны в главе «Диагностика и устранение неисправностей».



Защита файлов ЦС

Права на остальные файлы ЦС (ключи, токены, конфигурации других сервисов) описаны в разделе Ограничение прав доступа к файлам (see page 178). Общие меры защиты приведены в главе Обеспечение безопасной среды (see page 167).

5.3.2 Сеть и HTTP/HTTPS-сервер

Ядро ЦС принимает по HTTP или HTTPS запросы центров регистрации (компонентов, которые принимают заявки на сертификаты и передают их в ЦС), служебных компонентов Clearway CA и утилиты mclient. Рабочий порт: 9443 (HTTPS).

Параметр	Назначение	Значение по умолчанию
`listen-ip`	IP-адрес, на котором сервис принимает соединения. Пустая строка означает все интерфейсы	`""` (все интерфейсы)
`http-port`	Порт HTTP, используется при выключенном TLS	9080
`https-port`	Порт HTTPS, используется при включённом TLS	9443
`timeout-ms`	Тайм-аут обработки запроса, мс	60000

Чтобы ЦС принимал соединения только на одном адресе, укажите этот адрес в `listen-ip`.

Отдельного параметра с адресом сервиса (server-url) в minica.yml нет: адрес складывается из `listen-ip`, порта и признака TLS. Клиенты обращаются к ЦС по адресу вида `https://<fqdn>:9443`. Для утилиты mclient адрес ЦС задаётся параметром `server-url` в файле /opt/itc/minica/conf/mclient.yml.

5.3.3 TLS и взаимная аутентификация (mTLS)

TLS шифрует соединения клиентов с ЦС. Взаимная аутентификация (mTLS) дополнительно требует, чтобы клиент предъявил сертификат, которому доверяет ЦС.

В поставляемом шаблоне TLS выключен (`tls.enable: false`). Установщик при установке корневого и выдающего ЦС выпускает TLS-сертификат ЦС на имя узла (или на псевдоним, заданный при установке в поле «Alias сервера ЦС») и включает TLS. Поэтому после установки ЦС работает по HTTPS на порту 9443, а порт 9080 не используется.

При ручном развертывании TLS включают отдельным шагом (см. «Включение TLS» в разделе «Развертывание ядра ЦС (minica)»), до этого ЦС работает по HTTP на порту 9080. Взаимная аутентификация (`tls.verify`) после установки выключена, для доверенной эксплуатации её рекомендуется включить вручную.

Параметр	Назначение	Значение по умолчанию
`tls.enable`	Включение TLS: сервис переходит на порт `https-port`	`false` в шаблоне; `true` после установки
`tls.verify`	Включение mTLS (проверки клиентского сертификата)	`false`
`tls.key`	Файл закрытого ключа сервера	`conf/tls.key`
`tls.cert`	Файл сертификата сервера	`conf/tls.crt`
`tls.ca-cert`	Сертификат ЦС для проверки клиентских сертификатов (mTLS)	`conf/ca.crt`
`tls.ca-certs`	Дополнительные доверенные сертификаты (mTLS)	`[]`
`tls.insecure`	Не проверять сертификат ЦС (только для отладки)	`false`

Порядок включения mTLS (выпуск клиентского сертификата mclient, настройка секций `tls` в mclient.yml и minica.yml, проверка) и пример рабочей секции `tls` приведены в подразделе

«Включение взаимной аутентификации (mTLS)» главы Обеспечение безопасной среды (see page 167).

Перед включением `tls.verify` убедитесь, что у всех клиентов ЦС, включая mclient, есть клиентские сертификаты, которым ЦС доверяет по `tls.ca-cert` или `tls.ca-certs`. Иначе запросы этих клиентов будут отклонены. Обращайтесь к ЦС по имени, на которое выпущен TLS-сертификат ЦС.

> **Отладочный режим**
>
> Не оставляйте `tls.insecure: true` в промышленной эксплуатации: в этом режиме сертификаты не проверяются. Рекомендации по защите канала приведены в разделе «Защита соединений центра сертификации (TLS/mTLS)» главы Обеспечение безопасной среды (see page 167).

5.3.4 Контроль рассогласования времени (max-dt-sec)

Подпись каждого запроса и JWT проверяется с учётом метки времени. Параметр `max-dt-sec` задаёт, насколько время узла-отправителя может отличаться от времени ЦС.

Параметр	Назначение	Значение по умолчанию
`max-dt-sec`	Допустимое рассогласование времени при проверке подписи, с	600 (окно ±10 минут)

Если часы расходятся больше, чем на заданное окно, запрос отклоняется. Это защищает от повторной отправки ранее перехваченных запросов.

> **Синхронизация времени**
>
> Синхронизируйте время по NTP на **всех** узлах ЦС и на сервере СУБД. Иначе легитимные запросы будут отклоняться при выходе за окно ±10 минут. Настройка описана в разделе «Синхронизация времени и защита от повторов» главы Обеспечение безопасной среды (see page 167).

5.3.5 Ключ и сертификат ЦС

Параметры ключевой пары и сертификата ЦС, а также файлы и рабочие каталоги OpenSSL, через который ядро выполняет криптографические операции.

Параметр	Назначение	Значение по умолчанию

`ca.name`	Имя секции ЦС в файле openssl.cnf	`CA_default`
`ca.key`	Файл закрытого ключа ЦС	`ca/ca.key`
`ca.cert`	Файл сертификата ЦС	`ca/ca.crt`
`ca.old-keys`	Каталог ключей и сертификатов, которые ЦС использовал раньше (используется после ротации ключа ЦС)	`ca/old-keys`
`ca.new-keys`	Каталог, в котором ядро хранит новый ключ ЦС при «Перевыпуске» из Контрольной панели до установки сертификата	`ca/new-keys`
`ca.start-sn`	Стартовый серийный номер выпускаемых сертификатов. В бесплатной редакции не действует	`100000000000000000000`
`ca.start-crl`	Стартовый серийный номер CRL	`100000000000000000000`
`ca.rand-sn`	Использовать случайные серийные номера вместо порядковых	`false`
`chain-file`	Файл цепочки сертификатов ЦС (PEM)	`ca/chain.pem`
`openssl.cnf` (ключ `cnf` секции `openssl`)	Базовый конфигурационный файл OpenSSL	`conf/openssl.cnf`
`ca.outdir`	Каталог для временного сохранения выпускаемых сертификатов	`ca/certs`
`ca.crldir`	Каталог для временного сохранения CRL	`ca/crl`

Замена ключа и сертификата ЦС описана в разделе «Ротация ключа ЦС». Если ключ ЦС хранится в файле, сертификат ЦС можно заменить и из Контрольной панели: «Настройки → Ротация ключей».

Сертификаты ГОСТ

Перед выпуском сертификатов ГОСТ установите пакет поддержки ГОСТ на сервере ЦС. Для Astra Linux:

```
sudo apt install libgost-astra
```

5.3.5.1 Хранение ключа ЦС на аппаратном носителе (PKCS#11)

Платная редакция

Хранение ключа ЦС на аппаратном носителе не поддерживается в бесплатной редакции Clearway CA.

Ключ ЦС можно хранить на аппаратном носителе (HSM, Rutoken). Тогда ядро ЦС обращается к ключу через интерфейс PKCS#11, а ключ не покидает носитель. Для этого предусмотрена подсекция `pkcs11` секции `openssl` файла minica.yml: в поставляемом файле её нет, её добавляют вручную.

Параметр	Назначение	Значение по умолчанию
`openssl.pkcs11.enable`	Использовать ключ ЦС на носителе через PKCS#11	`false`
`openssl.pkcs11.ca-uri`	PKCS#11-URI ключа ЦС: строка, по которой модуль носителя находит ключ (метка токена, метка ключа, PIN)	пусто

В режиме PKCS#11 ядро ЦС берёт ключ с носителя по URI из `openssl.pkcs11.ca-uri`. Сертификат ЦС по-прежнему задаётся параметром `ca.cert`. Носитель с ключом ЦС подготовьте заранее средствами модуля (pkcs11-tool, утилиты производителя): при инициализации задаются метка токена и метка ключа.

Шаг 1. На сервере ЦС от имени itc-svc добавьте в существующую секцию `openssl` файла minica.yml подсекцию `pkcs11`. Остальные параметры секции не меняйте, вторую секцию `openssl` не создавайте.

```
openssl:
  cmd: openssl                              # существующие параметры секции не меняются
  cnf: /opt/itc/minica/conf/openssl.cnf
  sha-size: 256
```

```
  timeout-us: 0
  pkcs11:                                  # добавляемая подсекция
    enable: true
    ca-uri: "pkcs11:token=<метка_токена>;object=<метка_ключа>;type=private;pin-
value=<pin>"
```

Значения `token`, `object` и `pin-value` замените значениями своего носителя. URI содержит PIN носителя, поэтому права 0600 на minica.yml обязательны.

Шаг 2. На сервере ЦС от имени itc-svc активируйте движок PKCS#11 в файле /opt/itc/minica/conf/openssl.cnf. Раскомментируйте подключение секции движков и саму секцию `[engine_section]`; секция `[pkcs11_section]` в файле уже раскомментирована.

```
[ default_conf ]
ssl_conf = ssl_sect
engines = engine_section          # раскомментировать
[ engine_section ]
pkcs11 = pkcs11_section           # раскомментировать (RSA/ECDSA + Rutoken)
[ pkcs11_section ]
engine_id = pkcs11
dynamic_path = /usr/lib/x86_64-linux-gnu/engines-1.1/libpkcs11.so
MODULE_PATH  = /usr/lib/librtpkcs11ecp.so   # библиотека PKCS#11 (пример для Rutoken)
default_algorithms = ALL
```

Значения `dynamic_path` и `MODULE_PATH` приведены как пример: они зависят от модуля, ОС и версии OpenSSL.

Шаг 3. Перезапустите службу и проверьте, что ЦС отвечает на запросы (см. Управление службой (see page 138)).

Ротация из Контрольной панели для ключа на носителе недоступна: ядро ЦС возвращает ошибку. Ограничения при замене такого ключа описаны в разделе «Ротация ключа ЦС» (подраздел «Ограничения механизма old-keys»).

5.3.6 Шаблоны, поля CDP/AIA и сроки действия

ЦС записывает в выпускаемые сертификаты адреса, по которым клиенты проверяют их статус и строят цепочку:

- CDP (точка распространения CRL): адрес, откуда скачивается список отзыва;
- AIA (доступ к сведениям об издателе): адрес сертификата ЦС-издателя и, при настройке, адрес OCSP-ответчика;
- Freshest CRL: адрес разностного списка отзыва Delta CRL, который содержит только изменения с момента выпуска последнего полного CRL.

Установщик вычисляет эти адреса из имени ЦС и CDP_SERVER_NAME (FQDN сервера точек распространения, который указывается при установке ЦС). Если CDP_SERVER_NAME не задать,

списки адресов остаются пустыми. Параметры задаются списками YAML: в каждом можно указать несколько адресов.

Параметр	Назначение	Значение по умолчанию
`ca.exts`	Секция расширений openssl.cnf, по которой выпускаются сертификаты	`usr_cert`
`ca.exts-crl`	Секция расширений openssl.cnf, по которой выпускаются CRL	`crl_ext`
`ca.cdp`	Адреса CDP в сертификатах	`["http://<CDP_SERVER_NAME>/cdp/<CN>.crl"]`
`ca.aia`	Элементы расширения AIA в сертификатах	`["caIssuers;URI:http://<CDP_SERVER_NAME>/aia/<CN>.crt"]`
`ca.freshest-crl`	Адреса Delta CRL (суффикс `+` в имени файла)	Выдающий ЦС: `["http://<CDP_SERVER_NAME>/cdp/<CN>+.crl"]` Корневой ЦС: не задан

Здесь `<CN>` обозначает значение CN из субъекта сертификата ЦС, в котором пробелы заменены на «_». Например, для ЦС «Issuing CA» список отзыва называется Issuing_CA.crl. ЦС только записывает эти адреса в сертификаты и CRL. Сами файлы по этим адресам публикуются на сервере CDP (см. раздел «Точка распространения CRL и сертификатов (CDP)»).

Приоритет шаблона

Шаблон сертификата может переопределить адреса CDP/AIA и сроки действия из этого раздела, в том числе отключить расширения AIA и CDP. Значения из minica.yml применяются, когда шаблон не задаёт собственных. Шаблоны хранятся в БД, ядро ЦС кеширует их на время `database.templ-cache-minutes`.

5.3.6.1 Адрес OCSP-ответчика в AIA

Приложения находят OCSP-ответчик по адресу из расширения AIA сертификата. Этот адрес попадает в выпускаемые сертификаты, только если в `ca.aia` есть элемент `OCSP;URI:`. Проверьте значение `ca.aia`. Если такого элемента нет, добавьте его с адресом сервиса mOCSP (по умолчанию порт 8888, путь /ocsp):

```
ca:
  aia: ["caIssuers;URI:http://<CDP_SERVER_NAME>/aia/<CN>.crt", "OCSP;URI:http://<fqdn-ocsp>:8888/ocsp"]
```

Здесь `<fqdn-ocsp>` обозначает имя сервера mOCSP. После перезапуска службы адрес попадает только в новые сертификаты: у выпущенных ранее сертификатов AIA не меняется (см. подраздел «Учет OCSP URL в выпущенных сертификатах» раздела «Ротация ключа ЦС»).

Проверка. Выпустите тестовый сертификат и проверьте его расширение AIA. Команду можно выполнить на любом узле с OpenSSL; для файла в формате DER добавьте `-inform DER`:

```
openssl x509 -in <файл сертификата> -noout -text | grep -A 2 "Authority Information Access"
```

В выводе должны быть строки `OCSP - URI:` и `CA Issuers - URI:` с заданными адресами.

5.3.6.2 Сроки действия

Сроки действия сертификатов и CRL, которые ЦС применяет, если шаблон не задаёт собственных:

Параметр	Назначение	Значение по умолчанию
`ca.days.cert`	Срок действия выпускаемых сертификатов, дней. При значении 0 действует срок из конфигурации OpenSSL	1460 (≈4 года)
`ca.days.crl`	Срок действия CRL, дней. При значении 0 действует срок из конфигурации OpenSSL	365
`ca.days.delta-crl`	Срок действия Delta CRL, дней. При значении 0 действует срок CRL из конфигурации OpenSSL	0
`ca.overlap`	Смещение начала срока действия сертификата в прошлое, мин	10
`ca.overlap-crl`	Разница между nextUpdate и nextPublish для CRL, ч	0
`ca.no-limit-ca`	Не ограничивать срок действия выпускаемых сертификатов сроком сертификата ЦС	`false`

5.3.7 Шифрование данных ЦС (ca.encryption)

Конфиденциальные данные ЦС шифруются на **отдельной** ключевой паре, которая не совпадает с ключом подписи ЦС. Как включить шифрование данных сертификатов, описано в разделе «Шифрование данных сертификатов в БД» главы Обеспечение безопасной среды (see page 167).

Параметр	Назначение	Значение по умолчанию
`ca.encryption.key`	Закрытый ключ пары шифрования данных	`ca/encr.key`
`ca.encryption.cert`	Сертификат пары шифрования данных	`ca/encr.crt`



Отдельная пара ключей

Не указывайте здесь ключ подписи ЦС (`ca/ca.key`). Для шифрования используется самостоятельная пара encr.key и encr.crt.

5.3.8 Восстановление ключей субъектов (KRA)

Агент восстановления ключей (KRA) позволяет восстановить закрытый ключ субъекта, если тот его утратил. Резервная копия закрытого ключа субъекта шифруется на сертификате KRA, а расшифровать её может только владелец закрытого ключа KRA. Порядок работы описан в разделе «Механизм KRA (резервное копирование и восстановление ключей)» главы Обеспечение безопасной среды (see page 167).

Параметр	Назначение	Значение по умолчанию
`ca.kra.cert`	Сертификат агента восстановления ключей (KRA)	`ca/kra.crt`

5.3.9 База данных

Ядро ЦС хранит в PostgreSQL запросы, сертификаты, CRL, шаблоны, JWT и события. Поддерживаются версии PostgreSQL 11–18.

Параметр	Назначение	Значение по умолчанию
`database.type`	Тип СУБД	`postgres`

`database.url`	Строка подключения к БД ЦС: `postgres://<пользователь>:<пароль>@<сервер БД>:<порт>/<имя БД>`	Заполняет установщик; по умолчанию пользователь и БД `cwica`, порт 5432
`database.templ-cache-minutes`	Время хранения шаблонов в кеше, мин. Снижает число обращений к БД при выпуске сертификатов	3
`database.allow-without-tls`	Разрешить подключение к БД без TLS	`true`

Если пароль в `database.url` содержит символы со специальным значением в URL (например, `@`, `:`, `/`), запишите их в URL-кодировке: например, `@` записывается как `%40`.

Параметр `allow-without-tls` управляет **каналом к СУБД**, а не подключениями клиентов к самому ЦС (за них отвечает секция `tls`). Установщик задаёт значение `true`. Для доверенной эксплуатации установите `false`: тогда при отсутствии TLS на стороне БД сервис завершит работу и не откроет незащищённое соединение.

Чтобы ЦС проверял сертификат сервера БД, добавьте в строку подключения `sslmode=verify-full`, а параметром `sslrootcert` укажите сертификат ЦС, выпустившего сертификат сервера БД. Без `sslmode=verify-full` сервис доверяет любому TLS-сертификату сервера БД.

Порядок настройки и проверки, включая пример секции `database`, описан в разделе «Защита подключений к СУБД и каталогу LDAP» главы Обеспечение безопасной среды (see page 167), в подразделах «Подключение ядра ЦС и mOCSP к СУБД» (подключение ЦС) и «Сервер PostgreSQL» (TLS на стороне PostgreSQL).

5.3.10 Сервисные роли JWT (ролевая модель)

Компоненты Clearway CA и утилита mclient обращаются к API ЦС с JWT (токеном доступа, который выпускает сам ЦС). В токене указана роль компонента, и ЦС выполняет только те функции API, которые разрешены этой роли. Не путайте сервисные роли с правами пользователей Контрольной панели (см. главу Описание ролевой модели (see page 47)).

Роль объединяет под одним именем набор атомарных прав; одно атомарное право разрешает одну функцию API. Всего в API 27 функций. Роли задаются в секции `jwt.roles` файла minica.yml: имя роли и список прав через запятую.

Роль компонента записывается в его JWT при выпуске (параметр `-jwt-role` команды `mclient mkjwt`). Рекомендуемые роли: admin для Контрольной панели, crluser (получение CRL) или exporter (выгрузка сертификатов) для сервиса публикации, causer для SCEP-адаптера.

Установщик по умолчанию выпускает JWT с ролью admin; роль меняется в дополнительных настройках шага «Выпуск JWT» (см. «Выпуск JWT для сервиса»).

В поставке определено 7 ролей:

Роль	Назначение	Атомарные права
`admin`	Администратор: полный доступ ко всем функциям API	Все 27 функций: ping, ca, revoke, status, get, updcrl, crl, updeltacrl, deltacrl, find, delete, export, templates, csr, repair, stat, gettempl, addtempl, deltempl, mkjwt, revokejwt, repairjwt, deljwt, getjwt, renewjwt, revivejwt, chain
`causer`	Центр регистрации: выпуск, отзыв и проверка сертификатов	ping, ca, revoke, status, get, repair, csr, templates, gettempl, chain
`crluser`	Клиент CDP: обновление и получение CRL и Delta CRL	ping, updcrl, crl, updeltacrl, deltacrl, chain
`archuser`	Архивариус: удаление устаревших данных из БД	ping, status, get, crl, deltacrl, find, delete, export, templates, stat, gettempl, csr, getjwt
`exporter`	Экспортер: чтение любых данных из БД	ping, status, get, crl, deltacrl, find, export, templates, stat, gettempl, csr, getjwt
`templadmin`	Администратор шаблонов сертификатов	ping, templates, gettempl, addtempl, deltempl
`jwtadmin`	Администратор JWT	ping, mkjwt, revokejwt, repairjwt, getjwt, deljwt, revivejwt

Ключи и режим проверки токенов и подписи запросов:

Параметр	Назначение	Значение по умолчанию
`jwt.no-check`	Не проверять JWT клиента. Тогда подпись запросов проверяется только мастер-ключом	`false`
`jwt.key`	Файл ключа подписи JWT (PEM). Если файл не указан, ЦС не может выпускать JWT	`conf/jwt.key`
`jwt.keys`	Каталог ключей проверки подписи JWT. Имя файла: SKI (идентификатор ключа, вычисляемый по открытому ключу; 20 байт в HEX, верхний регистр) с расширением .pem	`conf/jwt-keys`

`jwt.cache-minutes`	Время хранения ключей проверки подписи запросов и JWT в кеше, мин	5
`master-key`	Мастер-ключ проверки подписи запросов (PEM): открытый ключ, HMAC-секрет, сертификат X.509 или закрытый ключ. При пустой строке ключи проверки берутся только из таблицы jwt БД	`conf/mclient.pem`
`sign-off`	Отключить проверку подписи запросов	`false`

Контроль полномочий

Не устанавливайте `jwt.no-check: true` и `sign-off: true` в промышленной эксплуатации: это отключает контроль полномочий и подлинности запросов.

5.3.11 Управление службой

Ядро ЦС работает как пользовательский юнит systemd технологической учётной записи itc-svc, а не как системная служба. Файл юнита: /home/itc-svc/.config/systemd/user/minica.service. Службой управляют командами `systemctl --user` из сеанса itc-svc. Чтобы служба запускалась при загрузке узла и работала после выхода из SSH-сеанса, для учётной записи включается режим linger.

Команды управления выполняются на сервере ЦС в сеансе itc-svc (вход из учётной записи администратора: `sudo su - itc-svc`):

```
systemctl --user daemon-reload              # только после изменения файла юнита
systemctl --user enable  minica.service     # включить автозапуск
systemctl --user restart minica.service     # применить изменения minica.yml; на время
перезапуска ЦС недоступен
systemctl --user status  minica.service     # состояние службы
```

Режим linger включает администратор с правами sudo командой `sudo loginctl enable-linger itc-svc`; как проверить режим linger и окружение `systemctl --user`, описано в подразделе «Условия работы пользовательских юнитов» раздела «Управление службами».

Проверка после перезапуска. В сеансе itc-svc убедитесь, что служба активна и ЦС отвечает на запросы:

```
systemctl --user is-active minica.service    # ожидается: active
cd /opt/itc/minica
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
```

```
./mclient ping                                  # успех: выводятся версия ЦС и SKI его
ключа
```

Если служба не запустилась, просмотрите её последние сообщения командой `journalctl --user -u minica.service -n 50` и журнал /opt/itc/minica/logs/minica.log. Типовые причины описаны в главе «Диагностика и устранение неисправностей». Управление остальными службами Clearway CA описано в разделе «Управление службами».

> **Смежные разделы**
>
> Параметры журналирования (секции `log` и `secLog`, контроль целостности записей) описаны в главе Журналирование и мониторинг (see page 256). Меры по защите файлов, ключей и каналов ЦС приведены в главе Обеспечение безопасной среды (see page 167). В этом разделе они не повторяются.

5.4 Настройка вспомогательных сервисов

Раздел служит справочником по настройке вспомогательных сервисов Clearway CA: где лежат их конфигурационные файлы, что означают параметры и какие значения действуют по умолчанию. Порядок первичного развертывания описан в разделе «Ручное развертывание сервисов (без инсталлятора)», меры защиты приведены в главе Обеспечение безопасной среды (see page 167).

Помимо ядра ЦС (сервис minica) в состав Clearway CA входят вспомогательные сервисы: сервис аутентификации OpenID, Контрольная панель, сервис публикации, OCSP-ответчик mOCSP, архиватор, сервис резервного копирования, SCEP-адаптер и точка распространения CRL и сертификатов (CDP). Сервисы взаимодействуют по HTTP и HTTPS и могут размещаться как на одном, так и на разных серверах.

Общие принципы для всех вспомогательных сервисов:

- Сервисы работают как пользовательские юниты systemd технологической учётной записи (по умолчанию itc-svc): ими управляют командами `systemctl --user`, для учётной записи включён режим linger.
- Контрольная панель, сервис публикации и SCEP-адаптер доступны клиентам через Nginx по HTTPS на порту 443: Nginx передаёт запросы на порт приложения.
- Точка распространения CDP работает как сайт Nginx на порту 80 и раздаёт по HTTP файлы CRL и сертификаты ЦС.
- Сервис OpenID (порт 443) и OCSP-ответчик mOCSP (порт 8888) принимают подключения клиентов напрямую, без Nginx.
- Каталог установки по умолчанию: /opt/itc. Каталоги сервисов: /opt/itc/minica (ядро ЦС), /opt/itc/openid (OpenID), /opt/itc/ocsp (mOCSP), /opt/itc/web (CDP).
- Пользователи и сервис-клиенты проходят аутентификацию в сервисе OpenID (OIDC); при обращении к ЦС сервисы предъявляют JWT.

Порты приложений .NET-сервисов (8407, 9407, 9507, 8440, а также порт сервиса резервного копирования, в примерах 8409) принимают подключения на всех сетевых интерфейсах сервера, если в параметре `--urls` задан адрес 0.0.0.0. Закройте их для внешних подключений, как описано в разделе «Ограничение сетевого доступа» главы Обеспечение безопасной среды (see page 167).

Сводка сервисов:

Сервис	Юнит systemd	Порт приложения	Как доступен клиентам	Конфигурационный файл
Сервис аутентификации OpenID	openid.service	443 (HTTPS), 5559 (служебный HTTP)	Напрямую, порт 443	/opt/itc/openid/config.yaml
Контрольная панель	controlPanel.service	8407	Через Nginx, HTTPS 443: веб-интерфейс и путь `/api/minica`	/opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json
Сервис публикации	publication.service	9407	Через Nginx, HTTPS 443, путь `/api/publication`	/opt/itc/itc/config/itc_api_miniCA_publication_config.json
OCSP-ответчик mOCSP	mocsp.service	8888, путь `/ocsp`	Напрямую, HTTP 8888	/opt/itc/ocsp/conf/mocsp.yml
Архиватор	archiver.service	9507	Клиентам не публикуется	/opt/itc/itc/config/itc_api_miniCA_archiver_config.json
Сервис резервного копирования	Служба сервиса резервного копирования	Задаётся параметром запуска `--urls`, в примерах 8409; askpass на 127.0.0.1:5225	Клиентам не публикуется, к нему обращается серверная часть Контрольной панели	/opt/itc/itc/config/itc_api_miniCA_backup_config.json
SCEP-адаптер	scepAdapter.service	8440	Через Nginx, HTTPS 443, путь `/api/v1/scep`	/opt/itc/itc/config/itc_acm_scepAdapter_clearwayCa_config.json
Точка распространения CDP	nginx (системная служба)	80	Напрямую, HTTP 80, пути `/cdp/` и `/aia/`	/opt/itc/web/nginx.conf

> **Значения по умолчанию**
>
> Все пути указаны для значений по умолчанию: каталог установки /opt/itc, учётная запись itc-svc. Если при установке заданы иные значения, скорректируйте пути и имя учётной записи во всех приведённых командах.

5.4.1 Конфигурационные файлы .NET-сервисов

Контрольная панель, сервис публикации, архиватор и сервис резервного копирования читают JSON-файлы из каталога `<ITCRoot>/itc/config`, по умолчанию /opt/itc/itc/config. Переменная ITCRoot задаётся в файле /opt/itc/itc_svc_env.conf, который подключают юниты сервисов; без неё сервисы не запускаются.

Файлы читаются по порядку. Значения из следующего файла переопределяют значения из предыдущих, и любой из файлов может отсутствовать.

Порядок	Файл	Что в нём
1	itc_api_miniCA_config.json	Общие параметры всех .NET-сервисов сервера
2	itc_api_miniCA_config.runtime.json	Переопределение общих параметров. В этом файле Контрольная панель сохраняет подключения ЦС, добавленные через интерфейс
3	`<сервис>_config.json`	Параметры сервиса, например itc_api_miniCA_controlPanel_config.json
4	`<сервис>_config.runtime.json`	Переопределение параметров сервиса
5	`<сервис>_config.<окружение>.json`	Параметры сервиса для заданного окружения .NET

Вместо `<сервис>` подставляется имя сервиса: itc_api_miniCA_controlPanel, itc_api_miniCA_publication, itc_api_miniCA_archiver или itc_api_miniCA_backup. Изменения конфигурации применяйте перезапуском службы сервиса. Файлы содержат пароли и пути к ключам, поэтому ограничьте доступ к ним (см. Ограничение прав доступа к файлам (see page 178)).

Управление сервисами. Команды выполняются на сервере сервиса в сеансе технологической учётной записи (из учётной записи администратора: `sudo su - itc-svc`):

```
su - itc-svc
systemctl --user enable --now <юнит>     # включить автозапуск и запустить
systemctl --user restart <юнит>          # перезапустить после правки конфигурации; на
время перезапуска сервис недоступен
systemctl --user status  <юнит>          # состояние
```

Условия работы пользовательских юнитов и остальные команды описаны в разделе «Управление службами».

5.4.2 Сервис аутентификации OpenID

Сервис OpenID служит провайдером OpenID Connect (OIDC) и построен на Dex. Он проверяет учётные записи пользователей и сервис-клиентов в каталоге LDAP или Active Directory, сопоставляет группы каталога с правами Clearway CA и выдаёт клиентам токены доступа и обновления (JWT). Состояние сервиса (запросы авторизации, сессии, ключи подписи токенов) хранится в БД PostgreSQL.

Конфигурационный файл: /opt/itc/openid/config.yaml. Исполняемый файл: /opt/itc/openid/openid. Юнит openid.service запускает сервис командой `openid serve /opt/itc/openid/config.yaml`.

Порт 443 привилегированный, поэтому исполняемому файлу выдаётся право на привязку к нему командой `sudo setcap 'cap_net_bind_service=+ep' /opt/itc/openid/openid`. Право теряется при замене файла, например при обновлении, и команду повторяют.

Ключевые параметры:

Параметр	Назначение	Значение по умолчанию
`issuer`	Адрес издателя токенов (claim iss). Должен быть доступен браузерам пользователей и сервисам Clearway CA	`https://<имя сервера OpenID>`
`storage.type`, `storage.config`	Хранилище состояния: сервер, порт, БД, пользователь и пароль PostgreSQL	postgres; порт 5432, БД openid, пользователь cwi
`storage.config.ssl.mode`	Режим TLS при подключении к БД	disable
`web.https`	Адрес и порт приёма HTTPS-запросов. Сервис слушает порт на адресе, в который разрешается указанное имя	`<имя сервера OpenID>:443`

`web.tlsCert`, `web.tlsKey`	TLS-сертификат и ключ сервиса	/opt/itc/itc/certs/openid-tls.crt, /opt/itc/itc/certs/openid-tls.key
`web.allowedOrigins`	Адреса веб-приложений, которым разрешено обращаться к сервису из браузера	`['*']` (любые адреса)
`adminServer.http`	Служебный (административный) HTTP-интерфейс	`<имя сервера OpenID>:5559`
`enablePasswordDB`, `staticPasswords`	Вход встроенными учётными записями без каталога	true; встроенная учётная запись admin
`connectors[].config`	Коннектор к каталогу LDAP/AD: host (`<сервер>:<порт>`), bindDN, bindPW, insecureNoSSL, периодический импорт import	insecureNoSSL: true; import: active true, timer 1h, объекты person и groups
`connectors[].config.userSearch`, `groupSearch`	Правила поиска пользователей и групп: baseDN, filter, атрибуты, поиск при входе (onDemand)	baseDN ou=Users и ou=Groups; фильтры (objectClass=person) и (objectClass=group); атрибуты sAMAccountName, userPrincipalName, displayName, cn; onDemand true
`connectors[].config.ldapGroupRolesMap`	Сопоставление групп каталога правам Clearway CA	Группы CWCA_Administrator, CWCA_Auditor, CWCA_User
`expiry.idTokens`	Время жизни токена доступа	2m
`expiry.authRequests`	Время на прохождение авторизации	10m
`expiry.signingKeysAlgorithm`	Алгоритм подписи токенов: RS256, ES256, ES384, ES512 или EdDSA	RS256
`expiry.signingKeys`	Период ротации ключей подписи токенов	6h

`expiry.refreshTokens`	Токены обновления: validIfNotUsedFor задаёт срок действия при неиспользовании, sessionLimit задаёт число одновременных сессий, sessionLimitPolicy принимает значение reject (отклонять новую сессию) или terminate_oldest (завершать самую старую), absoluteLifetime задаёт предельный срок	3m, 1, reject, 24h
`staticClients[]`	Клиенты OIDC, которым сервис выдаёт токены: name, id, secret, redirectURIs	Клиент clearway с id itc-clearway; redirectURIs содержит адрес Контрольной панели

Пример файла приведён в шаге 6 раздела «Развертывание сервиса аутентификации OpenID». Настройки, которые нужно изменить для безопасной эксплуатации, описаны в главе Обеспечение безопасной среды (see page 167):

- подключение к БД и каталогу по TLS: подраздел «Остальные подключения»;
- ограничение `web.allowedOrigins`: подраздел «Параметры клиента OpenID Контрольной панели»;
- отключение встроенной учётной записи: подраздел «Встроенная учётная запись OpenID».

> **Сопоставление групп и прав**
>
> Карта `ldapGroupRolesMap` задаёт только соответствие «группа каталога → набор прав». Состав прав и стандартных ролей приведён в главе Описание ролевой модели (see page 47), а порядок назначения ролей описан в разделе Назначение ролей через группы AD/LDAP (see page 196). Изменённая карта применяется после перезапуска openid.service. Права передаются в токене при входе, поэтому пользователю нужно выйти из Контрольной панели и войти снова.

Управление службой (на время перезапуска вход в Контрольную панель недоступен):

```
su - itc-svc
systemctl --user restart openid.service
systemctl --user status  openid.service
```

5.4.3 Контрольная панель

Контрольная панель служит веб-интерфейсом управления центрами сертификации. В ней просматривают, выпускают и отзывают сертификаты, управляют шаблонами, запросами и списками отзыва, настраивают задачи публикации, выполняют резервное копирование и восстановление ЦС, ротацию ключей ЦС и типовые задачи замены сертификатов.

Панель состоит из серверной части ITC.Api.MiniCA.ControlPanel (.NET, порт 8407) и одностраничного веб-приложения (SPA). Nginx раздаёт SPA и файлы настроек браузера, а запросы `/api/minica` передаёт серверной части. Пользователи входят в панель через сервис OpenID. К ЦС серверная часть подключается по HTTPS (порт 9443) и авторизуется JWT панели. К сервису публикации она подключается по адресу из параметра `Services.PublicationCrlUrl`, к сервису резервного копирования по адресу из параметра `Services.BackupUrl`.

Конфигурационные файлы:

Файл	Назначение
/opt/itc/itc/config/ itc_api_miniCA_controlPanel_config.json	Конфигурация серверной части: OpenID, адреса сервисов, включаемые функции, подключения ЦС, созданные установщиком или из командной строки
/opt/itc/itc/config/ itc_api_miniCA_config.runtime.json	Подключения ЦС, добавленные и изменённые в интерфейсе («Настройки → Конфигурация»)
/opt/itc/itc/certs/ `<имя ЦС>`.crt, .key, .jwt	TLS-сертификат ЦС, ключ подписи запросов и JWT панели для подключения к ЦС, созданного установщиком или из командной строки
/opt/itc/itc/env/client_env.js	Параметры OpenID для веб-приложения: issuer, clientId, secret, rolesClaim, useSilentRefresh, scope
/opt/itc/itc/env/config.json	Настройки веб-интерфейса; установщик создаёт файл из spa.config.json дистрибутива
/opt/itc/itc/env/brand.svg, logo.svg, favicon.ico	Элементы оформления интерфейса
/opt/itc/itc/nginx/controlPanel.conf	Блоки location Nginx: раздача SPA из /opt/itc/ itc.minica.spa, каталога /opt/itc/itc/env по пути `/assets/env` и файла /favicon.ico; передача `/api/minica` на `http://localhost:8407/api`

Каталог /opt/itc/itc/env доступен всем

Nginx отдаёт любой файл каталога /opt/itc/itc/env по адресу `https://<fqdn>/assets/env/<имя файла>` без входа в панель. Не размещайте в этом каталоге секреты и оставляйте параметр `secret` в client_env.js пустым (`null`), см. подраздел «Параметры клиента OpenID Контрольной панели» главы Обеспечение безопасной среды (see page 167).

5.4.3.1 Параметры серверной части

Параметры задаются в файле /opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json. Пример файла приведён в шаге 2 раздела «Развертывание Контрольной панели».

Параметр	Назначение	Значение по умолчанию
`OpenId.Authority`	Адрес сервиса OpenID (issuer), у которого серверная часть проверяет токены пользователей	Адрес OpenID, указанный при установке
`OpenId.IgnoreCertificateErrors`	Не проверять TLS-сертификат сервиса OpenID	true
`OpenId.JwtRoleClaim`	Имя claim с правами пользователя в токене; при null используется claim roles	null
`Services.OpenId`	Адрес сервиса OpenID; совпадает с `OpenId.Authority`	Как `OpenId.Authority`
`Services.PublicationCrlUrl`	Адрес сервиса публикации	`http://localhost:9407`
`Services.BackupUrl`	Адрес сервиса резервного копирования. Серверная часть передаёт по нему запросы раздела «Задачи → Резервное копирование» вместе с токеном пользователя. Пока адрес не задан, раздел не работает	Не задан; например `http://localhost:8409`, если сервис работает на том же сервере
`Features.TypicalTask`	Включает типовые задачи замены сертификатов Nginx и PostgreSQL по SSH (пункт меню «Задачи → Типовые»). Если параметра нет, функция выключена	true
`ReportMaxSize`	Наибольшее число строк при выгрузке реестров в файл Excel	10000
`MiniCa.<имя ЦС>`	Подключение к ЦС: массив узлов ЦС с адресом `Url` (`https://<fqdn ЦС>:9443`) и учётными данными `Auth.Tls`, `Auth.Sign`, `Auth.Jwt`. Параметр `Ess` означает, что ключ подписи берётся из внешнего хранилища секретов (EXC)	Добавляется при подключении ЦС

Если сервис публикации развёрнут на другом сервере, укажите в `Services.PublicationCrlUrl` адрес `http://<сервер публикации>:9407` и разрешите подключение к порту 9407 только с сервера панели. Так же задаётся `Services.BackupUrl`: для сервиса резервного копирования на другом сервере укажите адрес `http://<сервер сервиса>:<порт>` и разрешите подключение к этому порту только с сервера панели (см. «Сервис резервного копирования»).

Параметр `ReportMaxSize` задаётся на верхнем уровне файла, рядом с секциями OpenId и Services. Установщик его не записывает, поэтому действует значение по умолчанию. Ограничение относится к выгрузке реестров «Сертификаты», «Запросы», «События» и «Шаблоны»: если под условия отбора попадает больше записей, в файл выгружаются только первые `ReportMaxSize` строк.

Как включить типовые задачи и что для них подготовить, описано в разделе «Типовые задачи замены сертификатов». Получение ключа подписи из EXC описано в разделе «Хранение секретов во внешнем хранилище (EXC)», проверка сертификата OpenID (`IgnoreCertificateErrors: false`) описана в подразделе «Остальные подключения» главы Обеспечение безопасной среды (see page 167).

5.4.3.2 Подключение ЦС

Каждое подключение к ЦС описывается отдельным блоком `MiniCa.<имя ЦС>`. Установщик создаёт подключение на шаге настройки Контрольной панели. Администратор может добавить или изменить подключение двумя способами: в интерфейсе или из командной строки. Порядок действий описан в подразделе «Подключение ЦС к панели» раздела «Развертывание Контрольной панели».

В интерфейсе подключение добавляют на странице «Настройки → Конфигурация → Действия → Добавить новый ЦС» (право `rleMiniCAConfig-w`). Для этого нужны наименование подключения, адрес ЦС, пароль ключа и три файла: TLS-сертификат ЦС, зашифрованный закрытый ключ и JWT панели. Подключения из интерфейса панель сохраняет в файле itc_api_miniCA_config.runtime.json. Если ЦС с тем же наименованием описан и в itc_api_miniCA_controlPanel_config.json, действуют значения из него: этот файл читается позже (см. «Конфигурационные файлы .NET-сервисов»).

Чтобы изменить подключение, выберите ЦС и нажмите «Действия → Обновить конфигурацию». Наименование подключения не меняется. Заново выберите все три файла и введите пароль ключа: в форму они не подставляются.

После сохранения ЦС появляется в карточке «Выбор Clearway CA» у всех пользователей. Пользователь выбирает текущий ЦС на странице «Настройки → Конфигурация» (право `rleMiniCAConfig-r`), и все операции панели, включая ротацию ключей и резервное копирование, выполняются для выбранного ЦС. Выбор хранится в браузере пользователя; если ЦС ещё не выбирался или выбранного подключения больше нет, используется первый ЦС списка.

5.4.3.3 Параметры веб-интерфейса

Параметры OpenID для веб-приложения задаются в файле /opt/itc/itc/env/client_env.js; назначение параметров описано в шаге 3 раздела «Развертывание Контрольной панели». Если параметр в файле не задан, для `scope` действует значение openid profile email offline_access, для `useSilentRefresh` значение true, для `rolesClaim` значение roles. Установщик записывает `useSilentRefresh: false`.

Панель выполняет вход по потоку authorization code и возвращает пользователя на свой адрес, например `https://cp.example.local`, поэтому этот адрес указывается в redirectURIs клиента в OpenID. Права пользователя интерфейс читает из ID-токена, из claim, указанного в `rolesClaim`. Если используется другой провайдер OIDC, он должен передавать права rleMiniCA* в ID-токене под этим именем.

Поведение интерфейса задаётся в файле /opt/itc/itc/env/config.json. В таблице приведены значения, которые записывает установщик.

Параметр	Назначение	Значение по умолчанию
`listTable__submitSingleFiltersOnChange`	Применять фильтр по столбцу сразу после изменения значения	true
`listTable__isSingleLineMode`	Однострочный режим таблиц: длинное значение ячейки выводится в одну строку и обрезается. Пользователь может изменить режим в окне «Настройки таблицы»	true
`listTable__isSingleColumnSort`	Сортировка только по одному столбцу: выбор сортировки по столбцу сбрасывает сортировку по остальным	true
`layoutPanel__panelEnabled`	Показывать панель с кнопками скрытия и показа меню и фильтров	true
`auth__loginInPopup`	Действие, если не удалось обновить токен: true означает повторный вход во всплывающем окне, false означает выход и переход на страницу входа	false
`auth__modal_is_success`	Показывать после входа окно «Успешная авторизация» с предупреждением	true
`auth__lastSession`	Выводить в этом окне дату последнего входа пользователя	true

`crl__download__fileName`	Имя файла, под которым браузер сохраняет CRL по кнопке «Скачать CRL»	crl.crl
`deltaCrl__download__fileName`	Имя файла, под которым браузер сохраняет Delta CRL по кнопке «Скачать Delta CRL»	delta-crl.crl

Если параметра в файле нет, соответствующая настройка выключена, а файлы CRL сохраняются под именами cwica-crl.crl и cwica-delta-crl.crl. Требования к окну с предупреждением описаны в подразделе «Предупреждение при входе» главы Обеспечение безопасной среды (see page 167).

Веб-интерфейс читает config.json и client_env.js при загрузке страницы. Изменения действуют после повторной загрузки страницы в браузере, перезапуск служб не нужен.

Управление службой (на время перезапуска панель недоступна):

```
su - itc-svc
systemctl --user restart controlPanel.service
sudo nginx -t && sudo systemctl reload nginx     # после изменения конфигурации Nginx;
выполняется администратором с правом sudo
```

Состояние и версии ЦС, серверной части панели и сервиса публикации показывает страница «Настройки → Конфигурация». Типовые неисправности описаны в главе «Диагностика и устранение неисправностей».

5.4.4 Сервис публикации

Сервис публикации (ITC.Api.MiniCA.Publication, .NET) по расписанию cron публикует списки отзыва (полные CRL и Delta CRL) в локальный каталог (например, в каталог точки распространения CDP), на удалённый сервер по SCP или в каталог LDAP. Выпущенные сертификаты он публикует в каталог LDAP (Active Directory или Samba). Сервис публикуется через Nginx (путь `/api/publication`), входящие запросы авторизуются через OpenID/JWT.

Конфигурационный файл: /opt/itc/itc/config/itc_api_miniCA_publication_config.json. Юнит: publication.service (порт 9407). Блок Nginx в файле /opt/itc/itc/nginx/publication.conf передаёт `/api/publication` на `http://localhost:9407/api` с тайм-аутами 600 с.

Ключевые параметры:

Параметр	Назначение	Значение по умолчанию
`OpenId.authority`	Адрес сервиса OpenID (issuer)	Адрес OpenID, указанный при установке

`OpenId.ignoreCertificateErrors`	Не проверять TLS-сертификат сервиса OpenID	true
`OpenId.jwtRoleClaim`	Имя claim с правами; при null используется claim roles	null
`MiniCa.<имя ЦС>`	Подключение к ЦС, из которого сервис получает CRL и сертификаты. Формат такой же, как у Контрольной панели	Задаётся при развертывании
`PublishingCrlTasks`	Задания публикации CRL и Delta CRL	`[]`
`PublishingCertificationTasks`	Задания публикации сертификатов	`[]`

После установки списки заданий пусты. Задания создают в Контрольной панели, в разделах «Задачи → Публикация CRL» и «Задачи → Публикация сертификатов». Добавлять, изменять и удалять задания может пользователь с правом `rleMiniCAConfig-w`, а запускать вне расписания может пользователь с правом `rleMiniCACRL-w`. Задание создаётся для ЦС, выбранного в панели. Время в расписании cron задания считается в UTC.

Задания можно задать и в файле конфигурации, например при развертывании; такие задания изменяют в файле, после чего перезапускают сервис. Параметры заданий и пример описаны в подразделе «Задания публикации» раздела «Развертывание сервиса публикации».

Проверка работоспособности сервиса по HTTP (`/api/health`) описана в разделе «Мониторинг и самодиагностика» главы Журналирование и мониторинг (see page 256).

Управление службой:

```
su - itc-svc
systemctl --user restart publication.service
```

5.4.5 OCSP-респондер mOCSP

OCSP-ответчик (респондер) mOCSP отвечает на запросы о статусе сертификатов по протоколу OCSP (RFC 6960). Он принимает OCSP-запросы по HTTP (методы GET и POST), определяет статус сертификата в момент запроса по выбранному источнику и возвращает подписанный ответ: good, revoked или unknown. Источником служит БД ЦС, актуальный файл CRL либо БД с переходом на CRL при ошибке запроса к БД. Ответы подписываются отдельным сертификатом подписи OCSP.

Конфигурационный файл: /opt/itc/ocsp/conf/mocsp.yml. Юнит mocsp.service запускает /opt/itc/ocsp/mocsp с параметрами из файла окружения /opt/itc/ocsp/mocsp.env: MOCSP_CONF задаёт путь к конфигурационному файлу, MOCSP_OPTIONS задаёт параметры командной строки, например уровень журнала (см. шаг 4 раздела «Развертывание OCSP-ответчика (mOCSP)»).

Клиенты обращаются к ответчику напрямую, без Nginx, по адресу `http://<fqdn>:8888/ocsp`; порт 8888/tcp открывают на межсетевом экране для клиентов OCSP. Адрес ответчика попадает в выпускаемые сертификаты, если он указан в параметре `ca.aia` ядра ЦС (см. «Адрес OCSP-ответчика в AIA» в разделе Настройка ядра ЦС (see page 126)).

Сеть и подпись ответов:

Параметр	Назначение	Значение по умолчанию
`listen-ip`	Адрес прослушивания; пустая строка означает все интерфейсы	`""`
`listen-port`	Порт OCSP-сервера	8888
`url-path`	Базовый путь URL ответчика	/ocsp
`timeout-ms`	Тайм-аут обработки запроса, мс	60000
`key-file`	Закрытый ключ подписи OCSP-ответов	/opt/itc/ocsp/conf/mocsp.key
`cert-file`	Сертификат подписи OCSP-ответов	/opt/itc/ocsp/conf/mocsp.crt
`ca-file`	Сертификат ЦС, статус сертификатов которого проверяет ответчик	/opt/itc/ocsp/conf/ca.crt
`hash`	Алгоритм хеширования	SHA256
`add-cert`	Включать сертификат подписанта в ответ	true
`nmin`	Интервал для поля NextUpdate ответа, мин	60
`update-expired`	Обновлять в БД статус сертификатов с истекшим сроком действия	true

`no-trim-zeros`	Не удалять незначащие нули серийного номера при поиске сертификата в БД	false

База данных и TLS:

Параметр	Назначение	Значение по умолчанию
`database.type`	Тип СУБД. Подключение к БД ЦС используется в режимах db-only и db-first	postgres
`database.url`	Строка подключения к БД ЦС: `postgres://<пользователь>:<пароль>@<сервер БД>:<порт>/<имя БД>`	Заполняет установщик; по умолчанию пользователь и БД cwica, порт 5432
`database.allow-without-tls`	Разрешить подключение к БД без TLS	true
`tls.enable` / `tls.verify`	TLS и взаимная аутентификация (mTLS) самого ответчика	false / false
`tls.key` / `tls.cert`	Ключ и сертификат TLS-сервера	/opt/itc/ocsp/conf/tls.key / /opt/itc/ocsp/conf/tls.crt
`tls.ca-cert`, `tls.ca-certs`	Сертификат ЦС и дополнительные доверенные сертификаты для проверки клиентов (mTLS)	/opt/itc/ocsp/conf/ca.crt, `[]`
`tls.insecure`	Не проверять сертификат ЦС (только для отладки)	false

Параметры журнала ответчика (секция `log`) описаны в разделе «Параметры журналирования (секции log и rotate)», а защищённое подключение к БД описано в подразделе «Подключение ядра ЦС и mOCSP к СУБД» главы Обеспечение безопасной среды (see page 167).

Сертификат подписи OCSP-ответов

Сертификат подписи mocsp.crt выпускается ЦС по шаблону ocsp: назначение «Подпись OCSP» (OCSP Signing) и расширение OCSP No Check. Установщик размещает ключ, запрос и сертификат в каталоге /opt/itc/ocsp/conf под именами mocsp.key, mocsp.csr и mocsp.crt. Без

этого сертификата сервис не может подписывать ответы. Один процесс mOCSP обслуживает один ЦС: тот, чей сертификат указан в `ca-file` (см. «Как mOCSP обслуживает прежний и новый ЦС» в разделе «Ротация ключа ЦС»).

5.4.5.1 Выбор источника сведений о статусе сертификата

Параметр `status-source` определяет источник сведений, на основании которого mOCSP формирует статусы good, revoked и unknown. Если параметр в файле не задан, действует режим db-only.

Значение	Назначение	Использование БД	Использование CRL
db-only	Получение статуса только из БД ЦС. Режим по умолчанию	Всегда	Не используется
crl-only	Постоянное получение статуса из актуального CRL	Не используется	Всегда
db-first	Получение статуса из БД с переходом на CRL при технической ошибке запроса к БД	Основной источник	Резервный источник

Значение указывается строчными буквами точно в приведённом виде. Неизвестное или ошибочно написанное значение сервис интерпретирует как db-only.

В режиме db-first переход на CRL выполняется, если запрос сертификата к БД завершился ошибкой. Если БД доступна, но сертификат в ней не найден, сервис возвращает статус unknown и к CRL не обращается.

Параметры CRL (секция `crl`, для режимов crl-only и db-first):

Параметр	Назначение	Значение по умолчанию
`crl.crl-path`	Точный путь к основному файлу CRL	Не задан; обязателен для db-first и crl-only
`crl.delta-crl-path`	Путь к необязательному Delta CRL	Не задан
`crl.reload-enabled`	Автоматически перезагружать CRL после изменения файла	true
`crl.reload-timeout`	Наибольшая продолжительность одной попытки перезагрузки	30s

`crl.time-skew`	Допуск рассогласования времени при проверке периода действия CRL	5s
`crl.fail-on-expired`	Запрещать использование просроченного CRL	true
`crl.source.watch.method`	Способ обнаружения изменения CRL: inotify или poll	inotify
`crl.source.watch.poll-interval`	Интервал проверки файла при методе poll	30s
`crl.source.watch.debounce`	Задержка перед перезагрузкой после файлового события inotify	300ms

5.4.5.2 Подготовка и настройка CRL

Подписанный центром сертификации список отозванных сертификатов называется CRL (Certificate Revocation List).

Администратор доставляет актуальный CRL обслуживаемого ЦС в локальный файл, указанный в параметре `crl.crl-path`, и регулярно его обновляет. Сервис mOCSP не загружает CRL из точки распространения, а читает указанный локальный файл. Файл должен быть доступен для чтения технологической учётной записи.

mOCSP принимает CRL в одном из форматов:

- DER;
- PEM с типом блока X509 CRL.

Сертификат ЦС, указанный в параметре `ca-file`, используется для проверки издателя и подписи CRL. Файл CRL должен содержать поля `ThisUpdate` и `NextUpdate`. При обработке каждого OCSP-запроса сервис проверяет время вступления CRL в действие и срок его действия с учётом допуска `time-skew`.

CRL, который ещё не вступил в действие, отклоняется. При `fail-on-expired: true` просроченный CRL также отклоняется; при `fail-on-expired: false` его использование допускается.

Пример для постоянного использования CRL:

```
status-source: crl-only

ca-file: /opt/itc/ocsp/conf/ca.crt
```

```
crl:
  # Рекомендуемый локальный путь; разместите здесь CRL обслуживаемого ЦС.
  crl-path: /opt/itc/ocsp/conf/ca.crl

  # Оставьте пустым, если Delta CRL не используется.
  delta-crl-path: ""

  reload-enabled: true
  reload-timeout: 30s

  time-skew: 5s
  fail-on-expired: true

  source:
    watch:
      method: inotify
      poll-interval: 30s
      debounce: 300ms
```

Для резервного использования CRL при ошибке обращения к БД применяется та же секция `crl`, но источник задаётся так:

```
status-source: db-first
```

В режиме db-first секция `database` должна содержать действующие параметры подключения к БД ЦС. Значение `crl-path` берётся из принятой процедуры доставки CRL обслуживаемого ЦС. Если указан `delta-crl-path`, соответствующий файл становится обязательным и должен загружаться вместе с основным CRL.

Значения `reload-timeout`, `time-skew`, `poll-interval` и `debounce` записываются с единицей измерения, например 300ms, 5s, 30s или 1m. Допуск `time-skew` должен соответствовать принятой политике синхронизации времени.

Метод `inotify` предназначен для Linux и отслеживает изменение либо атомарную замену файла по точному пути. На другой операционной системе выберите `poll`.

Если задан `delta-crl-path`, Delta CRL проверяется и загружается вместе с основным CRL. При определении статуса запись из Delta CRL имеет приоритет.

Если серийный номер отсутствует и в Delta CRL, и в основном CRL, mOCSP возвращает статус good. В режиме `crl-only` сервис не обращается к БД и не проверяет, был ли сертификат с таким серийным номером действительно выпущен. Поэтому статус good в этом режиме означает отсутствие записи об отзыве в загруженных CRL. Запись с причиной отзыва removeFromCRL также интерпретируется как статус good.

Управление службой (на время перезапуска ответчик не принимает запросы):

```
su - itc-svc
systemctl --user restart mocsp.service
```

Проверка ответчика командой openssl приведена в разделе «Развертывание OCSP-ответчика (mOCSP)», а типовые неисправности описаны в главе «Диагностика и устранение неисправностей».

5.4.6 Архиватор

Архиватор (ITC.Api.MiniCA.Archiver, .NET) по расписанию переносит из рабочей БД ЦС в отдельную архивную БД сертификаты с давно истекшим сроком действия, а также запросы и записи журнала событий старше заданного срока. Так рабочая БД остаётся компактной, а история выпуска сохраняется. Архивная БД по умолчанию называется cwica_archive, её схема та же, что у рабочей БД.

Конфигурационный файл: /opt/itc/itc/config/itc_api_miniCA_archiver_config.json. Юнит: archiver.service (порт 9507, клиентам не публикуется). Архиватору нужен сетевой доступ к серверам рабочей и архивной БД (по умолчанию 5432/tcp).

Ключевые параметры (элемент массива `Tasks`, одно задание архивации):

Параметр	Назначение	Значение по умолчанию
`Tasks[].Key`	Идентификатор задания архивации	ClearwayCA
`Tasks[].Cron`	Расписание запуска (выражение cron); время считается в UTC	@daily (раз в сутки в 00:00 UTC)
`Tasks[].Source`	Строка подключения к рабочей БД ЦС в формате Npgsql: Host, Port, Database, Username, Password	Параметры рабочей БД, указанные при установке (БД cwica, пользователь cwica)
`Tasks[].Target`	Строка подключения к архивной БД в том же формате	Параметры архивной БД, указанные при установке (БД cwica_archive, пользователь cwica)
`Tasks[].CertificateDays`	Сколько дней должно пройти после окончания срока действия сертификата, чтобы он был перенесён в архив	30
`Tasks[].EventDays`	Возраст запросов и записей журнала событий, после которого они переносятся в архив, дней	50

`Tasks[].BatchSizes.Certificate`, `.Request`, `.Event`	Сколько сертификатов, запросов и событий переносится за одну итерацию	5, 50, 100
`Tasks[].Ess`	Пароли в строках подключения заданы ссылками на секреты во внешнем хранилище секретов (EXC)	Не задан

Пример файла приведён в шаге 4 раздела «Развертывание Архиватора». Изменение расписания и проверка работы архиватора описаны в разделе «Архивирование истекших сертификатов». О получении паролей из EXC см. раздел «Хранение секретов во внешнем хранилище (EXC)», о защищённом подключении к БД см. подраздел «Остальные подключения» главы Обеспечение безопасной среды (see page 167).

Управление службой:

```
su - itc-svc
systemctl --user restart archiver.service
```

5.4.7 Сервис резервного копирования

Сервис резервного копирования (ITC.Api.MiniCA.Backup, .NET) выполняет задания резервного копирования и восстановления ЦС. По расписанию cron и по командам из Контрольной панели он копирует БД ЦС и файлы узлов ЦС, хранит копии и восстанавливает ЦС из копии. Пользователи работают с ним в разделе «Задачи → Резервное копирование»: серверная часть Контрольной панели передаёт запросы раздела сервису по адресу из `Services.BackupUrl` (см. «Контрольная панель») вместе с токеном пользователя, а права проверяет сам сервис. Порядок работы с заданиями и требования к серверу сервиса (утилиты `pg_dump` и `pg_restore`, каталог копий, права на файлы) и к узлам ЦС описаны в разделе «Резервное копирование».

Порт сервиса задаётся параметром запуска `--urls`, в примерах `http://0.0.0.0:8409`. Кроме того, сервис слушает локальный адрес 127.0.0.1 на порту `SshAskPass.LocalPort`: через этот канал askpass узлы ЦС получают пароль для sudo по обратному туннелю SSH. Серверная часть Контрольной панели тоже занимает локальный порт askpass, у обеих служб по умолчанию 5225. Если сервис работает на сервере Контрольной панели, задайте ему другой порт, например 5226, в его собственном файле конфигурации, а не в общем.

Конфигурационные файлы:

Файл	Назначение

/opt/itc/itc/config/ itc_api_miniCA_backup_config.json	Параметры сервиса. Ключи задаются на верхнем уровне файла, без общей секции. Права только для владельца (0600): при более широких правах сервис не запускается
/opt/itc/itc/config/ itc_api_miniCA_backup_config.runtime.json	Задания, созданные и изменённые в интерфейсе Контрольной панели. Файл записывает сам сервис
/opt/itc/itc/config/itc_api_miniCA_config.json	Общий файл .NET-сервисов сервера. Из него сервис берёт секции `OpenId` (проверка токенов пользователей) и `Ess` (пароли по ссылкам EXC)

Секции `OpenId` и `Ess` можно задать и в `itc_api_miniCA_backup_config.json`: файлы читаются по порядку, описанному в подразделе «Конфигурационные файлы .NET-сервисов». Секцию `OpenId` из файла `itc_api_miniCA_controlPanel_config.json` сервис не читает. Значения `OpenId` должны соответствовать настройкам Контрольной панели, потому что сервис проверяет тот же токен пользователя.

Ключевые параметры:

Параметр	Назначение	Значение по умолчанию
`StandId`	Идентификатор установки. Входит в имя архива копии	Не задан
`StoragePath`	Каталог копий на сервере сервиса. Обязателен	Не задан; например `/opt/itc/itc/backups`
`RetentionDays`	Срок хранения копий, дней. Последняя пригодная копия задания не удаляется	30
`MinFreeSpaceBytes`	Наименьший свободный объём в каталоге копий, байт. При меньшем объёме операция не начинается	1073741824 (1 ГиБ)
`OperationTimeoutMinutes`	Предельная длительность операции на одном узле, минут	60
`Tools.PgDumpPath`, `Tools.PgRestorePath`	Пути к утилитам `pg_dump` и `pg_restore`. Без пути утилиты ищутся в каталогах переменной PATH	`pg_dump`, `pg_restore`

`SshAskPass.LocalPort`	Локальный порт канала askpass на адресе 127.0.0.1	5225
`SshAskPass.CertificatePath`, `SshAskPass.CertificatePassword`	Сертификат TLS в формате PFX и его пароль для HTTPS-канала askpass	Не заданы
`Encryption.Enabled`	Шифровать копии алгоритмом AES-256-GCM; к имени зашифрованной копии добавляется `.enc`	false
`Encryption.KeyFile`	Файл ключа шифрования: 32 байта в base64 или hex. Размещайте его вне каталога копий	Не задан
`Mailing.Enabled`	Отправлять письма о неуспешном копировании и восстановлении. Об успешных операциях письма не отправляются	false
`Mailing.Host`, `Mailing.Port`	SMTP-сервер и его порт	Не задан, 25
`Mailing.Email`	Адрес отправителя	Не задан
`Mailing.UserName`, `Mailing.PasswordKey`	Учётная запись SMTP и имя переменной окружения службы, в которой задан её пароль. Если `UserName` пуст, письма отправляются без аутентификации	Не заданы
`Mailing.SecureSocket`	Режим TLS при подключении к SMTP-серверу: `None`, `Auto`, `SslOnConnect`, `StartTls`, `StartTlsWhenAvailable`	`StartTls`
`Mailing.Recipients`	Адреса получателей писем	`[]`
`Tasks`	Задания резервного копирования	`[]`

Общие параметры меняются только правкой файла, в интерфейсе Контрольной панели их нет. Секция `Mailing` сервиса не связана с почтовыми настройками Контрольной панели.

Задания в файле. Задания обычно создают в интерфейсе, и сервис сам записывает их в файл itc_api_miniCA_backup_config.runtime.json. Поля задания в файле соответствуют полям формы (см. «Создание задания» в разделе «Резервное копирование»):

Параметр	Поле формы	Значение по умолчанию
`Key`	Ключ. Уникален, после создания не меняется	Нет
`Description`	Описание	Пустая строка
`MiniCa`	ЦС задания: наименование подключения ЦС в Контрольной панели. При сохранении из интерфейса подставляется текущий ЦС	Нет
`Cron`	Cron: пять полей, время UTC	`0 2 * * *`
`Database.Host`, `Database.Port`, `Database.Name`, `Database.Username`	Блок «База данных ЦС»: одна БД на задание, общая для всех узлов	Порт 5432
`Database.Password`, `Database.Ess`	Пароль БД ЦС или ссылка EXC `<mountPoint>:<path>:<key>` при `Ess: true`	`Ess: false`
`Nodes[].Name`	Имя узла, уникальное в задании	Нет
`Nodes[].Ssh.Host`, `Nodes[].Ssh.Port`, `Nodes[].Ssh.Username`	SSH: хост, порт, пользователь	Порт 22
`Nodes[].Ssh.Password`, `Nodes[].Ssh.Ess`	Пароль SSH или ссылка EXC при `Ess: true`	`Ess: false`
`Nodes[].ServiceUnit`	systemd-юнит службы ЦС	`minica`
`Nodes[].SystemdUserScope`	systemctl --user: служба ЦС работает как пользовательский юнит	true

`Nodes[].SystemUser`	Пользователь сервиса: технологическая учётная запись, владелец службы ЦС	`itc-svc`
`Nodes[].BinaryPath`	Бинарник: путь к исполняемому файлу ЦС. Обязателен	Нет
`Nodes[].ConfigPaths`	Файлы конфигурации: не меньше одного пути, порядок задаёт порядок восстановления	Нет

Пример файла `/opt/itc/itc/config/itc_api_miniCA_backup_config.json` для сервиса на сервере Контрольной панели с шифрованием и почтовыми уведомлениями. Задания в нём не перечислены: их создают в интерфейсе.

```
{
  "StandId": "prod",
  "StoragePath": "/opt/itc/itc/backups",
  "RetentionDays": 30,
  "MinFreeSpaceBytes": 1073741824,
  "OperationTimeoutMinutes": 60,
  "Tools": {
    "PgDumpPath": "pg_dump",
    "PgRestorePath": "pg_restore"
  },
  "SshAskPass": {
    "LocalPort": 5226,
    "CertificatePath": "/opt/itc/itc/certs/backup-askpass.pfx",
    "CertificatePassword": "<пароль PFX>"
  },
  "Encryption": {
    "Enabled": true,
    "KeyFile": "/opt/itc/itc/certs/backup-aes.key"
  },
  "Mailing": {
    "Enabled": true,
    "Host": "smtp.example.local",
    "Port": 25,
    "Email": "ca-backup@example.local",
    "UserName": "",
    "PasswordKey": "",
    "SecureSocket": "StartTls",
    "Recipients": ["ca-admins@example.local"]
  },
  "Tasks": []
}
```

Секция `Ess` задаётся в том же формате, что у Контрольной панели, сервиса публикации и архиватора (см. «Хранение секретов во внешнем хранилище (EXC)» главы Обеспечение безопасной среды (see page 167)). Ссылки EXC в заданиях сервис разрешает сам в момент

операции, поэтому секция `Ess` нужна в конфигурации, которую читает сервис резервного копирования. Секция `Ess` Контрольной панели для этого не используется.

Управление службой. После правки файла конфигурации перезапустите службу сервиса резервного копирования. Перезапускайте её, когда ни по одному заданию не выполняется операция: операция, прерванная остановом службы, завершается неуспехом, а прерванное восстановление при следующем запуске сервиса откатывается к снимкам.

Журнал сервиса описан в главе Журналирование и мониторинг (see page 256), типовые отказы приведены в главе «Диагностика и устранение неисправностей», раздел «Резервное копирование и восстановление».

5.4.8 SCEP-адаптер

SCEP-адаптер (сервис itc.acm.scepAdapter.clearwayCa, .NET) выпускает и обновляет сертификаты сетевых устройств и серверов по протоколу SCEP. Адаптер публикуется через Nginx; внешний адрес для клиентов: `https://<fqdn>/api/v1/scep`. Путь чувствителен к регистру букв.

Право на выпуск клиент подтверждает одноразовым паролем (challenge), вложенным в запрос. Пароль выдаётся по адресу `https://<fqdn>/api/v1/scep/challenge-password` пользователю, вошедшему через сервис OpenID: запрос выполняется с токеном доступа OpenID. Сами операции SCEP токена не требуют. Пароль подходит для одного запроса, в том числе на обновление сертификата, и по умолчанию действует 60 минут.

Состояние адаптера, в том числе выданные пароли, хранится в локальной БД SQLite. Сертификаты выпускаются в ЦС по шаблону из параметра `CaTemplate`.

Каталоги, юнит и файлы:

- каталог приложения: /opt/itc/itc.acm.scepAdapter.clearwayCa;
- юнит systemd: scepAdapter.service (файл /home/itc-svc/.config/systemd/user/scepAdapter.service), порт приложения 8440;
- конфигурационный файл: /opt/itc/itc/config/itc_acm_scepAdapter_clearwayCa_config.json;
- блок Nginx /opt/itc/itc/nginx/scep.conf: `location /api/v1/scep` передаёт запросы на `http://localhost:8440/api/v1/scep`.

Ключевые параметры:

Параметр	Назначение	Значение по умолчанию
`Database.Provider`	Провайдер БД адаптера	sqlite
`ConnectionStrings.DefaultConnection`	Строка подключения к БД адаптера	`Data Source=/opt/itc/itc/sqlite/scep.db`

`CaChainPath`	Цепочка сертификатов ЦС (корневой и выдающий ЦС) без сертификата адаптера	/opt/itc/itc/certs/scep-chain.pem
`ScepCertPath` / `ScepKeyPath`	Сертификат адаптера (RA) и его ключ: ими адаптер подписывает ответы клиентам	/opt/itc/itc/certs/scep.crt / /opt/itc/itc/certs/scep.key
`SupportedAlgorithms`	Поддерживаемые алгоритмы подписи	`["SHA256withRSA", "SHA384withECDSA"]`
`SupportsRenewal`	Поддержка обновления сертификатов	true
`AuthenticationConfig.AuthenticationType`	Способ аутентификации при выдаче одноразовых паролей	openId
`AuthenticationConfig.OpenId.Authority`	Адрес сервиса OpenID (issuer)	Адрес OpenID, указанный при установке
`AuthenticationConfig.OpenId.IgnoreCertificateErrors`	Не проверять TLS-сертификат сервиса OpenID	true
`ClearwayCaClientConfig.CaTemplate`	Шаблон ЦС, по которому выпускаются сертификаты	tls
`ClearwayCaClientConfig.CaCluster`	Имя подключения к ЦС; совпадает с ключом в блоке MiniCa	Имя, заданное при установке (по умолчанию sub-ca или root-ca)
`ClearwayCaClientConfig.MiniCa`	Подключение к ЦС: Url и учётные данные Auth.Tls, Auth.Sign, Auth.Jwt. Формат такой же, как у Контрольной панели	Задаётся на шаге настройки адаптера

Пример файла приведён в шаге 5 раздела «Развертывание адаптера SCEP». Порядок получения одноразового пароля и выпуска сертификата клиентом (например, службой certmonger) описан в Руководстве пользователя, раздел «SCEP».

Состав адаптеров и хранилище

Установщик разворачивает адаптеры SCEP и MS-WSTEP; адаптеры ACMEv2, CMP, EST и MS-RPC поставляются отдельно. По умолчанию SCEP-адаптер использует БД SQLite (/opt/itc/itc/sqlite/scep.db) и аутентификацию OpenID; настройки Kerberos и PostgreSQL для этого сервиса не применяются.

Управление службой:

```
su - itc-svc
systemctl --user restart scepAdapter.service
sudo nginx -t && sudo systemctl reload nginx     # после изменения конфигурации Nginx;
выполняется администратором с правом sudo
```

Причины отклонения запросов SCEP записываются в журнал адаптера (см. «Адаптер SCEP» в главе «Диагностика и устранение неисправностей»).

5.4.9 Точка распространения CRL и сертификатов (CDP)

Точка распространения работает на веб-сервере Nginx и раздаёт по HTTP списки отзыва (CRL и Delta CRL) и сертификаты ЦС для построения цепочки доверия (AIA). Эти адреса ЦС записывает в поля CDP и AIA выпускаемых сертификатов, поэтому от их доступности зависят проверка статуса и построение цепочки сертификатов у внешних потребителей.

Корневой каталог публикации: /opt/itc/web. Файлы размещаются в подкаталогах cdp/ (списки отзыва) и aia/ (сертификаты ЦС). Конфигурационный файл сайта: /opt/itc/web/nginx.conf. Nginx подключает его символической ссылкой cdp.conf: в каталоге /etc/nginx/sites-enabled.d/ на ALT Linux, в каталоге /etc/nginx/conf.d/ на остальных ОС. Раздача ведётся по HTTP на порту 80; откройте 80/tcp для клиентов.

Содержимое файла /opt/itc/web/nginx.conf после установки (`<fqdn>` обозначает имя сервера CDP):

```
server {
    listen 80;
    server_name <fqdn>;
    root /opt/itc/web;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Адреса раздачи:

Ресурс	URL	Файл на сервере CDP
Список отзыва (CRL)	`http://<fqdn>/cdp/<CN>.crl`	/opt/itc/web/cdp/ `<CN>` .crl
Список отзыва (Delta CRL)	`http://<fqdn>/cdp/<CN>+.crl`	/opt/itc/web/cdp/ `<CN>` +.crl
Сертификат ЦС (AIA)	`http://<fqdn>/aia/<CN>.crt`	/opt/itc/web/aia/ `<CN>` .crt

Здесь `<CN>` обозначает общее имя (CN) соответствующего ЦС, в котором пробелы заменены на «_». Адреса, которые ЦС записывает в сертификаты, задают параметры `ca.cdp`, `ca.aia` и `ca.freshest-crl` (см. «Шаблоны, поля CDP/AIA и сроки действия» в разделе Настройка ядра ЦС (see page 126)). Адрес Delta CRL по умолчанию записывается только в сертификаты выдающего ЦС.

Списки отзыва в каталог cdp/ помещает сервис публикации по заданиям публикации CRL. Если сервис публикации работает на сервере CDP, в задании используется локальное копирование в /opt/itc/web/cdp/, иначе используется передача по SCP (см. «Задания публикации» в разделе «Развертывание сервиса публикации»).

5.4.9.1 Размещение сертификата ЦС для AIA

Сертификат каждого ЦС (корневого и выдающего) размещают в каталоге aia/ под именем `<CN>.crt`. По RFC 5280 по адресу caIssuers ожидается сертификат в кодировке DER. Повторите процедуру после замены сертификата ЦС.

Шаг 1. На сервере ЦС от имени itc-svc сохраните сертификат ЦС в кодировке DER:

```
openssl x509 -in /opt/itc/minica/ca/ca.crt -outform DER -out /tmp/<CN>.crt
```

Шаг 2. Скопируйте файл на сервер CDP в каталог /opt/itc/web/aia/ от имени itc-svc, владельца каталога /opt/itc/web.

Проверка. С любого узла, которому доступен сервер CDP, скачайте сертификат и выведите его субъект. Первая команда должна вывести код `200`, а вторая должна вывести субъект ЦС.

```
curl -s -o /tmp/aia-check.crt -w '%{http_code}\n' "http://<fqdn>/aia/<CN>.crt"
openssl x509 -inform DER -in /tmp/aia-check.crt -noout -subject
```

Так же проверяется доступность списка отзыва: для адреса `http://<fqdn>/cdp/<CN>.crl` ожидается код `200`.

Управление службой. CDP обслуживает системный Nginx, а не пользовательский юнит. После изменения /opt/itc/web/nginx.conf проверьте синтаксис и перезагрузите Nginx (от учётной записи с правом sudo):

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

6 Обеспечение безопасной среды

После установки Clearway CA приведите продукт и среду его работы в безопасное состояние (hardening). Глава описывает, что и зачем настроить, какие значения действуют после установки и как проверить результат.

Порядок приведения в безопасное состояние:

1. Проверьте TLS на подключениях к ЦС и при необходимости включите взаимную аутентификацию (см. «Защита соединений центра сертификации (TLS/mTLS)»).
2. Включите TLS на каналах к СУБД и каталогу LDAP (см. «Защита подключений к СУБД и каталогу LDAP»).
3. Закройте служебные порты межсетевым экраном (см. «Ограничение сетевого доступа»).
4. Приведите права на файлы к минимально необходимым (см. Ограничение прав доступа к файлам (see page 178)).
5. Если в организации есть внешнее хранилище секретов, перенесите в него пароли и ключи (см. «Хранение секретов во внешнем хранилище (EXC)»).
6. При необходимости перенесите ключ ЦС на аппаратный носитель (см. «Хранение ключа ЦС на HSM/Rutoken (PKCS#11)»).
7. Настройте шифрование данных сертификатов и хранение ключей пользователей (см. «Шифрование данных сертификатов в БД» и «Механизм KRA (резервное копирование и восстановление ключей)»).
8. Включите синхронизацию времени на всех узлах (см. «Синхронизация времени и защита от повторов»).
9. Защитите учётные записи и вход в Контрольную панель (см. «Защита учётных записей и входа»).
10. Назначьте пользователям и сервисам минимально необходимые права (см. «Разграничение доступа»).

Параметры ядра ЦС задаются в файле `/opt/itc/minica/conf/minica.yml`, а параметры остальных компонентов задаются в их конфигурационных файлах (см. Настройка ядра ЦС (see page 126) и Настройка вспомогательных сервисов (see page 139)). Редактируйте файлы от имени технологической учётной записи `itc-svc`: так у файла сохраняются владелец и права.

Службы Clearway CA работают как пользовательские юниты systemd учётной записи `itc-svc`. Изменения конфигурации вступают в силу после перезапуска службы компонента. Пример для ядра ЦС (на сервере ЦС):

```
su - itc-svc
systemctl --user restart minica.service
systemctl --user status minica.service
```

Имена юнитов: `minica.service` (ядро ЦС), `mocsp.service` (OCSP-ответчик), `controlPanel.service` (Контрольная панель), `publication.service` (сервис публикации), `archiver.service` (архиватор), `scepAdapter.service` (SCEP-адаптер), `wstepCep.service` и `wstepCesAdapter.service` (службы CEP и CES адаптера MS-WSTEP), `openid.service` (OpenID). Подробнее см. раздел «Управление службами».

Перезапуск служб

На время перезапуска служба не обслуживает запросы. Вносите изменения в период обслуживания и после перезапуска проверяйте состояние службы командой `systemctl --user status <юнит>`.

6.1 Защита соединений центра сертификации (TLS/mTLS)

TLS шифрует канал между ядром ЦС и его клиентами: Контрольной панелью, сервисом публикации, SCEP-адаптером, службами MS-WSTEP и утилитой mclient. Взаимная аутентификация (mTLS) дополнительно требует от клиента предъявить сертификат, которому доверяет ЦС. Клиент без такого сертификата к ЦС не подключится.

Установщик включает TLS при установке корневого и выдающего ЦС. После установки ЦС принимает подключения по HTTPS на порту 9443, HTTP-порт 9080 не используется. Взаимная аутентификация после установки выключена (`tls.verify: false`).

В поставляемом шаблоне `minica.yml` TLS выключен (`tls.enable: false`). Поэтому при ручном развертывании ЦС работает по HTTP на порту 9080, пока не выполнен шаг включения TLS.

Параметры секции `tls` файла `/opt/itc/minica/conf/minica.yml` :

Параметр	Назначение	После установки	Безопасное значение
`tls.enable`	Включает TLS: ЦС принимает подключения по HTTPS на порту `https-port` (9443)	true	true
`tls.verify`	Включает mTLS: клиент обязан предъявить сертификат, доверенный по `tls.ca-cert` или `tls.ca-certs`	false	true

`tls.key`	Закрытый ключ сервера ЦС	`/opt/itc/minica/conf/tls.key`	Без требований
`tls.cert`	Сертификат сервера ЦС	`/opt/itc/minica/conf/tls.crt`	Без требований
`tls.ca-cert`	Сертификат ЦС, по которому проверяются клиентские сертификаты (mTLS)	`/opt/itc/minica/conf/ca.crt`	Сертификат ЦС, выпустившего клиентские сертификаты, например `/opt/itc/minica/ca/ca.crt`
`tls.ca-certs`	Дополнительные доверенные сертификаты (mTLS)	[]	По необходимости
`tls.insecure`	Отключает проверку сертификата ЦС	false	false

6.1.1 Включение взаимной аутентификации (mTLS)

После включения mTLS ЦС отклоняет подключения клиентов, которые не предъявили доверенный клиентский сертификат. Клиентский сертификат выпускается по шаблону `tls`: такой сертификат можно использовать и для аутентификации клиента (назначение clientAuth).

Порядок включения

Включайте `tls.verify` в minica.yml только после того, как у каждого клиента ЦС (Контрольной панели, сервиса публикации, SCEP-адаптера, служб MS-WSTEP и mclient) настроено предъявление клиентского сертификата. Иначе эти компоненты не смогут обращаться к ЦС.

Все шаги выполняются на сервере ЦС в сеансе технологической учётной записи `itc-svc`.

Шаг 1. Выпустите клиентский ключ и сертификат для mclient. Команды повторяют выпуск TLS-сертификата ЦС (см. «Включение TLS» в разделе «Развертывание ядра ЦС (minica)»), но создают файлы `mtls.key` и `mtls.crt`: их mclient по умолчанию предъявляет как клиентские. Последние две команды ограничивают доступ к ключу и удаляют временный файл запроса.

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
openssl req -new -keyout /opt/itc/minica/conf/mtls.key -out /opt/itc/minica/conf/
mtls.req \
```

```
  -sha256 -nodes -subj "/CN=mclient" -addext "extendedKeyUsage = serverAuth,
clientAuth"
/opt/itc/minica/mclient ca -csr /opt/itc/minica/conf/mtls.req \
  -out /opt/itc/minica/conf/mtls.crt -template tls
chmod 600 /opt/itc/minica/conf/mtls.key
rm /opt/itc/minica/conf/mtls.req
```

Шаг 2. В секции `tls` файла `/opt/itc/minica/conf/mclient.yml` задайте `verify: true` и проверьте пути к клиентскому ключу и сертификату. Остальные параметры секции не меняйте.

```
tls:
  enable: true
  verify: true                          # предъявлять клиентский сертификат (mTLS)
  key: /opt/itc/minica/conf/mtls.key    # клиентский ключ
  cert: /opt/itc/minica/conf/mtls.crt   # клиентский сертификат
  # остальные параметры секции не меняются
```

Шаг 3. В секции `tls` файла `/opt/itc/minica/conf/minica.yml` включите проверку клиентских сертификатов. В `ca-cert` укажите файл с сертификатом ЦС, который выпустил клиентские сертификаты. Если клиентские сертификаты выпустил этот же ЦС, укажите `/opt/itc/minica/ca/ca.crt`.

```
tls:
  enable: true                          # HTTPS на порту 9443
  verify: true                          # mTLS: проверка клиентского сертификата
  key: /opt/itc/minica/conf/tls.key     # закрытый ключ сервера
  cert: /opt/itc/minica/conf/tls.crt    # сертификат сервера
  ca-cert: /opt/itc/minica/ca/ca.crt    # сертификат ЦС, выпустившего клиентские
сертификаты
  ca-certs: []                          # дополнительные доверенные сертификаты
  insecure: false                       # не отключать проверку сертификатов
```

Шаг 4. Перезапустите службу ЦС. На время перезапуска ЦС не принимает запросы.

```
systemctl --user restart minica.service
```

Проверка. Убедитесь, что ЦС принимает подключения только по HTTPS. Команда выполняется на сервере ЦС администратором с правом sudo:

```
sudo ss -ltn | grep -E ':(9443|9080)\b'
```

В выводе должен быть порт 9443 и не должно быть порта 9080. Затем в сеансе `itc-svc` проверьте, что ЦС принимает запросы mclient с клиентским сертификатом: `mclient ping` должен вывести версию ЦС и SKI.

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ping
```

Наконец, убедитесь, что ЦС отклоняет подключение без клиентского сертификата. Команда выполняется на любом узле, которому разрешено подключение к порту 9443. Вместо `<fqdn ЦС>` укажите имя, на которое выпущен TLS-сертификат ЦС:

```
curl -k --max-time 5 https://<fqdn ЦС>:9443/
```

Ожидается ошибка TLS-соединения: curl сообщает об ошибке SSL. Если вместо ошибки получен HTTP-ответ, проверка клиентских сертификатов не включена. Коды ошибок mclient описаны в главе «Диагностика и устранение неисправностей».

> **Параметр tls.insecure**
>
> Значение `tls.insecure: true` отключает проверку сертификата и допустимо только для отладки. В рабочей конфигурации задайте false.

6.2 Защита подключений к СУБД и каталогу LDAP

Каналы от компонентов к СУБД PostgreSQL, от OpenID и сервиса публикации к каталогу LDAP и от служб .NET к OpenID защищаются отдельно от TLS самого ЦС. После установки часть этих каналов допускает подключение без шифрования. Для безопасного состояния включите TLS и проверку сертификата сервера на каждом канале.

6.2.1 Подключение ядра ЦС и mOCSP к СУБД

За канал «ЦС → СУБД» отвечает параметр `allow-without-tls` в секции `database` файла `minica.yml`. Его не следует путать с секцией `tls`, которая защищает подключения клиентов к ЦС. У OCSP-ответчика mOCSP такой же параметр в файле `/opt/itc/ocsp/conf/mocsp.yml`.

Параметр	Назначение	После установки	Безопасное значение

`database.allow-without-tls`	Разрешает подключение к БД без TLS. При false служба завершает работу, если сервер БД не поддерживает TLS, и не передаёт данные по открытому каналу	true	false
`database.url`	Строка подключения к БД. Параметр `sslmode=verify-full` включает проверку сертификата сервера БД	`postgres://<пользователь>:<пароль>@<сервер БД>:<порт>/<БД>` (по умолчанию пользователь и БД называются cwica)	Та же строка с `sslmode=verify-full`

Без `sslmode=verify-full` служба даже при включённом TLS доверяет любому сертификату сервера БД. Сертификат ЦС, которым проверяется сертификат сервера БД, указывается в строке подключения параметром `sslrootcert`. Специальные символы в пароле кодируйте по правилам URL.

Шаг 1. Включите TLS на сервере PostgreSQL (см. «Сервер PostgreSQL» ниже).

Шаг 2. На сервере ЦС от имени `itc-svc` измените секцию `database` файла `/opt/itc/minica/conf/minica.yml`:

```
database:
  url: postgres://<пользователь>:<пароль>@<сервер БД>:<порт>/<БД>?sslmode=verify-full&sslrootcert=<путь к сертификату ЦС сервера БД>
  allow-without-tls: false   # запретить подключение к БД без TLS
```

Шаг 3. Перезапустите службу ЦС:

```
systemctl --user restart minica.service
```

Шаг 4. На сервере mOCSP от имени `itc-svc` внесите такие же изменения в `/opt/itc/ocsp/conf/mocsp.yml` и перезапустите `mocsp.service`.

Проверка. Убедитесь, что значение сохранено и служба работает. На сервере ЦС в сеансе `itc-svc`:

```
grep -n "allow-without-tls" /opt/itc/minica/conf/minica.yml
systemctl --user status minica.service
```

На сервере mOCSP в сеансе `itc-svc`:

```
grep -n "allow-without-tls" /opt/itc/ocsp/conf/mocsp.yml
systemctl --user status mocsp.service
```

Ожидается `allow-without-tls: false` и состояние `active (running)`. Если служба не запускается, причину покажет журнал: `journalctl --user -u minica.service -n 50` (для mOCSP: `journalctl --user -u mocsp.service -n 50`).

6.2.2 Сервер PostgreSQL

На сервере СУБД включите `ssl = on`, задайте сертификат и ключ сервера и минимальную версию протокола не ниже TLS 1.2. Пример фрагмента `postgresql.conf` для PostgreSQL 12–18:

```
ssl = on
ssl_cert_file = '/etc/postgresql/tls/server.crt'
ssl_key_file = '/etc/postgresql/tls/server.key'
ssl_min_protocol_version = 'TLSv1.2'
```

Сервис поддерживает PostgreSQL 11–18. В PostgreSQL 11 параметра `ssl_min_protocol_version` нет, поэтому для этой версии ограничение «TLS 1.2 и выше» настраивается отдельно.

- Файл ключа сервера должен принадлежать учётной записи postgres и иметь права 0600: при более широких правах PostgreSQL не примет ключ.
- Чтобы СУБД сама отклоняла подключения без TLS, используйте в `pg_hba.conf` для подключений компонентов записи `hostssl` вместо `host`.

Команды выполняются на сервере СУБД администратором с правом sudo. Первая показывает путь к `postgresql.conf` (он зависит от ОС), вторая перечитывает конфигурацию, третья проверяет, что TLS включён:

```
sudo -u postgres psql -c "SHOW config_file;"
sudo -u postgres psql -c "SELECT pg_reload_conf();"
sudo -u postgres psql -c "SHOW ssl;"
```

Ожидается `on`. Если в конфигурации ошибка, PostgreSQL сохраняет прежние настройки и записывает причину в свой журнал. Проверьте его. После перенастройки и перезапуска служб убедитесь, что их подключения защищены: выполните запрос на каждом сервере СУБД. Запрос

показывает подключения к БД ЦС (`cwica`), архивной БД (`cwica_archive`) и БД OpenID (`openid`); если при установке заданы другие имена БД, укажите их:

```
sudo -u postgres psql -c "SELECT a.datname, a.usename, a.client_addr, s.ssl, s.version
FROM pg_stat_ssl s JOIN pg_stat_activity a USING (pid) WHERE a.datname IN ('cwica',
'cwica_archive', 'openid');"
```

У подключений служб в столбце `ssl` должно быть `t` , а в столбце `version` TLSv1.2 или TLSv1.3.

6.2.3 Остальные подключения

Канал	Где задаётся	После установки	Что сделать
OpenID → СУБД	`/opt/itc/openid/config.yaml` , параметр `storage.config.ssl.mode`	disable	Задать режим с проверкой сертификата сервера (`verify-full`), перезапустить `openid.service`
Архиватор → рабочая и архивная БД	`/opt/itc/itc/config/itc_api_miniCA_archiver_config.json` , строки подключения `Tasks[].Source` и `Tasks[].Target`	Параметр `SSL Mode` не задан	Добавить в обе строки `SSL Mode=VerifyFull` и `Root Certificate=<путь>` , перезапустить `archiver.service`
OpenID → каталог LDAP	`/opt/itc/openid/config.yaml` , секция `connectors[].config`	`insecureNoSSL: true` : подключение по ldap:// (порт 389) без шифрования	Перейти на LDAPS (порт 636) с проверкой сертификата каталога, перезапустить `openid.service`
Сервис публикации → каталог LDAP	Задача публикации, блок LDAP, флажок «Защищённое соединение»	Задаётся при создании задачи	Включить «Защищённое соединение» (LDAPS) во всех задачах публикации в LDAP

Контрольная панель, сервис публикации, SCEP-адаптер, сервис резервного копирования → OpenID	Файлы `/opt/itc/itc/config/*.json`: параметр `OpenId.IgnoreCertificateErrors` у Контрольной панели и сервиса резервного копирования, `OpenId.ignoreCertificateErrors` у сервиса публикации, `AuthenticationConfig.OpenId.IgnoreCertificateErrors` у SCEP-адаптера	true: TLS-сертификат OpenID не проверяется	Добавить сертификат корневого ЦС в доверенные на сервере служб (см. «Добавление сертификата ЦС в доверенные»), задать false, перезапустить `controlPanel.service`, `publication.service`, `scepAdapter.service` и службу сервиса резервного копирования

Пример строки подключения архиватора с проверкой сертификата сервера БД:

```
"Source": "Host=<сервер БД>;Port=5432;Database=cwica;Username=cwica;Password=<пароль>;SSL Mode=VerifyFull;Root Certificate=<путь к сертификату ЦС сервера БД>"
```

Пример фрагментов `config.yaml` OpenID: подключение к СУБД с проверкой сертификата и коннектор LDAP по LDAPS.

```
storage:
  type: postgres
  config:
    ssl:
      mode: verify-full
      caFile: <путь к сертификату ЦС сервера БД>

connectors:
  - type: ldap
    id: ldap
    config:
      host: "<сервер каталога>:636"
      insecureNoSSL: false
      rootCA: <путь к сертификату ЦС каталога>
```

Остальные параметры секций оставьте без изменений. После правки перезапустите `openid.service` на сервере OpenID от имени `itc-svc` и проверьте вход в Контрольную панель учётной записью из каталога.

6.3 Ограничение сетевого доступа

Веб-сервисы Clearway CA публикуются через Nginx (порт 443, TLS). Прикладные службы .NET принимают подключения по HTTP на всех сетевых интерфейсах сервера (адрес 0.0.0.0): Контрольная панель (порт 8407), сервис публикации (9407), архиватор (9507), SCEP-адаптер (8440), сервис резервного копирования (порт из параметра запуска `--urls`, в примерах 8409). Службы CEP и CES адаптера MS-WSTEP принимают подключения по HTTPS на портах 8441 и 8442, тоже на всех интерфейсах.

К портам 8407, 9407, 8440, 8441 и 8442 обращается Nginx на том же сервере, к порту 9407 обращается также серверная часть Контрольной панели (параметр `Services.PublicationCrlUrl`), а к порту сервиса резервного копирования обращается только серверная часть Контрольной панели (параметр `Services.BackupUrl`). Порт архиватора 9507 клиентам не нужен. Закройте эти порты межсетевым экраном для подключений извне.

Канал askpass сервиса резервного копирования (порт `SshAskPass.LocalPort`, по умолчанию 5225) слушает только адрес 127.0.0.1, узлы ЦС обращаются к нему через туннель SSH. Отдельного правила межсетевого экрана он не требует.

Порт	Служба	Правило межсетевого экрана
8407/tcp	Контрольная панель	Закрыть для подключений извне
9407/tcp	Сервис публикации	Закрыть для подключений извне. Если Контрольная панель работает на другом сервере, разрешить подключение только с него (параметр `Services.PublicationCrlUrl`)
9507/tcp	Архиватор	Закрыть для подключений извне
8440/tcp	SCEP-адаптер	Закрыть для подключений извне
8441/tcp, 8442/tcp	MS-WSTEP (службы CEP и CES)	Закрыть для подключений извне
8409/tcp (порт из `--urls`)	Сервис резервного копирования	Закрыть для подключений извне. Если Контрольная панель работает на другом сервере, разрешить подключение только с него (параметр `Services.BackupUrl`)
9443/tcp	Ядро ЦС	Разрешить только с серверов Контрольной панели, сервиса публикации, SCEP-адаптера и MS-WSTEP

5432/tcp	PostgreSQL	Разрешить только с серверов компонентов, которые подключаются к БД (ядро ЦС, mOCSP, архиватор, OpenID, сервис резервного копирования), и с АРМ администраторов БД
22/tcp	SSH	Разрешить с АРМ администраторов и, на время установки, с рабочего места, на котором запущен установщик. К серверам ЦС разрешить подключение также с сервера сервиса резервного копирования (задания резервного копирования), а к серверам nginx и PostgreSQL разрешить подключение с сервера Контрольной панели, если используются типовые задачи

Порты 443 (Nginx и OpenID), 80 (точка распространения CDP) и 8888 (mOCSP) должны оставаться доступными клиентам. Полный перечень портов приведён в разделе «Приложение Б. Сетевые порты».

Проверка. На сервере служб выведите прослушиваемые порты и процессы, которые их открыли (администратор с правом sudo):

```
sudo ss -ltnp
```

Сверьте вывод с таблицей: он показывает, какие порты открыты на сервере. Правила межсетевого экрана на этот вывод не влияют, поэтому доступность портов извне проверяют с другого узла сети. Команды проверяют HTTP-порты 8407, 9407, 9507, 8440 и 8409 (сервис резервного копирования) и HTTPS-порты 8441 и 8442 сервера служб:

```
for p in 8407 9407 9507 8440 8409; do curl -s -o /dev/null --max-time 5 -w "$p: %
{http_code}\n" http://<адрес сервера служб>:$p/; done
for p in 8441 8442; do curl -k -s -o /dev/null --max-time 5 -w "$p: %{http_code}\n"
 https://<адрес сервера служб>:$p/; done
```

Для каждого порта ожидается код `000`: подключение не установлено (отказ или истечение тайм-аута). Любой другой код означает, что порт доступен извне.

6.3.1 Веб-интерфейс установщика

Установщик запускается на рабочем месте оператора (см. «Установщик» в разделе «Общие сведения об установке»). Его веб-интерфейс принимает подключения на порту 5224 на всех сетевых интерфейсах этого рабочего места. На время установки разрешите подключение к порту 5224 только с АРМ администратора.

Если браузер открывается не на рабочем месте оператора, а на другом АРМ, подключайтесь к интерфейсу через SSH-туннель. Команда выполняется на АРМ, где открывается браузер:

```
ssh -L 5224:localhost:5224 <учётная запись>@<рабочее место оператора>
```

После этого откройте в браузере этого АРМ адрес `https://localhost:5224/installer`.

По завершении установки остановите установщик. Если он запущен командой `itc-installer` в терминале, нажмите в этом терминале Ctrl+C. Если он запущен ярлыком «ITC Installer», завершите его процесс на рабочем месте оператора от имени учётной записи, которая его запустила:

```
pkill -u "$USER" -f itc-installer
```

Установщик хранит состояние (базу installer.db, журналы и введённые значения параметров) в каталоге `~/.local/share/itc-installer` учётной записи, от которой он запущен. Ограничьте доступ к каталогу владельцем. Команды выполняются на рабочем месте оператора от имени учётной записи, которая запускала установщик; вторая команда должна вывести `700` и имя этой учётной записи:

```
chmod 700 ~/.local/share/itc-installer
stat -c '%a %U' ~/.local/share/itc-installer
```

6.4 Ограничение прав доступа к файлам

Конфигурационные файлы компонентов содержат пароли к БД и каталогу, а в каталогах `conf`, `ca` и `certs` хранятся закрытые ключи. Если такие файлы доступны на чтение другим учётным записям, секреты может получить любой пользователь сервера. Приведите права к минимально необходимым и повторяйте проверку после установки, обновления и изменения конфигурации.

Каталоги установки по умолчанию: `/opt/itc/minica` для ядра ЦС, `/opt/itc/ocsp` для mOCSP, `/opt/itc/openid` для OpenID, `/opt/itc/itc` для общих файлов служб .NET (конфигураций, сертификатов, журналов).

Правила:

- Владельцем каталогов установки и файлов компонентов должна быть технологическая учётная запись `itc-svc` (группа `itc-svc`).
- Файлы с паролями, закрытые ключи и токены JWT доступны только владельцу: права 0600.
- Открытые сертификаты и открытые ключи можно оставить доступными на чтение.
- Право на запись в исполняемые файлы должно быть только у владельца.

- Каталоги, которые раздаёт Nginx (`/opt/itc/web` , `/opt/itc/itc/env` , `/opt/itc/itc.minica.spa`), должны оставаться доступными на чтение. Не храните в них закрытые ключи и пароли.

Файлы, доступ к которым нужно ограничить:

Компонент	Файлы	Что содержат	Права
Ядро ЦС	`/opt/itc/minica/conf/minica.yml`	Пароль БД в `database.url` , пути к ключам, PIN носителя при хранении ключа ЦС на токене	0600
Ядро ЦС	`/opt/itc/minica/conf/mclient.yml` , `/opt/itc/minica/*.env`	Параметры mclient и окружение службы	0600
Ядро ЦС	`/opt/itc/minica/ca/*.key` , файлы `*.key` в каталогах `/opt/itc/minica/ca/old-keys/` и `/opt/itc/minica/ca/new-keys/`	Закрытый ключ ЦС (`ca.key`), ключ шифрования данных (`encr.key`), прежние ключи ЦС, новый ключ ЦС, созданный при «Перевыпуске» и ожидающий установки сертификата	0600
Ядро ЦС	`/opt/itc/minica/conf/*.key` , `/opt/itc/minica/conf/mclient.jwt`	Ключ TLS (`tls.key`), ключ подписи JWT (`jwt.key`), ключ и токен mclient (`mclient.key` , `mclient.jwt`), клиентский ключ mTLS (`mtls.key`)	0600
Ядро ЦС	`/opt/itc/minica/logs/minica.log`	Прикладной журнал ЦС; права задаёт параметр `log.fileMode` (после установки 0600)	0600

Контрольная панель, сервис публикации, архиватор, SCEP-адаптер, сервис резервного копирования	`/opt/itc/itc/config/*.json`	Конфигурации служб: пароли БД архиватора, подключения ЦС, в том числе добавленные через интерфейс (`itc_api_miniCA_config.runtime.json`), задания резервного копирования с паролями БД ЦС и SSH узлов, если они не заданы ссылками EXC (`itc_api_miniCA_backup_config.json`, `itc_api_miniCA_backup_config.runtime.json`)	0600. Сервис резервного копирования не запускается, если права на `itc_api_miniCA_backup_config.json` шире
Сервис резервного копирования	Файл ключа шифрования копий (`Encryption.KeyFile`), сертификат канала askpass (`SshAskPass.CertificatePath`)	Ключ, которым шифруются копии; закрытый ключ сертификата askpass	0600
Сервис резервного копирования	Каталог копий (`StoragePath`, например `/opt/itc/itc/backups`)	Копии БД ЦС и файлов узлов ЦС, в том числе закрытого ключа ЦС	Каталог 0700, доступ только у учётной записи службы сервиса
Контрольная панель, сервис публикации, SCEP-адаптер, Nginx	`/opt/itc/itc/certs/*.key`, `*.jwt`, `*.pfx`	Ключи подписи запросов к ЦС (`<имя ЦС>.key`), TLS-ключи Nginx и SCEP-адаптера, токены JWT сервисов	0600
SCEP-адаптер	`/opt/itc/itc/sqlite/scep.db`	Одноразовые пароли SCEP	0600

mOCSP	`/opt/itc/ocsp/conf/mocsp.yml`, `/opt/itc/ocsp/conf/mocsp.key`, `/opt/itc/ocsp/mocsp.env`	Пароль БД, ключ подписи OCSP-ответов, окружение службы	0600
MS-WSTEP	`/opt/itc/itc_wstep_env.conf`, `/etc/krb5.keytab`	Пароль учётной записи LDAP, ключи Kerberos узла	0600
OpenID	`/opt/itc/openid/config.yaml`	Пароль БД, пароль привязки к каталогу (`bindPW`), секрет клиента, хэш пароля встроенной учётной записи	0600
OpenID	`/opt/itc/itc/certs/openid-tls.key`	TLS-ключ OpenID	0600

6.4.1 Проверка и исправление прав

Шаг 1. Проверьте текущие права. На каждом сервере выполните от имени администратора с правом sudo (каталоги, которых на сервере нет, команда пропускает):

```
sudo find /opt/itc/minica /opt/itc/ocsp /opt/itc/openid /opt/itc/itc/config /opt/itc/
itc/certs /opt/itc/itc/sqlite -type f -exec stat -c "%a %U:%G %n" {} \; 2>/dev/null
sudo find /opt/itc -xdev \( -type f -o -type d \) -perm -o+w -exec ls -ld {} +
```

Первая команда выводит права, владельца и путь каждого файла. Вторая ищет файлы и каталоги, доступные на запись всем (например, 777): её вывод должен быть пустым.

Шаг 2. На сервере ЦС верните владельца `itc-svc` и ограничьте права на файлы с секретами:

```
sudo find /opt/itc/minica ! -user itc-svc -exec chown itc-svc:itc-svc {} +
sudo find /opt/itc/minica -type f \( -name "*.yml" -o -name "*.env" -o -name "*.key" -o
-name "*.jwt" \) -exec chmod 600 {} +
```

Шаг 3. На сервере служб (Контрольная панель, сервис публикации, архиватор, SCEP-адаптер, сервис резервного копирования) выполните:

```
sudo find /opt/itc/itc ! -user itc-svc -exec chown itc-svc:itc-svc {} +
sudo find /opt/itc/itc/config /opt/itc/itc/sqlite -type f -exec chmod 600 {} +
sudo find /opt/itc/itc/certs -type f \( -name "*.key" -o -name "*.jwt" -o -name "*.pfx"
 \) -exec chmod 600 {} +
```

На сервере сервиса резервного копирования закройте также каталог копий. Вместо `<StoragePath>` подставьте значение параметра из `itc_api_miniCA_backup_config.json`:

```
sudo chmod 700 <StoragePath>
```

На сервере, где установлен mOCSP, выполните:

```
sudo find /opt/itc/ocsp ! -user itc-svc -exec chown itc-svc:itc-svc {} +
sudo find /opt/itc/ocsp -type f \( -name "*.yml" -o -name "*.env" -o -name "*.key" \)
-exec chmod 600 {} +
```

Если на сервере установлен MS-WSTEP, ограничьте также права на его файлы с секретами:

```
sudo chmod 600 /opt/itc/itc_wstep_env.conf /etc/krb5.keytab
```

Шаг 4. На сервере OpenID выполните:

```
sudo chown itc-svc:itc-svc /opt/itc/openid/config.yaml
sudo chmod 600 /opt/itc/openid/config.yaml
sudo find /opt/itc/itc/certs -type f -name "*.key" -exec chmod 600 {} +
```

Исполняемый файл OpenID

Не меняйте владельца файла `/opt/itc/openid/openid`. Установщик назначает ему разрешение `cap_net_bind_service`, чтобы служба могла слушать порт 443, а смена владельца файла в Linux сбрасывает такие разрешения.

Если владелец уже изменён, служба не сможет занять порт 443. В этом случае назначьте разрешение повторно командой `sudo setcap 'cap_net_bind_service=+ep' /opt/itc/openid/openid` и перезапустите `openid.service` от имени `itc-svc`. Проверить разрешение можно командой `getcap /opt/itc/openid/openid`: в выводе должно быть `cap_net_bind_service`.

Проверка. Повторите команды шага 1. У файлов из таблицы должны быть права `600` и владелец `itc-svc:itc-svc`, вторая команда не должна ничего выводить. Перезапуск служб после изменения прав не требуется. Убедитесь, что службы работают: войдите в сеанс технологической учётной записи командой `sudo su - itc-svc` и выполните `systemctl --user status <юнит>` для служб этого сервера.

6.4.2 Утилита change_right.sh

Права можно привести к описанию поставки утилитой `change_right.sh`: она проверяет и выставляет права по описанию из файла `.right`. Утилите передаются путь к каталогу сервиса и файл прав:

```
chmod +x change_right.sh
./change_right.sh /opt/itc/minica minica.right
```

После запуска утилиты проверьте результат командами шага 1.

6.5 Хранение секретов во внешнем хранилище (EXC)

Пароли технологических учётных записей и другие секреты компонентов можно хранить не в конфигурационных файлах, а во внешнем хранилище секретов (далее EXC). Это пароль к БД, пароли закрытых ключей, пароли подключения к SCP и LDAP. В конфигурации тогда указывается не сам пароль, а путь к секрету в EXC, и открытых паролей на серверах Clearway CA не остаётся.

6.5.1 Как это работает

- При запуске служба подключается к EXC, получает секреты по путям из своей конфигурации и хранит их только в оперативной памяти.
- Ручной ввод паролей или других данных при запуске не требуется.
- Служба входит в EXC по клиентскому TLS-сертификату: соединение устанавливается по HTTPS с взаимной аутентификацией.
- Ядро ЦС и mOCSP дополнительно проверяют сертификат EXC по заданному корневому сертификату.
- Секреты считываются только при запуске службы. После изменения секрета в EXC перезапустите службы, которые его используют. Исключение составляют пароли в

заданиях резервного копирования: сервис резервного копирования получает их в момент каждой операции.

6.5.2 Путь к секрету

Путь состоит из трёх частей, разделённых двоеточием: `<точка монтирования>:<коллекция>:<ключ>`. Точка монтирования обозначает имя подключённого в EXC хранилища секретов (например, `kv`), коллекция обозначает набор секретов, ключ обозначает имя конкретного секрета. Пример: `kv:prod:db-pass`. В Руководстве пользователя тот же формат записан как `mountPoint:path:key`.

Точку монтирования, коллекцию, пространство Clearway CA в EXC и клиентский сертификат для входа предоставляет администратор EXC.

6.5.3 Подготовка

1. Получите у администратора EXC адрес EXC, имя пространства, клиентский сертификат и ключ для входа, корневой сертификат EXC и пути к созданным секретам.
2. Разместите сертификаты и ключ на серверах компонентов (в примерах ниже это каталог `/opt/itc/itc/certs`).
3. Ограничьте доступ к закрытому ключу клиентского сертификата: владелец `itc-svc`, права 0600 (см. Ограничение прав доступа к файлам (see page 178)).
4. Настройте компоненты, как описано ниже, и перезапустите их службы.

6.5.4 Какие секреты можно получать из EXC

Компонент	Файл конфигурации	Секреты из EXC
Ядро ЦС	`minica.yml`, секция `ess`	Строка подключения к БД (`db-url`), пароль закрытого ключа ЦС (`ca-pass`), пароль доступа к БД (`db-pass`), пароль закрытого ключа подписи JWT (`jwt-pass`), пароль закрытого ключа TLS-сервера (`tls-pass`)
OCSP-ответчик mOCSP	`mocsp.yml`, секция `ess`	Строка подключения к БД (`db-url`)

Контрольная панель	`itc_api_miniCA_controlPanel_config.json` или `itc_api_miniCA_config.json`, секция `Ess`	Ключ подписи запросов к ЦС
Сервис публикации	`itc_api_miniCA_publication_config.json`, секция `Ess`	Пароли подключения к SCP и LDAP, ключ подписи запросов к ЦС
Архиватор	`itc_api_miniCA_archiver_config.json`, секция `Ess`	Пароли подключения к рабочей и архивной БД
Сервис резервного копирования	`itc_api_miniCA_config.json` или `itc_api_miniCA_backup_config.json` на сервере сервиса, секция `Ess`	Пароли БД ЦС и SSH узлов в заданиях резервного копирования

Кроме того, путь к секрету можно указать в формах Контрольной панели: в задачах публикации и заданиях резервного копирования (см. «Пароли в формах Контрольной панели» ниже).

Контрольная панель, сервис публикации, архиватор и сервис резервного копирования читают JSON-файлы из `/opt/itc/itc/config` по порядку: общий `itc_api_miniCA_config.json`, затем `itc_api_miniCA_config.runtime.json`, затем файлы сервиса; значения из следующего файла переопределяют предыдущие (см. «Конфигурационные файлы .NET-сервисов» в разделе Настройка вспомогательных сервисов (see page 139)). Поэтому секцию `Ess`, заданную в общем файле, используют все эти сервисы на сервере.

6.5.5 Секция ess ядра ЦС и mOCSP

Добавьте в конфигурацию секцию `ess` и перечислите в подсекции `secrets` пути к нужным секретам. При `enable: false` секреты из EXC не запрашиваются.

Секция `ess` в `/opt/itc/minica/conf/minica.yml`:

```
ess:
  enable: true                                    # включить получение секретов из EXC
  address: https://<адрес EXC>
  namespace: <пространство>                       # пространство Clearway CA в EXC
  tls-cert: /opt/itc/itc/certs/ess-tls.crt        # корневой сертификат EXC (проверка
сервера)
  client-cert: /opt/itc/itc/certs/ess-client.crt  # клиентский сертификат для входа в
EXC
```

```
  client-key: /opt/itc/itc/certs/ess-client.key   # закрытый ключ клиентского
сертификата
  secrets:
    db-url: "kv:<коллекция>:db-url"
    ca-pass: "kv:<коллекция>:ca-pass"
    db-pass: "kv:<коллекция>:db-pass"
    jwt-pass: "kv:<коллекция>:jwt-pass"
    tls-pass: "kv:<коллекция>:tls-pass"
```

Секция `ess` в `/opt/itc/ocsp/conf/mocsp.yml`:

```
ess:
  enable: true
  address: https://<адрес EXC>
  namespace: <пространство>
  tls-cert: /opt/itc/itc/certs/ess-tls.crt
  client-cert: /opt/itc/itc/certs/ess-client.crt
  client-key: /opt/itc/itc/certs/ess-client.key
  secrets:
    db-url: "kv:<коллекция>:db-url"
```

6.5.6 Контрольная панель, сервис публикации и архиватор

Добавьте в конфигурацию секцию `Ess` с адресом EXC, пространством и клиентским сертификатом. Для сервиса резервного копирования секция задаётся так же, в конфигурации на его сервере:

```
"Ess": {
  "Url": "https://<адрес EXC>",
  "Namespace": "<пространство>",
  "Auth": {
    "Cert": "/opt/itc/itc/certs/ess-client.crt",
    "Key": "/opt/itc/itc/certs/ess-client.key"
  }
}
```

Параметр	Назначение	Значение по умолчанию
`Ess.Url`	Адрес EXC	Не задано
`Ess.Namespace`	Пространство Clearway CA в EXC	Не задано
`Ess.Auth.Cert`	Клиентский сертификат для входа в EXC	Не задано

`Ess.Auth.Key`	Закрытый ключ клиентского сертификата	Не задано
`Ess.MountPoint`	Точка монтирования метода входа по сертификату в EXC	cert
`Ess.RoleName`	Роль для входа по сертификату	Не задана

Затем в каждом месте, где нужен секрет, вместо пароля или пути к файлу ключа укажите путь к секрету. В том же блоке включите получение секрета из EXC параметром `"Ess": true`.

Ключ подписи запросов к ЦС (Контрольная панель и сервис публикации) указывается в параметре `Sign` описания подключения к ЦС. Имя блока (в примере `root-ca`) служит наименованием подключения ЦС:

```
"MiniCa": {
  "root-ca": [
    {
      "Url": "https://<адрес ЦС>:9443",
      "Ess": true,
      "Auth": {
        "Tls": "/opt/itc/itc/certs/root-ca.crt",
        "Sign": "kv:<коллекция>:root-jwt-key",
        "Jwt": "/opt/itc/itc/certs/root-ca.jwt"
      }
    }
  ]
}
```

Пароли SCP и LDAP в задачах сервиса публикации указываются в параметре `Password` блоков `Scp` и `Ldap` задачи (файл `itc_api_miniCA_publication_config.json`):

```
"PublishingCrlTasks": [
  {
    "Key": "<ключ задачи>",
    "MiniCa": "<наименование подключения ЦС>",
    "IsDeltaCrl": false,
    "Format": "Der",
    "Cron": "0 * * * *",
    "Scp": {
      "Host": "<сервер SCP>",
      "Port": 22,
      "Username": "<учётная запись>",
      "Password": "kv:<коллекция>:scp-pass",
      "Path": "<каталог на сервере>/crl.crl",
      "Ess": true
    },
```

```
    "Ldap": {
      "Host": "<сервер LDAP>",
      "Port": 636,
      "SecureSocketLayer": true,
      "Username": "<учётная запись>",
      "Password": "kv:<коллекция>:ldap-pass",
      "AttributeName": "certificateRevocationList",
      "BaseDn": "<DN записи>",
      "Ess": true
    }
  }
]
```

В примере сервис публикации подключается к каталогу по LDAPS: параметр `SecureSocketLayer: true` соответствует флажку «Защищённое соединение» в форме задачи (см. «Остальные подключения»).

Пароли базы данных в задачах архиватора указываются в параметре `Password` строк подключения `Source` и `Target` (файл `itc_api_miniCA_archiver_config.json`):

```
{
  "Tasks": [
    {
      "Key": "ClearwayCA",
      "Cron": "@daily",
      "Source": "Host=<сервер
БД>;Port=5432;Database=cwica;Username=cwica;Password=kv:<коллекция>:db-pass",
      "Target": "Host=<сервер
БД>;Port=5432;Database=cwica_archive;Username=cwica;Password=kv:<коллекция>:archive-db-
pass",
      "Ess": true
    }
  ]
}
```

6.5.7 Пароли в формах Контрольной панели

В Контрольной панели путь к секрету можно указать вместо пароля в формах:

- задач публикации (переключатель «EXC» в блоках SCP и LDAP);
- заданий резервного копирования (переключатели «Пароль из EXC» для пароля БД ЦС и SSH-пароля узла).

Ссылки в задачах публикации получает сервис публикации, а ссылки в заданиях резервного копирования получает сервис резервного копирования в момент каждой операции. Поэтому секция `Ess` нужна в конфигурации того сервиса, который выполняет задание; секция `Ess` Контрольной панели для этого не используется. В файл задания сервис записывает только ссылку, пароль в нём не хранится.

Порядок заполнения форм описан в Руководстве пользователя.

6.5.8 Проверка после настройки EXC

После перезапуска убедитесь, что службы запустились и работают (см. раздел «Управление службами»). Если служба не запустилась, причину покажет её журнал:

```
systemctl --user status <юнит>
journalctl --user -u <юнит> -n 50
```

Затем удалите из конфигурационных файлов пароли, которые теперь получаются из EXC.

6.6 Хранение ключа ЦС на HSM/Rutoken (PKCS#11)

Редакция продукта

Функция недоступна в бесплатной редакции Clearway CA.

Хранение закрытого ключа ЦС в аппаратном модуле (HSM, Rutoken) повышает защищённость: ключ нельзя несанкционированно извлечь даже при физическом доступе к носителю. Все криптографические операции с ключом выполняет модуль через интерфейс PKCS#11.

Порядок настройки (подготовка носителя, параметры minica.yml и включение движка PKCS#11 в openssl.cnf) описан в подразделе «Хранение ключа ЦС на аппаратном носителе (PKCS#11)» раздела Настройка ядра ЦС (see page 126).

Меры защиты при хранении ключа на носителе:

- PIN носителя записывается в minica.yml в составе PKCS#11-URI, поэтому права на minica.yml должны быть 0600, а владельцем должна быть учётная запись `itc-svc` (см. Ограничение прав доступа к файлам (see page 178)).
- Не оставляйте копии minica.yml с PIN в общедоступных каталогах и в резервных копиях без защиты.
- Порядок замены ключа, хранящегося на носителе, и ограничения этого режима описаны в разделе «Ротация ключа ЦС» (подраздел «Ограничения механизма old-keys»).
- Ротация из Контрольной панели для ключа на носителе недоступна: ядро ЦС отвечает ошибкой (см. «Ротация ключа ЦС»).

Раздел «Аппаратные токены» Контрольной панели

Раздел «Аппаратные токены» не связан с хранением ключа ЦС: он работает с носителями пользователя локально в браузере. Эти операции не передаются в ЦС, не разграничиваются правами ЦС и не записываются в журнал событий ЦС. Удаление ключа на носителе не отзывает сертификат.

6.7 Шифрование данных сертификатов в БД

Механизм защищает данные сертификата при хранении в БД. Если в шаблоне сертификата включено шифрование, сертификат шифруется перед сохранением и хранится в БД в зашифрованном виде, а при скачивании автоматически расшифровывается. Это снижает риск раскрытия данных при прямом доступе к БД.

Шифрование включается в шаблоне, а не глобально: в карточке шаблона в блоке «Дополнительно» установите флажок «Шифровать данные в БД». Флажок недоступен, если у шаблона установлен «Не добавлять сгенерированный сертификат в базу данных».

Для шифрования используется отдельная ключевая пара, её задаёт секция `ca.encryption` файла `minica.yml`. Использовать для шифрования ключ подписи ЦС (`ca.key`, `ca.crt`) недопустимо.

```
ca:
  encryption:
    key: /opt/itc/minica/ca/encr.key    # закрытый ключ для расшифрования данных
    cert: /opt/itc/minica/ca/encr.crt   # сертификат для шифрования данных
```

Сертификатом (`ca.encryption.cert`) данные шифруются перед сохранением в БД, а закрытым ключом (`ca.encryption.key`) расшифровываются при чтении. Без закрытого ключа прочитать зашифрованные данные нельзя, поэтому включайте `encr.key` в резервные копии и храните их защищённо. Параметры секции описаны в разделе Настройка ядра ЦС (see page 126).

Производительность

Шифрование при записи и расшифрование при чтении могут снижать производительность.

6.8 Механизм KRA (резервное копирование и восстановление ключей)

KRA (Key Recovery Agent, агент восстановления ключей) позволяет хранить и восстанавливать закрытые ключи пользователей. ЦС шифрует закрытый ключ пользователя сертификатом KRA и хранит его в БД; восстановить ключ может только владелец закрытого ключа KRA.

Хранение ключа включается в шаблоне флажком «Сохранять приватный ключ, зашифрованный сертификатом KRA» (блок «Дополнительно» карточки шаблона). При запросе сертификата по такому шаблону вместе с CSR загружается файл закрытого ключа. В карточке выпущенного сертификата появляется кнопка «Скачать зашифрованный ключ».

В конфигурации ЦС указывается только сертификат KRA (файл `kra.crt`). Закрытый ключ KRA в продукте не хранится: он находится у ответственного лица, которое применяет его для расшифрования скачанного ключа.

Шаг 1. Подготовьте ключевую пару KRA вне серверов ЦС, на защищённом рабочем месте ответственного лица:

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:4096 -out kra.key
openssl req -new -key kra.key -out kra.csr -subj "/CN=KRA/O=MyOrg/C=RU"
```

Шаг 2. Выпустите сертификат KRA средствами ЦС по сформированному запросу `kra.csr`.

Шаг 3. Разместите сертификат на сервере ЦС в файле `/opt/itc/minica/ca/kra.crt` и укажите его в `minica.yml`:

```
ca:
  kra:
    cert: /opt/itc/minica/ca/kra.crt   # сертификат KRA (открытый ключ)
```

Шаг 4. Перезапустите службу ЦС: `systemctl --user restart minica.service`.

Закрытый ключ KRA

Закрытый ключ KRA (`kra.key`) служит единственным средством восстановления сохранённых ключей пользователей. Храните его вне ЦС в защищённом месте: его утрата делает восстановление невозможным.

6.9 Синхронизация времени и защита от повторов

ЦС проверяет метки времени (формат RFC 3339) в подписанных запросах клиентов. Если время клиента и время сервера расходятся больше допустимого, запрос отклоняется. Это защищает от повторного воспроизведения (replay) ранее перехваченных запросов.

Допустимое окно задаёт параметр `max-dt-sec` файла `minica.yml` (в секундах). Значение по умолчанию равно 600 с, то есть ±10 минут:

```
max-dt-sec: 600
```

Чтобы штатные запросы не отклонялись, включите синхронизацию времени по протоколу NTP с единым доверенным источником на всех узлах с компонентами Clearway CA и на сервере СУБД. Команды выполняются на каждом узле администратором с правом sudo:

```
sudo timedatectl set-ntp true
timedatectl status
```

В выводе `timedatectl status` должны быть строки `System clock synchronized: yes` и `NTP service: active`. Рассогласование часов между узлами и клиентами должно быть заведомо меньше `max-dt-sec`; при превышении обращения к системе отклоняются с ошибкой проверки времени.

6.10 Защита учётных записей и входа

Раздел описывает меры защиты входа в Контрольную панель: встроенную учётную запись OpenID, параметры клиента OpenID и предупреждение, которое пользователь видит при входе.

6.10.1 Встроенная учётная запись OpenID

Установщик создаёт в OpenID встроенную (статическую) учётную запись для первого входа. Её параметры задаются в группе «Статическая учётная запись» компонента OpenID: email (по умолчанию `admin@company.ru`), имя (по умолчанию `admin`) и пароль. Учётная запись получает права администратора Контрольной панели и сохраняется после подключения каталога LDAP. В файле `/opt/itc/openid/config.yaml` её задают параметр `enablePasswordDB: true` и секция `staticPasswords`.

- При установке задайте для встроенной учётной записи стойкий пароль.
- Пока встроенная учётная запись используется, сверьте её права с перечнем прав версии 26.3.0 (см. «Права встроенной учётной записи» в разделе Назначение ролей через группы AD/LDAP (see page 196)).
- После подключения каталога LDAP и проверки входа администратора из каталога отключите вход по встроенным учётным записям.

Перед отключением

> Отключайте встроенную учётную запись только после того, как администратор ЦС успешно вошёл в Контрольную панель учётной записью из каталога и получил нужные права. После отключения войти без каталога нельзя. На время перезапуска OpenID вход в Контрольную панель недоступен.

Все шаги выполняются на сервере OpenID в сеансе технологической учётной записи `itc-svc`.

Шаг 1. Сохраните копию файла конфигурации с сохранением прав. Копия содержит пароли, её удаляют в шаге 5.

```
cp -p /opt/itc/openid/config.yaml /opt/itc/openid/config.yaml.bak
```

Шаг 2. В файле `/opt/itc/openid/config.yaml` задайте `enablePasswordDB: false` и удалите всю секцию `staticPasswords`: от строки `staticPasswords:` до следующей секции верхнего уровня (например, `expiry:`). Сервис OpenID построен на Dex, а Dex не запускается, если при `enablePasswordDB: false` в файле остались встроенные учётные записи.

```
enablePasswordDB: false
# секция staticPasswords удалена целиком
```

Шаг 3. Перезапустите службу OpenID и через 10–15 секунд проверьте её состояние. Ожидается `active (running)`.

```
systemctl --user restart openid.service
systemctl --user status openid.service
```

Если служба не запустилась, причину покажет журнал. Исправьте файл или верните копию из шага 1 и повторите шаг 3:

```
journalctl --user -u openid.service -n 50
cp -p /opt/itc/openid/config.yaml.bak /opt/itc/openid/config.yaml
```

Шаг 4. Войдите в Контрольную панель учётной записью администратора из каталога. Вход встроенной учётной записью должен быть недоступен.

Шаг 5. Удалите копию файла:

```
rm /opt/itc/openid/config.yaml.bak
```

6.10.2 Параметры клиента OpenID Контрольной панели

Контрольная панель входит в OpenID от имени клиента OpenID (по умолчанию `itc-clearway`). Параметры клиента для браузера хранятся в файле `/opt/itc/itc/env/client_env.js`, а Nginx отдаёт этот файл любому посетителю по адресу `https://<адрес Контрольной панели>/assets/env/client_env.js`. Поэтому при установке компонента «Контрольная панель» оставьте поле «Секрет клиента OpenID» пустым.

Проверка на сервере Контрольной панели:

```
grep -n "secret" /opt/itc/itc/env/client_env.js
```

Ожидается строка `secret: null`. Затем откройте Контрольную панель в браузере и убедитесь, что вход выполняется.

Параметр `web.allowedOrigins` в `/opt/itc/openid/config.yaml` задаёт адреса веб-приложений, которым разрешено обращаться к OpenID из браузера. После установки задано `['*']`, то есть любые адреса. Ограничьте список адресом Контрольной панели. Измените только этот параметр: остальные параметры секции `web` (адрес, сертификат и ключ TLS) оставьте без изменений.

```
web:
  # остальные параметры секции не меняются
  allowedOrigins: ['https://<адрес Контрольной панели>']
```

После правки перезапустите `openid.service` на сервере OpenID от имени `itc-svc`, проверьте состояние службы (`systemctl --user status openid.service`) и вход в Контрольную панель.

6.10.3 Предупреждение при входе

После входа Контрольная панель один раз за сеанс показывает окно «Успешная авторизация» с предупреждением: «Предупреждение! В ИС реализованы меры по защите информации, пользователю необходимо соблюдать меры обращения с информацией установленные оператором». Пользователь подтверждает ознакомление кнопкой «Подтвердить».

В окно можно добавить дату и время последнего входа пользователя. Текст предупреждения фиксирован.

Окно настраивается в файле `/opt/itc/itc/env/config.json` на сервере Контрольной панели:

Параметр	Назначение	Значение после установки

`auth__modal_is_success`	Показывать после входа окно «Успешная авторизация» с предупреждением	true
`auth__lastSession`	Выводить в окне строку «Дата последнего входа»	true

Для безопасного состояния оставьте `auth__modal_is_success: true`: окно с предупреждением входит в обязательные меры защиты информации (см. «Вход в систему» в Руководстве пользователя).

Файл читает веб-интерфейс при загрузке страницы, поэтому изменения действуют после повторной загрузки страницы Контрольной панели в браузере; перезапуск служб не нужен.

6.11 Разграничение доступа

Безопасное состояние обязательно включает ограничение полномочий пользователей и сервисов. Доступ к операциям ЦС разграничивается по ролям: пользователи получают права через членство в группах каталога, а сервисные компоненты обращаются к ЦС по JWT с ограниченным набором функций API.

- Модель прав, стандартные роли и матрица доступа описаны в главе 4 Описание ролевой модели (see page 47).
- Назначение ролей группам каталога описано в разделе Назначение ролей через группы AD/LDAP (see page 196).
- Сервисные роли JWT и их состав описаны в подразделе «Сервисные роли JWT (ролевая модель)» раздела Настройка ядра ЦС (see page 126).

6.11.1 Минимальные права сервисов

Установщик по умолчанию выпускает JWT с ролью admin (полный доступ к API ЦС) для Контрольной панели, сервиса публикации, SCEP-адаптера и MS-WSTEP. На шаге «Выпуск JWT» выберите для каждого компонента роль с минимально достаточным набором прав: поле «Идентификатор роли администратора» в дополнительных настройках шага. На том же шаге заполните поле «Парольная фраза для шифрования ключа»: если оставить его пустым, ключ выпускается без шифрования. Расшифрованные ключи подписи запросов к ЦС храните только в `/opt/itc/itc/certs` с правами 0600.

7 Типичные операции

7.1 Управление доступом

Раздел описывает, как выдать пользователям права Clearway CA через группы каталога и как включить выпуск сертификатов с одобрением. Перечень прав, стандартные роли и матрица доступа приведены в главе Описание ролевой модели (see page 47).

7.1.1 Назначение ролей через группы AD/LDAP

Права выдаются через группы каталога: сервис аутентификации OpenID сопоставляет группы пользователя с наборами прав по карте `ldapGroupRolesMap` (как права доходят до пользователя, см. главу Описание ролевой модели (see page 47)).

Где находится карта:

- файл: /opt/itc/openid/config.yaml на сервере OpenID (при установке в каталог по умолчанию);
- место в файле: секция `connectors`, коннектор LDAP (`type: ldap`), блок `config` → `ldapGroupRolesMap`;
- служба: openid.service, пользовательский юнит технологической учётной записи (по умолчанию itc-svc).

Коннектор LDAP вместе с картой добавляет в config.yaml установщик на шаге «Интеграция с LDAP» группы «Аутентификация (OpenID)». Если этот шаг пропущен, коннектора и карты в файле нет: их добавляют вручную по примеру config.yaml из раздела «Развертывание сервиса аутентификации OpenID». Карта из поставки сопоставляет стандартным ролям три группы каталога:

- CWCA_Administrator соответствует роли «Администратор ЦС» с полным набором прав;
- CWCA_Auditor соответствует роли «Аудитор» с шестью правами только на просмотр;
- CWCA_User соответствует роли «Пользователь»: просмотр сертификатов и шаблонов, создание запросов на выпуск.

Имена групп и состав прав можно изменить в карте. Имя группы в карте должно совпадать с именем группы в каталоге, то есть со значением атрибута, заданного параметром `groupSearch.nameAttr` коннектора (по умолчанию `cn`).

В версии 26.3.0 прав 17. Четыре из них (`rleMiniCABackup-r`, `rleMiniCABackup-w`, `rleMiniCABackupRestore-w` и `rleMiniCARotation-w`) открывают разделы «Задачи → Резервное копирование» и «Настройки → Ротация ключей». Проверьте, что у группы администраторов ЦС в карте есть все 17 прав, как в примере ниже. Пользователь, которому эти права не сопоставлены, указанных разделов не увидит.

7.1.1.1 Как назначить роль пользователю

Включите учётную запись пользователя в нужную группу каталога средствами AD/LDAP. Файлы Clearway CA при этом не меняются.

OpenID получает группы пользователя из каталога при входе (параметр `groupSearch.onDemand`, по умолчанию включён) и при периодической синхронизации (параметр `import.timer`, по умолчанию `1h`, то есть раз в час). Права передаются в токене, который выдаётся при входе. Поэтому новые права начинают действовать после того, как пользователь выйдет из Контрольной панели и войдёт снова.

7.1.1.2 Пример карты групп для 26.3.0

Фрагмент config.yaml: группе администраторов ЦС сопоставлены все 17 прав, права версии 26.3.0 отмечены комментарием. Параметры подключения к каталогу и поиска пользователей и групп пропущены. Отступы такие же, как в файле, который создаёт установщик: строки списка `roles` начинаются в той же колонке, что и ключ `roles:`.

```
connectors:
  - type: ldap
    id: ldap
    name: LDAP
    config:
      # ... параметры подключения, import, userSearch, groupSearch ...
      ldapGroupRolesMap:
        - ldapGroup: CWCA_Administrator   # Администратор ЦС: все 17 прав
          roles:
          - rleMiniCACertList-r
          - rleMiniCAEventList-r
          - rleMiniCATemplList-r
          - rleMiniCAReqList-r
          - rleMiniCACRL-r
          - rleMiniCAConfig-r
          - rleMiniCACertReq-w
          - rleMiniCACertRev-w
          - rleMiniCACertUnRev-w
          - rleMiniCACRL-w
          - rleMiniCAConfig-w
          - rleMiniCATempl-w
          - rleMiniCAReqApprove-w
          - rleMiniCABackup-r            # 26.3.0: просмотр заданий и копий
          - rleMiniCABackup-w            # 26.3.0: запуск копирования
          - rleMiniCABackupRestore-w     # 26.3.0: восстановление из копии
          - rleMiniCARotation-w          # 26.3.0: ротация ключей ЦС
        - ldapGroup: CWCA_Auditor         # Аудитор: 6 прав на просмотр
          roles:
          - rleMiniCACertList-r
          - rleMiniCAEventList-r
          - rleMiniCATemplList-r
          - rleMiniCAReqList-r
          - rleMiniCACRL-r
```

```
    - rleMiniCAConfig-r
  - ldapGroup: CWCA_User            # Пользователь: 3 права
    roles:
    - rleMiniCACertList-r
    - rleMiniCATemplList-r
    - rleMiniCACertReq-w
```

Составляя собственные роли, сверьтесь с подразделом «Права и пункты меню Контрольной панели» главы Описание ролевой модели (see page 47): некоторые пункты меню отображаются только при сочетании нескольких прав.

7.1.1.3 Изменение карты групп

Изменение нужно, чтобы использовать другие имена групп каталога, изменить состав прав роли или добавить роль.

Перед изменением карты

Карта определяет права всех пользователей перечисленных групп. Перед правкой убедитесь, что имена групп в карте совпадают с фактическими группами каталога: если имя не совпадёт, пользователи этой группы войдут без прав.

Все команды выполняются на сервере OpenID в сеансе технологической учётной записи (по умолчанию itc-svc). Пути приведены для установки в каталог по умолчанию /opt/itc.

Шаг 1. Войдите в сеанс технологической учётной записи (например, из сеанса root) и сохраните копию файла с сохранением прав. Файл содержит пароли, поэтому копию удалите в шаге 6 (требования к правам на файлы приведены в разделе Ограничение прав доступа к файлам (see page 178)).

```
su - itc-svc
cp -p /opt/itc/openid/config.yaml /opt/itc/openid/config.yaml.bak
```

Шаг 2. Отредактируйте блок `ldapGroupRolesMap` в /opt/itc/openid/config.yaml. Идентификаторы прав берите из перечня в главе Описание ролевой модели (see page 47). Добавляйте каждое право отдельной строкой с тем же отступом, что у соседних строк списка `roles`. Строку с большим отступом YAML объединит с предыдущей в одно значение, и оба права пропадут; при других ошибках в структуре файла служба может не запуститься.

Шаг 3. Перезапустите службу. На время перезапуска вход в Контрольную панель недоступен.

```
systemctl --user restart openid.service
```

Шаг 4. Через 10–15 секунд после перезапуска проверьте, что служба работает. Команда должна вывести `active`.

```
systemctl --user is-active openid.service
```

Вывод `activating` или `failed` означает, что служба не запустилась. Сразу после перезапуска команда может вывести `active`, даже если служба затем завершится с ошибкой: после сбоя юнит openid.service запускает её повторно через 10 секунд (параметры `Restart=always` и `RestartSec=10`).

Если выведено не `active`, просмотрите журнал службы:

```
journalctl --user -u openid.service -n 50
```

Исправьте файл или верните копию из шага 1 и повторите шаги 3–4:

```
cp -p /opt/itc/openid/config.yaml.bak /opt/itc/openid/config.yaml
```

Шаг 5. Попросите пользователя выйти из Контрольной панели и войти снова. Проверить права можно в профиле: разверните левое меню, щёлкните имя пользователя и выберите «Профиль». Назначенные права перечислены в блоке «Права».

Шаг 6. Убедившись, что права применились, удалите копию файла:

```
rm /opt/itc/openid/config.yaml.bak
```

7.1.1.4 Права встроенной учётной записи

При установке сервиса OpenID установщик создаёт в config.yaml встроенную учётную запись для входа без каталога (параметр `enablePasswordDB: true`, секция `staticPasswords`). Карта групп на неё не действует: права встроенной учётной записи перечислены в её списке `customClaims.roles`. Если такая учётная запись есть, сверьте список с перечнем прав версии 26.3.0 и при необходимости дополните его. Для этого выполните шаги 1–6 процедуры «Изменение карты групп», в шаге 2 дополнив список `customClaims.roles`.

```
staticPasswords:
  - email: "<адрес для входа>"
    # ... hash, userID, username ...
    customClaims:
```

```
roles:
  - rleMiniCACertList-r
  # ... остальные права ...
  - rleMiniCARotation-w
```

7.1.2 Выпуск сертификатов с одобрением

Для отдельных шаблонов можно включить обязательное одобрение выпуска. Тогда запрос по такому шаблону не выпускается сразу, а ждёт решения ответственного лица. В карточке шаблона признак называется «Создавать запрос на выпуск» (блок «Дополнительно»), а в базе данных ЦС ему соответствует `requires_approval`.

Порядок работы:

1. Пользователь с правом `rleMiniCATempl-w` включает у шаблона флажок «Создавать запрос на выпуск».
2. Пользователь с правом `rleMiniCACertReq-w` создаёт запрос по этому шаблону. Запрос получает статус Pending и ждёт решения.
3. Пользователь с правом `rleMiniCAReqApprove-w` одобряет или отклоняет запрос в разделе «Запросы», по одному или группой. При отклонении нужно указать причину.
4. Сертификат выпускается только после одобрения: запрос получает статус Satisfied, а при ошибке выпуска статус Erroneous. Отклонённый запрос получает статус Rejected.
5. Решения фиксируются в журнале событий: одобрению соответствует событие CA_APPROVE, а отклонению событие CA_REJECT.

Кто одобряет запросы

В карте групп из поставки право `rleMiniCAReqApprove-w` есть только у группы администраторов ЦС (CWCA_Administrator). Чтобы запросы одобряла отдельная роль, добавьте это право её группе в карте `ldapGroupRolesMap` (см. Назначение ролей через группы AD/LDAP (see page 196)). Чтобы видеть раздел «Запросы», такой роли нужно также право `rleMiniCAReqList-r`.

7.2 Добавление сертификата ЦС в доверенные

Компьютер доверяет сертификату сервиса, только если корневой сертификат его цепочки находится в хранилище доверенных сертификатов операционной системы. Пока корневой сертификат Clearway CA туда не добавлен, браузер предупреждает о небезопасном подключении, а программы, проверяющие сертификат (например, curl), прерывают TLS-соединение.

Поэтому корневой сертификат цепочки Clearway CA добавляют в доверенные на всех серверах и рабочих местах (АРМ), где проверяются сертификаты, выпущенные центром сертификации.

Раздел описывает, какой сертификат добавлять, где его взять и как установить его в Linux и Windows.

7.2.1 Какой сертификат добавлять

- В доверенные добавляют только корневой сертификат цепочки Clearway CA (сертификат корневого ЦС).
- Сертификат корневого ЦС самоподписанный: в нём поля «Кому выдан» (Subject) и «Кем выдан» (Issuer) совпадают.
- Сертификат выдающего ЦС в доверенные корневые не добавляют. Он нужен для построения цепочки и должен быть доступен по адресу AIA, то есть по адресу сертификата издателя, который записан в расширении AIA выпущенных сертификатов (см. раздел «Точка распространения CRL и сертификатов (CDP)»).
- На Windows сертификат выдающего ЦС можно дополнительно установить в хранилище «Промежуточные центры сертификации».

Где взять сертификат корневого ЦС:

Источник	**Файл или адрес**
Сервер корневого ЦС	/opt/itc/minica/ca/ca.crt (параметр `ca.cert` в minica.yml), формат PEM. Скопируйте файл по SSH, например с помощью WinSCP
Установщик	Файл root-ca.crt (результат «Сертификат» компонента корневого ЦС) скачивается по завершении установки этого компонента. Не путайте его с root-ca-tls.crt (результат «Сертификат TLS»): это TLS-сертификат сервера ЦС
Контрольная панель	«Настройки → Конфигурация» → «Действия» → «Скачать цепочку сертификатов» (право `rleMiniCAConfig-r`). Файл содержит цепочку текущего ЦС. В доверенные корневые добавляют только сертификат корневого ЦС из неё (у него Subject и Issuer совпадают), а не весь файл; перед добавлением сверьте отпечаток этого сертификата (см. «Проверка подлинности файла»)
Точка распространения (AIA)	`http://<fqdn>/aia/<CN>.crt`, где `<fqdn>` означает имя сервера CDP, `<CN>` означает общее имя корневого ЦС (пробелы в нём заменяются знаком «_»). Адрес доступен, если файл сертификата размещён в каталоге /opt/itc/web/aia/ сервера CDP

> **Закрытый ключ не нужен**
>
> Для доверия нужен только сертификат. Файл закрытого ключа ЦС (/opt/itc/minica/ca/ca.key) не копируйте и не передавайте на другие компьютеры.

7.2.2 Проверка подлинности файла

Перед добавлением убедитесь, что файл не подменён: его отпечаток SHA-256 должен совпадать с отпечатком сертификата на сервере корневого ЦС.

Шаг 1. На сервере корневого ЦС в сеансе учётной записи itc-svc выведите субъект, издателя и отпечаток сертификата:

```
openssl x509 -in /opt/itc/minica/ca/ca.crt -noout -subject -issuer -fingerprint -sha256
```

Отпечаток выводится в строке `SHA256 Fingerprint=` в виде пар шестнадцатеричных цифр через двоеточие (в разных версиях OpenSSL имя алгоритма в этой строке может быть записано строчными буквами).

Шаг 2. Проверьте скопированный файл на компьютере, где будете его устанавливать:

- в Linux той же командой `openssl x509`, указав путь к скопированному файлу;
- в Windows командой `certutil -dump root-ca.crt`: отпечаток выводится в строке «Хеш сертификата(sha256)» без двоеточий и строчными буквами.

Если отпечатки различаются, сертификат не добавляйте.

7.2.3 Linux

Команды выполняются на каждом сервере или АРМ от имени пользователя с правами sudo. Сертификат должен быть в формате PEM: файл начинается строкой `-----BEGIN CERTIFICATE-----`. Файл в двоичном формате DER преобразуйте в PEM:

```
openssl x509 -inform DER -in root-ca.cer -out root-ca.crt
```

Astra Linux

```
sudo cp root-ca.crt /usr/local/share/ca-certificates/clearway-root-ca.crt
sudo update-ca-certificates
```

Имя файла в каталоге /usr/local/share/ca-certificates должно оканчиваться на .crt, иначе команда его пропустит. При успешном добавлении команда сообщает `1 added`.

Альт СП, РЕД ОС

```
sudo cp root-ca.crt /etc/pki/ca-trust/source/anchors/clearway-root-ca.crt
sudo update-ca-trust extract
```

При успешном выполнении команда `update-ca-trust extract` ничего не выводит.

Проверка. Обратитесь к сервису, сертификат которого выпущен Clearway CA, например к Контрольной панели. В адресе укажите имя сервера (FQDN) так, как оно записано в сертификате сервиса:

```
curl -sS -o /dev/null -w "%{http_code}\n" https://<fqdn>/
```

Если сертификат добавлен, команда выводит код HTTP-ответа (например, `200`). Если нет, она завершается ошибкой `curl: (60) SSL certificate problem: unable to get local issuer certificate`.

Хранилище браузера

Браузеры в Linux могут использовать собственное хранилище сертификатов. Если после добавления сертификата в систему браузер по-прежнему предупреждает о небезопасном подключении, импортируйте корневой сертификат в настройках браузера (раздел управления сертификатами, вкладка центров сертификации) и разрешите доверять ему для идентификации сайтов.

7.2.4 Windows

Действия выполняются учётной записью с правами локального администратора. Сертификат устанавливается в хранилище компьютера («Локальный компьютер»): тогда ему доверяют все пользователи и службы этого компьютера. Windows принимает файлы в форматах PEM и DER с расширением .crt или .cer.

Мастер импорта сертификатов

1. Дважды щёлкните файл root-ca.crt и в окне «Сертификат» нажмите «Установить сертификат…».
2. Выберите расположение хранилища «Локальный компьютер», нажмите «Далее» и подтвердите запрос контроля учётных записей.
3. Выберите «Поместить все сертификаты в следующее хранилище», нажмите «Обзор…», выберите «Доверенные корневые центры сертификации» и нажмите «ОК», затем «Далее».

4. Нажмите «Готово». Мастер сообщает «Импорт успешно выполнен».

Командная строка

Запустите командную строку или PowerShell от имени администратора и выполните одну из команд, заменив путь к файлу своим:

```
# certutil: в командной строке или PowerShell
certutil -addstore -f Root C:\Temp\root-ca.crt
# командлет PowerShell
Import-Certificate -FilePath C:\Temp\root-ca.crt -CertStoreLocation Cert:
\LocalMachine\Root
```

Сертификат выдающего ЦС при необходимости устанавливается в промежуточные: `certutil -addstore -f CA C:\Temp\sub-ca.crt`.

Групповая политика

Способ подходит для компьютеров домена.

1. В консоли управления групповой политикой откройте политику, которая применяется к нужным компьютерам.
2. Перейдите в раздел «Конфигурация компьютера → Политики → Конфигурация Windows → Параметры безопасности → Политики открытого ключа → Доверенные корневые центры сертификации».
3. В контекстном меню выберите «Импорт...» и укажите файл сертификата.
4. Компьютеры получат сертификат при следующем обновлении политик; чтобы применить политику сразу, выполните на компьютере `gpupdate /force`.

Проверка. Откройте оснастку certlm.msc, раздел «Доверенные корневые центры сертификации → Сертификаты»: в списке есть сертификат корневого ЦС (поле «Кому выдан»). Затем перезапустите браузер и откройте адрес сервиса с сертификатом Clearway CA, например Контрольной панели: страница открывается без предупреждения о небезопасном подключении.

7.2.5 Типовые проблемы

Что видно	Причина	Что сделать
После добавления по-прежнему `unable to get local issuer certificate` или предупреждение браузера	Добавлен не корневой сертификат (например, сертификат выдающего ЦС) или не выполнена команда обновления хранилища	Проверьте, что Subject и Issuer сертификата совпадают, и повторите `update-ca-certificates` или `update-ca-trust extract`

Astra Linux: `update-ca-certificates` сообщает `0 added`	Имя файла не оканчивается на .crt или файл не в формате PEM	Переименуйте файл или преобразуйте его в PEM
`curl: (60) SSL: no alternative certificate subject name matches target host name`	Имя в адресе не совпадает с именем в сертификате сервиса. Доверие к ЦС при этом настроено верно	Обращайтесь к сервису по имени, указанному в его сертификате
Windows: сертификат виден в certmgr.msc, но службы ему не доверяют	Сертификат установлен в хранилище текущего пользователя	Установите сертификат в хранилище «Локальный компьютер»

После ротации ключа ЦС добавьте в доверенные новый сертификат корневого ЦС (см. раздел «Ротация ключа ЦС»). Прежний сертификат не удаляйте, пока действуют выпущенные им сертификаты.

7.3 Ротация ключа ЦС

Ротацией ключа ЦС называют замену сертификата центра сертификации на действующем ключе или вместе с ключевой парой. После ротации ЦС продолжает обслуживать и новые, и ранее выпущенные сертификаты. Раздел описывает, как выполнить ротацию из Контрольной панели или вручную, как сохранить прежнюю ключевую пару, настроить ядро ЦС и OCSP-ответчик mOCSP для одновременной работы с прежним и новым ЦС и как проверить результат.

После ротации необходимо обеспечить:

- отзыв сертификатов, выпущенных прежним ЦС;
- выпуск CRL и Delta CRL для этих сертификатов;
- ответы на OCSP-запросы по сертификатам как прежнего, так и нового ЦС.

7.3.1 Что меняется при ротации

Связь сертификата с ключом ЦС, которым он выпущен, определяют два идентификатора:

Идентификатор	**Что обозначает**
SKI (Subject Key Identifier)	Идентификатор ключа владельца сертификата. У сертификата ЦС это идентификатор ключа самого ЦС
AKI (Authority Key Identifier)	Идентификатор ключа ЦС, которым выпущен сертификат. AKI сертификата равен SKI выпустившего его ЦС

Дальнейшие действия зависят от того, менялась ли ключевая пара ЦС:

Что заменено	SKI сертификата ЦС	Что нужно для ранее выпущенных сертификатов
Только сертификат ЦС, ключевая пара прежняя (в Контрольной панели режим «Продление»)	Не меняется	Прежние и новые сертификаты относятся к одному ключу ЦС. Каталог `ca.old-keys` настраивать не нужно
Ключевая пара и сертификат ЦС (режим «Перевыпуск» или ручная замена)	Новый	Прежняя пара должна быть в каталоге `ca.old-keys`: при «Перевыпуске» её сохраняет ядро ЦС, при ручной замене администратор. Для нового ЦС нужен отдельный экземпляр mOCSP (см. «Ротация с новой ключевой парой»)

Сертификат ЦС с новой ключевой парой получает новый SKI. Фактически это новый ЦС, даже если Subject и Common Name остались прежними. Образуются два поколения сертификатов:

- прежние сертификаты → AKI прежнего ЦС → прежний ключ ЦС;
- новые сертификаты → AKI нового ЦС → новый ключ ЦС.

Прежние сертификаты не перестают действовать после ротации. Пока их срок не истёк, поддерживайте их отзыв, выпуск CRL и ответы OCSP.

7.3.2 Способы ротации

Способ	Что делает администратор	Что нужно дальше
Контрольная панель, режим «Продление»	Запрашивает у ядра ЦС CSR на действующем ключе, получает по нему сертификат у вышестоящего ЦС и загружает его	Каталог old-keys и второй экземпляр mOCSP не нужны
Контрольная панель, режим «Перевыпуск»	Запрашивает у ядра ЦС CSR на новой ключевой паре, которую создаёт ядро, указывает причину («Плановая замена» или «Компрометация»), получает по CSR сертификат и загружает его	Отдельный экземпляр mOCSP и проверки (см. «Ротация с новой ключевой парой»). Прежнюю пару в old-keys сохраняет ядро ЦС
Вручную на сервере ЦС	Заменяет файлы ключа и сертификата ЦС, подготовленные заранее, и перезапускает службу minica	Прежняя пара в old-keys, отдельный экземпляр mOCSP, проверки (см. «Ротация с новой ключевой парой»)

Ротация из Контрольной панели доступна, только если ключ ЦС хранится в файле. Для ЦС с ключом в КриптоПро CSP или на носителе PKCS#11 (HSM, Rutoken) ядро ЦС отклоняет формирование CSR и установку сертификата (см. «Ограничения механизма old-keys»).

mOCSP сопоставляет запросы с ЦС по хешам имени и ключа издателя (см. «Как mOCSP обслуживает прежний и новый ЦС»). При «Продлении» ключ ЦС прежний, и если Subject сертификата ЦС не изменился, эти хеши тоже остаются прежними. Поэтому действующий экземпляр mOCSP должен отвечать по всем сертификатам; проверьте это после замены (шаг 7 раздела «Ротация из Контрольной панели»).

7.3.3 Ротация из Контрольной панели

В Контрольной панели ротация выполняется на странице «Настройки → Ротация ключей». По команде со страницы ядро ЦС формирует запрос на сертификат (CSR), администратор получает по нему сертификат у ЦС, который выпускает сертификат для этого ЦС, и загружает его на той же странице. Работа со страницей подробно описана в Руководстве пользователя (раздел «Ротация ключей»), а здесь описано, что подготовить администратору и как связать ротацию с настройкой прежних ключей и mOCSP.

Страница обращается к API Контрольной панели `/api/minica/rotation/*`. Контрольная панель ротацию сама не выполняет: она передаёт запросы ядру ЦС, выбранного в карточке «Выбор Clearway CA», и показывает его ответы. Ключевую пару генерирует и хранит ядро ЦС, Контрольная панель получает от него только CSR. На сервер ЦС Контрольная панель не заходит.

Действие на странице	Что делает Контрольная панель	Что делает ядро ЦС
Открытие страницы	Запрашивает у ядра ЦС состояние ротации	Возвращает сведения о действующем сертификате ЦС и состояние ротации
Сформировать в окне формирования CSR	Передаёт ядру режим, а при «Перевыпуске» также тип и размер ключа, если они выбраны. Записывает в свой журнал режим, причину и имя пользователя	При «Перевыпуске» создаёт новую ключевую пару и хранит её в каталоге `ca.new-keys` до установки сертификата. Формирует CSR и возвращает его
Выбор файла в окне «Заменить сертификат»	Разбирает файл и показывает поля сертификата. Сертификат не проверяет	Не участвует
Заменить	Передаёт сертификат ядру ЦС и показывает результат	Проверяет сертификат (см. шаг 5), сохраняет прежнюю пару и устанавливает новый сертификат

Состояние ротации и ожидающий CSR ядро хранит в БД ЦС, поэтому они сохраняются при перезапуске службы. Состояния: ротации нет (`NONE`), CSR ожидает сертификата (`PENDING`),

ротация завершена (`COMPLETED`), запрос заменён новым (`SUPERSEDED`). Ожидание сертификата страница показывает плашкой «CSR сформирован, ожидается загрузка сертификата», завершённую ротацию плашкой «Замена сертификата выполнена».

При установке сертификата ядро ЦС само выполняет действия, которые при ручной ротации выполняет администратор:

- копирует прежний сертификат ЦС в каталог `ca.old-keys` под именем `<AKI>.crt` , а при новой ключевой паре и прежний ключ под именем `<AKI>.key` (как ядро выбирает эти файлы, см. «Выбор ключа по значению AKI»);
- заменяет файл сертификата ЦС ca.crt, а при новой паре и файл ключа ca.key: новый ключ переносится из каталога `ca.new-keys` и защищается так же, как действующий (параметр `ca-pass`);
- перестраивает файл цепочки (параметр `chain-file`): прежний сертификат ЦС в нём заменяется новым, сертификаты вышестоящих ЦС и их порядок сохраняются, копия прежней цепочки остаётся в каталоге `ca.old-keys` ;
- начинает работать с новым сертификатом и ключом без перезапуска службы minica.

OCSP-ответчик mOCSP, публикацию сертификата ЦС во внешних точках (LDAP, CDP/AIA) и доверенные хранилища ядро не обновляет: это отдельные действия администратора (см. шаг 7).

Что нужно:

- право `rleMiniCARotation-w` : без него пункт «Ротация ключей» не отображается, а страница не открывается и по прямой ссылке. Это право нужно для всех действий страницы, включая просмотр состояния ротации: отдельного права на просмотр нет;
- право сопоставлено группе пользователя в карте `ldapGroupRolesMap` сервиса OpenID, а для встроенной учётной записи указано в её списке `customClaims.roles` (см. Назначение ролей через группы AD/LDAP (see page 196));
- после изменения прав перезапустите сервис OpenID, затем выйдите из Контрольной панели и войдите снова: права передаются в токене при входе;
- ЦС выбран в карточке «Выбор Clearway CA» на странице «Настройки → Конфигурация» (право `rleMiniCAConfig-r`); выбор хранится в браузере пользователя;
- ключ ЦС хранится в файле, а не в КриптоПро CSP и не на носителе PKCS#11;
- доступ к вышестоящему ЦС, который подпишет CSR (для выдающего ЦС нужен доступ к корневому ЦС).

Шаг 1. Проверьте, для какого ЦС выполняется ротация. Откройте «Настройки → Ротация ключей». В карточке «Действующий сертификат ЦС» убедитесь, что показан сертификат нужного ЦС: субъект, издатель, серийный номер, срок действия, идентификатор ключа (SKI) и отпечаток SHA-256. Запишите серийный номер и SKI: по ним вы проверите результат.

Шаг 2. Для «Перевыпуска» подготовьтесь к смене ключа. До формирования CSR выполните шаги 1.1, 1.2 и 1.4–1.6 этапа 1 раздела «Ротация с новой ключевой парой»: резервные копии, определение SKI прежнего ЦС, проверка комплекта OCSP и адреса OCSP, выпуск тестового

сертификата прежним ЦС. Шаг 1.3 пропустите: прежнюю пару в каталог old-keys ядро ЦС сохранит само при установке нового сертификата. Для «Продления» шаг пропустите.

Шаг 3. Сформируйте CSR. Нажмите **Пересоздать CSR**, заполните поля и нажмите **Сформировать**.

Поле	Что указать
Режим	«Продление» (по умолчанию): новый сертификат для действующего ключа ЦС, ключевая пара и SKI не меняются, тип и размер ключа ядру ЦС не передаются. «Перевыпуск»: ядро ЦС создаёт новую ключевую пару и формирует CSR на ней
Причина перевыпуска	Только для «Перевыпуска», обязательно: «Плановая замена» или «Компрометация»
Тип ключа	Только для «Перевыпуска»: «Как у текущего ключа» (по умолчанию), RSA, ECDSA, Ed25519, ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012. При значении «Как у текущего ключа» ядро ЦС берёт тип и размер действующего ключа
Размер ключа	Для всех типов, кроме «Как у текущего ключа» и Ed25519. Выберите размер явно: для RSA не менее 2048 бит, для ключа ЦС рекомендуется 4096. Допустимость типа и размера проверяет ядро ЦС

Subject запроса и срок действия нового сертификата на странице не задаются: Subject и расширения ядро ЦС копирует в CSR из действующего сертификата ЦС, а срок действия определяет ЦС, который выпускает сертификат.

Результат: появляется сообщение «CSR сформирован», на странице отображается плашка «CSR сформирован, ожидается загрузка сертификата», становятся доступны кнопки **Скачать CSR** и **Заменить сертификат**.

Если CSR уже ожидает сертификата, новый запрос заменяет ожидающий: ключ прежнего запроса больше не используется, и сертификат, выпущенный по прежнему CSR, установить нельзя. Пересоздавайте CSR, только если сертификат по прежнему запросу не нужен, и передавайте вышестоящему ЦС последний сформированный CSR.

Причина перевыпуска ядру ЦС не передаётся: она попадает только в журнал Контрольной панели (каталог `/opt/itc/itc/logs/itc_api_miniCA_controlPanel/`, см. Журналирование и мониторинг (see page 256)). Прежний сертификат ЦС Контрольная панель не отзывает и CRL не выпускает. Если ключ ЦС скомпрометирован, отзыв прежнего сертификата ЦС и выпуск CRL выполните отдельно.

Шаг 4. Получите сертификат ЦС. Нажмите **Скачать CSR**, и браузер сохранит файл `csr-<наименование ЦС>.pem`. Передайте его ЦС, который выпускает сертификат для этого ЦС.

Если вышестоящим ЦС служит корневой ЦС Clearway CA, скопируйте файл CSR с рабочего места на сервер корневого ЦС в каталог /tmp. Затем на сервере корневого ЦС от имени itc-svc выпустите сертификат по шаблону subca, так же как при развертывании выдающего ЦС:

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ca -csr /tmp/csr-<наименование ЦС>.pem -out /tmp/<новый
сертификат ЦС>.crt -template subca
```

Перед загрузкой проверьте сертификат там же, на сервере корневого ЦС от имени itc-svc. Если сертификат выпустил другой ЦС, выполните проверку на любом компьютере с OpenSSL, указав пути к своим файлам.

```
openssl req -in /tmp/csr-<наименование ЦС>.pem -noout -pubkey | openssl sha256
openssl x509 -in /tmp/<новый сертификат ЦС>.crt -noout -pubkey | openssl sha256
openssl x509 -in /tmp/<новый сертификат ЦС>.crt -noout -subject -issuer -dates -ext
subjectKeyIdentifier
```

Хеши открытых ключей в CSR и в сертификате должны совпасть: значит, сертификат выпущен по этому запросу. Субъект, издатель и срок действия должны соответствовать ожидаемым.

Перед заменой при «Перевыпуске»

Убедитесь, что выполнен шаг 2: созданы резервные копии и выпущен тестовый сертификат прежним ЦС. После установки нового сертификата ядро ЦС сразу выпускает сертификаты только новым ключом.

Шаг 5. Замените сертификат ЦС. Нажмите **Заменить сертификат** и выберите файл .pem, .crt, .cer или .der (сертификат X.509 в формате PEM или DER). Контрольная панель разберёт файл и покажет издателя, серийный номер, срок действия и отпечаток SHA-256; на этом шаге сертификат ещё не проверяется. Сверьте сведения и нажмите **Заменить**, затем прочитайте результат и нажмите **Готово**.

Перед установкой ядро ЦС проверяет сертификат:

- поля совпадают с полями действующего сертификата ЦС, кроме серийного номера и срока действия;
- открытый ключ совпадает с действующим ключом ЦС («Продление») или с новым ключом из каталога `ca.new-keys` («Перевыпуск»);
- сертификат подписан вышестоящим ЦС из файла цепочки (параметр `chain-file`);
- срок действия сертификата наступил и не истёк;
- расширение Basic Constraints критическое и содержит CA=TRUE, в Key Usage есть keyCertSign и cRLSign.

Если проверка пройдена, ядро сохраняет прежнюю пару и устанавливает сертификат, как описано в начале раздела.

Результат	Что означает и что делать
Замена завершена	Ядро ЦС проверило и установило сертификат ЦС
Сертификат уже установлен	Этот сертификат уже установлен у ЦС. Ядро ЦС ничего не меняет, замена не требуется
Сертификат отклонён проверкой ЦС	Ядро ЦС не приняло сертификат, причины перечислены в окне. Нажмите **Выбрать другой файл** и загрузите верный сертификат, выпущенный по последнему сформированному CSR
Не удалось прочитать файл	Браузер не смог прочитать файл. Проверьте, что это сертификат X.509 в формате PEM или DER

Шаг 6. Проверьте результат. На странице появляется плашка «Замена сертификата выполнена». Карточка «Действующий сертификат ЦС» показывает новые серийный номер и срок действия; SKI при «Продлении» остаётся прежним, при «Перевыпуске» меняется на новый. На сервере ЦС от имени itc-svc проверьте, с каким ключом работает ядро ЦС:

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ping
```

Команда выводит версию ЦС и SKI текущего ЦС. SKI должен совпадать с SKI в карточке. Службу minica перезапускать не нужно.

При «Перевыпуске» убедитесь, что ядро ЦС сохранило прежнюю пару: в каталоге old-keys есть файлы .key и .crt, имя которых совпадает с SKI прежнего ЦС из шага 1.2 раздела «Ротация с новой ключевой парой».

```
ls -l /opt/itc/minica/ca/old-keys
```

Шаг 7. Завершите ротацию.

При «Продлении» проверьте работу с новым сертификатом ЦС:

1. Выпустите тестовый сертификат (шаг 4.1 раздела «Ротация с новой ключевой парой»); его AKI должен совпадать с прежним SKI.
2. Проверьте его статус второй командой шага 4.2, указав в `-url` адрес действующего экземпляра mOCSP (по умолчанию порт 8888); ответ должен пройти проверки из шага 4.2.

3. Проверьте состав файла цепочки сертификатов ЦС командой из шага 2.3: ядро ЦС заменило в нём прежний сертификат ЦС новым.
4. Если продлён сертификат корневого ЦС, добавьте новый сертификат в доверенные, как в шаге 4.7.
5. Отзовите тестовый сертификат командой `/opt/itc/minica/mclient revoke -serial <серийный номер тестового сертификата>`.

При «Перевыпуске» проверьте состав файла цепочки командой из шага 2.3 и выполните этапы 3 и 4 раздела «Ротация с новой ключевой парой»: запустите отдельный экземпляр mOCSP для нового ЦС и проверьте отзыв, CRL и OCSP для обоих поколений сертификатов.

7.3.4 Ротация с новой ключевой парой

Процедура нужна после «Перевыпуска» в Контрольной панели и при ручной замене ключа ЦС. Прежняя ключевая пара сохраняется в каталоге old-keys, чтобы ядро ЦС могло отзывать прежние сертификаты и выпускать для них CRL, а OCSP-запросы по прежним и новым сертификатам обслуживают отдельные экземпляры mOCSP.

Команды выполняйте от имени технологической учётной записи itc-svc (`su - itc-svc`): часть на сервере ЦС, часть на сервере, где работает mOCSP. Пути приведены для установки по умолчанию: ядро ЦС в /opt/itc/minica, mOCSP в /opt/itc/ocsp.

Этап	Где выполняется	Результат
1. Подготовка	Сервер ЦС, сервер mOCSP	Резервные копии; прежняя пара в old-keys; проверены комплект OCSP и адрес OCSP; выпущен тестовый сертификат прежним ЦС
2. Замена ключа и сертификата ЦС	Контрольная панель или сервер ЦС	Ядро ЦС работает с новым ключом, `mclient ping` возвращает новый SKI
3. OCSP-ответчик для нового ЦС	Сервер ЦС, сервер mOCSP	Работают два экземпляра mOCSP: прежнего и нового ЦС
4. Проверка	Сервер ЦС	Отзыв, CRL и OCSP работают для сертификатов обоих ЦС

7.3.4.1 Этап 1. Подготовка

Этап выполняется до замены ключа. При «Перевыпуске» в Контрольной панели его выполняют до формирования CSR, кроме шага 1.3: прежнюю пару в каталог old-keys ядро ЦС сохраняет само.

Шаг 1.1. Создайте резервные копии. Сохраните ключи и сертификаты ЦС и комплект OCSP: ключ и сертификат ответчика, сертификат ЦС и конфигурацию mOCSP. На сервере ЦС:

```
tar czf <каталог копий>/minica_before_rotation_$(date +%F).tgz /opt/itc/minica/ca /opt/
itc/minica/conf
```

На сервере mOCSP:

```
tar czf <каталог копий>/mocsp_before_rotation_$(date +%F).tgz /opt/itc/ocsp/conf /opt/
itc/ocsp/mocsp.env
```

Убедитесь, что прежний ЦС и комплект OCSP можно восстановить из копий. Проверьте оба архива командой `tar tzf <архив>` (tar сохраняет пути без начального «/»):

- в архиве сервера ЦС: opt/itc/minica/ca/ca.key и opt/itc/minica/ca/ca.crt;
- в архиве сервера mOCSP: opt/itc/ocsp/conf/mocsp.key, opt/itc/ocsp/conf/mocsp.crt и opt/itc/ocsp/conf/ca.crt.

Требования к хранению копий приведены в разделе «Резервное копирование».

Шаг 1.2. Определите SKI прежнего сертификата ЦС. На сервере ЦС выполните:

```
openssl x509 -in /opt/itc/minica/ca/ca.crt -noout -ext subjectKeyIdentifier
```

Пример результата:

```
X509v3 Subject Key Identifier:
    AA:BB:CC:DD:EE:FF:00:11:22:33:44:55:66:77:88:99:AA:BB:CC:DD
```

Имена файлов в каталоге old-keys образуются из этого значения:

- без двоеточий;
- без пробелов;
- в верхнем регистре.

Для примера выше получится AABBCCDDEEFF00112233445566778899AABBCCDD. Чтобы не ошибиться при наборе, сохраните значение в переменной текущего сеанса:

```
SKI=$(openssl x509 -in /opt/itc/minica/ca/ca.crt -noout -ext subjectKeyIdentifier |
tail -n 1 | tr -d ' :' | tr 'a-f' 'A-F')
echo "$SKI"
```

Команда выводит SKI одной строкой без двоеточий и пробелов. Запишите его: он понадобится на этапе 4.

Шаг 1.3. Сохраните прежнюю пару в каталоге old-keys. Шаг нужен только при ручной замене файлов ключа и сертификата: в этом случае ядро ЦС архив прежних ключей не создаёт. При установке сертификата из Контрольной панели или командой `mclient new-ca-cert` ядро ЦС сохраняет прежнюю пару само. Команды шага используют переменную SKI из шага 1.2, поэтому выполняйте их в том же сеансе. Сначала проверьте переменную:

```
test -n "$SKI" && echo "SKI=$SKI" || echo "Переменная SKI не задана, повторите шаг 1.2"
```

Если переменная не задана, не продолжайте: иначе файлы получат имена old-keys/.key и old-keys/.crt, и ядро ЦС их не найдёт. Следующие команды создают каталог, если его ещё нет, и копируют текущие файлы под именами по SKI:

```
cd /opt/itc/minica/ca
mkdir -p old-keys
cp -p ca.key "old-keys/${SKI}.key"
cp -p ca.crt "old-keys/${SKI}.crt"
chmod 0600 "old-keys/${SKI}.key"
ls -l old-keys
```

Итоговая структура каталога:

```
/opt/itc/minica/ca/
├── ca.key
├── ca.crt
└── old-keys/
    ├── AABBCCDDEEFF00112233445566778899AABBCCDD.key
    └── AABBCCDDEEFF00112233445566778899AABBCCDD.crt
```

Проверка: в выводе `ls -l` оба файла принадлежат itc-svc, у файла .key права `-rw-------`.

> **Защита прежних ключей**
>
> Доступ к закрытым ключам в каталогах old-keys и new-keys ограничивайте так же, как к действующему ключу ЦС: назначьте владельцем технологическую учётную запись и установите права 0600 (см. подраздел Ограничение прав доступа к файлам (see page 178)). Не помещайте закрытые ключи в системы контроля версий.

Шаг 1.4. Проверьте комплект прежнего OCSP-ответчика. Файлы /opt/itc/ocsp/conf/mocsp.key, mocsp.crt и ca.crt остаются в работе у экземпляра mOCSP прежнего ЦС. Не изменяйте их. Проверьте издателя и срок действия сертификата ответчика:

```
openssl x509 -in /opt/itc/ocsp/conf/mocsp.crt -noout -subject -issuer -enddate
```

Издатель должен быть прежним ЦС. Если срок сертификата ответчика скоро истекает, выпустите новый сертификат ответчика прежним ЦС до ротации: после ротации ядро ЦС выпускает сертификаты только новым ключом.

Шаг 1.5. Определите адрес OCSP в выпущенных сертификатах. Возьмите любой сертификат, выпущенный прежним ЦС, и выполните:

```
openssl x509 -in <сертификат, выпущенный прежним ЦС>.crt -noout -ext
authorityInfoAccess
```

Строка `OCSP - URI:…` содержит адрес, по которому клиенты проверяют статус этого сертификата. Этот адрес должен остаться за экземпляром mOCSP прежнего ЦС. Если строки OCSP нет, адрес ответчика в сертификатах не записан, и клиенты обращаются к ответчику по адресу из своих настроек (см. «Учет OCSP URL в выпущенных сертификатах»).

Шаг 1.6. Выпустите тестовый сертификат прежним ЦС. Он понадобится на этапе 4 для проверки отзыва и CRL прежнего ЦС; после ротации выпустить его будет нельзя. Команды выпускают реальный сертификат, на этапе 4 он будет отозван. На сервере ЦС:

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
openssl req -new -newkey rsa:2048 -nodes -keyout /tmp/test-old.key -out /tmp/test-
old.csr -subj "/CN=<тестовое имя>"
/opt/itc/minica/mclient ca -csr /tmp/test-old.csr -template tls -out /tmp/test-old.crt
openssl x509 -in /tmp/test-old.crt -noout -serial -ext authorityKeyIdentifier
```

Значение AKI (keyid) тестового сертификата должно совпадать с SKI из шага 1.2. Запишите серийный номер.

7.3.4.2 Этап 2. Замена ключа и сертификата ЦС

При «Перевыпуске» в Контрольной панели выполните шаги 3–6 раздела «Ротация из Контрольной панели», проверьте состав файла цепочки командой из шага 2.3 и переходите к этапу 3. Файлы ключа, сертификата и цепочки ядро ЦС в этом случае заменяет само, без перезапуска службы. При ручной замене выполните шаги ниже.

Адрес OCSP в сертификатах нового ЦС

Если адрес OCSP записывается в AIA выпускаемых сертификатов, не выпускайте рабочие сертификаты новым ЦС до выполнения шага 3.5. Иначе они получат адрес ответчика прежнего ЦС, а у выпущенного сертификата адрес изменить нельзя.

Шаг 2.1. Подготовьте новую ключевую пару и сертификат. Создайте ключ и запрос в отдельном каталоге, а не сразу в файл /opt/itc/minica/ca/ca.key: команды раздела «Развертывание ядра ЦС (minica)» записывают ключ сразу в ca/ca.key и заменили бы действующий ключ. Например:

```
mkdir -p /opt/itc/minica/ca/new
openssl genrsa -out /opt/itc/minica/ca/new/ca.key 4096
chmod 0600 /opt/itc/minica/ca/new/ca.key
openssl req -new -key /opt/itc/minica/ca/new/ca.key -out /opt/itc/minica/ca/new/ca.csr
-subj "<Subject ЦС>"
```

Получите сертификат по запросу. Выдающему ЦС его выпускает вышестоящий ЦС (корневой ЦС Clearway CA выпускает его по шаблону subca, см. шаг 4 раздела «Ротация из Контрольной панели»). Для корневого ЦС выпустите самоподписанный сертификат, как при развертывании. Сохраните сертификат как /opt/itc/minica/ca/new/ca.crt и проверьте, что ключ соответствует сертификату (хеши должны совпасть):

```
openssl x509 -in /opt/itc/minica/ca/new/ca.crt -noout -pubkey | openssl sha256
openssl pkey -in /opt/itc/minica/ca/new/ca.key -pubout | openssl sha256
```

Шаг 2.2. Установите новые ключ и сертификат ЦС. Команды заменяют рабочие файлы ЦС: выполняйте их только после шага 1.3.

```
cp /opt/itc/minica/ca/new/ca.key /opt/itc/minica/ca/ca.key
cp /opt/itc/minica/ca/new/ca.crt /opt/itc/minica/ca/ca.crt
chmod 0600 /opt/itc/minica/ca/ca.key
```

Шаг 2.3. Обновите файл цепочки сертификатов. Файл /opt/itc/minica/ca/chain.pem (параметр `chain-file`) содержит сертификат ЦС и собирается так же, как при развертывании. Для корневого ЦС:

```
cp /opt/itc/minica/ca/ca.crt /opt/itc/minica/ca/chain.pem
```

Для выдающего ЦС в файл входят сертификат корневого ЦС и новый сертификат выдающего ЦС:

```
cat <сертификат корневого ЦС>.crt /opt/itc/minica/ca/ca.crt > /opt/itc/minica/ca/
chain.pem
```

Проверьте состав файла: команда выводит субъект и дату окончания срока действия каждого сертификата цепочки.

```
openssl crl2pkcs7 -nocrl -certfile /opt/itc/minica/ca/chain.pem | openssl pkcs7
-print_certs -text -noout | grep -E "Subject:|Not After"
```

У сертификата с субъектом этого ЦС дата Not After должна совпадать с датой «Действует по» нового сертификата ЦС. После замены из Контрольной панели проверьте файл так же: ядро ЦС перестраивает его само, собирать файл вручную не нужно.

Шаг 2.4. Перезапустите службу minica и проверьте SKI. При запуске служба читает текущий `ca.cert`, извлекает его SKI и считает этот ЦС текущим. На время перезапуска ЦС не принимает запросы (см. Управление службой (see page 138)).

```
systemctl --user restart minica.service
systemctl --user is-active minica.service
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ping
```

Результат: `active`, а `mclient ping` выводит новый SKI, отличный от SKI из шага 1.2. После проверки создайте резервную копию нового ключа и сертификата ЦС (см. «Резервное копирование») и удалите промежуточный каталог /opt/itc/minica/ca/new.

7.3.4.3 Этап 3. OCSP-ответчик для нового ЦС

Один процесс mOCSP обслуживает ровно один ЦС (см. «Как mOCSP обслуживает прежний и новый ЦС»). Действующая служба mocsp.service остаётся ответчиком прежнего ЦС: её файлы key-file, cert-file и ca-file относятся к прежнему ЦС, менять их не нужно, адрес сохраняется. Для нового ЦС запускается второй экземпляр со своей конфигурацией, файлом окружения, юнитом и адресом. Имена файлов второго экземпляра ниже (current, mocsp-current) даны для примера.

Шаг 3.1. Создайте ключ и запрос ответчика нового ЦС. На сервере mOCSP:

```
mkdir -p /opt/itc/ocsp/conf/current
openssl genrsa -out /opt/itc/ocsp/conf/current/mocsp.key 4096
chmod 0600 /opt/itc/ocsp/conf/current/mocsp.key
openssl req -new -key /opt/itc/ocsp/conf/current/mocsp.key -out /opt/itc/ocsp/conf/
current/mocsp.csr -sha256 -nodes -subj "/CN=<fqdn сервера mOCSP>" -addext
"subjectAltName = DNS:<fqdn сервера mOCSP>"
```

Шаг 3.2. Выпустите сертификат ответчика новым ЦС. Скопируйте mocsp.csr на сервер ЦС в /tmp и выпустите сертификат по шаблону ocsp (не tls):

```
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
/opt/itc/minica/mclient ca -csr /tmp/mocsp.csr -out /tmp/mocsp.crt -template ocsp
```

Скопируйте на сервер mOCSP файл /tmp/mocsp.crt в /opt/itc/ocsp/conf/current/mocsp.crt, а новый сертификат ЦС /opt/itc/minica/ca/ca.crt в /opt/itc/ocsp/conf/current/ca.crt. Затем проверьте комплект на сервере mOCSP:

```
cd /opt/itc/ocsp/conf/current
```

```
openssl x509 -in mocsp.crt -noout -issuer -enddate -ext extendedKeyUsage
openssl verify -partial_chain -CAfile ca.crt mocsp.crt
openssl x509 -in mocsp.crt -noout -pubkey | openssl sha256
openssl pkey -in mocsp.key -pubout | openssl sha256
```

Результат: издателем указан новый ЦС, в расширенном назначении ключа есть `OCSP Signing`, проверка выводит `mocsp.crt: OK`, хеши открытых ключей совпадают.

Шаг 3.3. Подготовьте конфигурацию второго экземпляра. Скопируйте рабочую конфигурацию. Файл содержит пароль БД, поэтому доступ к копии ограничьте владельцем:

```
cp -p /opt/itc/ocsp/conf/mocsp.yml /opt/itc/ocsp/conf/mocsp-current.yml
chmod 0600 /opt/itc/ocsp/conf/mocsp-current.yml
```

В файле mocsp-current.yml измените параметры:

```
listen-port: <порт экземпляра нового ЦС>
url-path: /ocsp
key-file: /opt/itc/ocsp/conf/current/mocsp.key
cert-file: /opt/itc/ocsp/conf/current/mocsp.crt
ca-file: /opt/itc/ocsp/conf/current/ca.crt
```

Экземпляры должны различаться портами (`listen-port`), IP-адресами (`listen-ip`) и URL либо разводиться правилами внешней маршрутизации. В примере второй экземпляр на том же сервере отличается портом; его можно разместить и на другом сервере. В секции `log` задайте отдельный файл журнала в параметре `file`, например /opt/itc/ocsp/logs/mocsp-current.log. Секцию `database` не меняйте: оба экземпляра работают с одной базой данных ЦС.

Шаг 3.4. Создайте и запустите службу второго экземпляра. Файл окружения задаёт путь к конфигурации (переменная `MOCSP_CONF`), а юнит читает его через `EnvironmentFile`. Команды создают копии файла окружения и юнита и запускают новую службу:

```
cp -p /opt/itc/ocsp/mocsp.env /opt/itc/ocsp/mocsp-current.env
sed -i 's|^MOCSP_CONF=.*|MOCSP_CONF="/opt/itc/ocsp/conf/mocsp-current.yml"|' /opt/itc/
ocsp/mocsp-current.env
cp ~/.config/systemd/user/mocsp.service ~/.config/systemd/user/mocsp-current.service
sed -i 's|^EnvironmentFile=.*|EnvironmentFile=/opt/itc/ocsp/mocsp-current.env|'
 ~/.config/systemd/user/mocsp-current.service
systemctl --user daemon-reload
systemctl --user enable --now mocsp-current.service
```

Проверка:

```
systemctl --user is-active mocsp.service mocsp-current.service
ss -ltn | grep -E ':8888|:<порт экземпляра нового ЦС>'
```

Обе службы должны быть в состоянии `active`, а оба порта должны прослушиваться. Откройте порт нового экземпляра на межсетевом экране для клиентов OCSP. После замены key-file, cert-file или ca-file любого экземпляра перезапускайте его службу.

Шаг 3.5. Укажите адрес OCSP нового ЦС в выпускаемых сертификатах. Если адрес OCSP записывается в AIA выпускаемых сертификатов (параметр `ca.aia`, см. «Адрес OCSP-ответчика в AIA» в разделе «Шаблоны, поля CDP/AIA и сроки действия»), замените в нём адрес на адрес экземпляра нового ЦС и перезапустите minica.service. Сделайте это до выпуска рабочих сертификатов новым ЦС: у выпущенного сертификата адрес изменить нельзя.

7.3.4.4 Этап 4. Проверка

Проверки выполняются отдельно для сертификатов прежнего и нового ЦС: корректная работа одного поколения не подтверждает работу другого. Команды выполняйте на сервере ЦС от имени itc-svc с заданной переменной `MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml`.

Шаг 4.1. Выпустите тестовый сертификат новым ЦС.

```
openssl req -new -newkey rsa:2048 -nodes -keyout /tmp/test-new.key -out /tmp/test-new.c
sr -subj "/CN=<тестовое имя>"
/opt/itc/minica/mclient ca -csr /tmp/test-new.csr -template tls -out /tmp/test-new.crt
openssl x509 -in /tmp/test-new.crt -noout -serial -ext authorityKeyIdentifier
```

Значение AKI (keyid) должно совпадать с новым SKI из этапа 2.

Шаг 4.2. Проверьте OCSP для прежнего и нового сертификатов. Сертификат прежнего ЦС проверяется запросом к экземпляру прежнего ЦС (по умолчанию порт 8888, путь /ocsp):

```
openssl ocsp -issuer /opt/itc/minica/ca/old-keys/<SKI прежнего ЦС>.crt -cert /tmp/test-
old.crt -url http://<сервер mOCSP>:8888/ocsp -CAfile <цепочка прежнего ЦС>.pem
-resp_text
```

Сертификат нового ЦС проверяется запросом к экземпляру нового ЦС:

```
openssl ocsp -issuer /opt/itc/minica/ca/ca.crt -cert /tmp/test-new.crt -url http://
<сервер mOCSP>:<порт экземпляра нового ЦС>/ocsp -CAfile <цепочка нового ЦС>.pem
-resp_text
```

В `-CAfile` укажите файл с сертификатом корневого ЦС и сертификатами выдающих ЦС соответствующего поколения. В ответе проверьте:

- статус ответа: `OCSP Response Status: successful`;
- статус сертификата: ожидаемый (`good`, `revoked` или `unknown`);
- подпись ответа: `Response verify OK`;
- ответ дан для нужного издателя (поля Issuer Name Hash и Issuer Key Hash);
- в ответе есть сертификат ответчика, если задано `add-cert: true`.

Шаг 4.3. Отзовите тестовый сертификат прежнего ЦС и сформируйте его CRL. Команды отзывают сертификат и формируют CRL прежнего ЦС, подписанный прежним ключом:

```
/opt/itc/minica/mclient revoke -serial <серийный номер тестового сертификата прежнего
ЦС>
/opt/itc/minica/mclient updcrl -aki <SKI прежнего ЦС>
```

Сертификат можно отозвать и в Контрольной панели, в карточке сертификата. Затем получите сформированный CRL прежнего ЦС с тем же параметром `-aki`:

```
/opt/itc/minica/mclient crl -aki <SKI прежнего ЦС>
```

Команда возвращает из базы данных последний полный CRL прежнего ЦС. Сохраните его в файл /tmp/old-ca.crl.

Шаг 4.4. Проверьте CRL прежнего ЦС. Реквизиты CRL:

```
openssl crl -in /tmp/old-ca.crl -noout -issuer -lastupdate -nextupdate -crlnumber
```

Подпись проверяется сертификатом прежнего ЦС из каталога old-keys:

```
openssl crl -in /tmp/old-ca.crl -noout -verify -CAfile /opt/itc/minica/ca/old-keys/<SKI
прежнего ЦС>.crt
openssl crl -in /tmp/old-ca.crl -noout -text | grep "<серийный номер тестового
сертификата прежнего ЦС>"
```

Результат: издателем указан прежний ЦС, проверка подписи выводит `verify OK`, серийный номер тестового сертификата есть в списке. Если файл в формате DER, добавьте к командам `openssl crl` параметр `-inform DER`.

Шаг 4.5. Убедитесь, что OCSP показывает отзыв. Повторите первую команду шага 4.2: для /tmp/test-old.crt должен вернуться статус `revoked`.

Шаг 4.6. Повторите проверки для нового ЦС. Отзовите /tmp/test-new.crt, сформируйте CRL текущего ЦС командой `mclient updcrl` без параметра `-aki`, проверьте подпись CRL сертификатом /opt/itc/minica/ca/ca.crt и статус `revoked` в ответе экземпляра mOCSP нового ЦС.

Шаг 4.7. Обновите доверие к ЦС. Если заменён ключ корневого ЦС, добавьте его новый сертификат в доверенные (см. «Добавление сертификата ЦС в доверенные»). Прежний сертификат не удаляйте, пока действуют выпущенные им сертификаты.

7.3.5 Как ядро ЦС работает с прежними ключами

Ядро ЦС (minica) хранит прежние ключевые пары в каталоге `ca.old-keys` и выбирает нужную пару по значению AKI. Каталог используется только ядром ЦС: mOCSP его не читает.

7.3.5.1 Параметры конфигурации ядра ЦС

За выбор ключа ЦС, доступ к прежним ключам и хранение нового ключа при ротации отвечают следующие параметры /opt/itc/minica/conf/minica.yml (показаны значения по умолчанию):

```
ca-pass: ""
openssl:
  pkcs11:
    enable: false
    ca-uri: ""

ca:
  key: ca/ca.key
  cert: ca/ca.crt
  old-keys: ca/old-keys
  new-keys: ca/new-keys
```

Параметр	Назначение	Значение по умолчанию
`ca.key`	Закрытый ключ текущего ЦС	ca/ca.key
`ca.cert`	Сертификат текущего ЦС	ca/ca.crt
`ca.old-keys`	Каталог ключей и сертификатов предыдущих ЦС	ca/old-keys
`ca.new-keys`	Каталог, в котором ядро ЦС хранит новый ключ при «Перевыпуске» из Контрольной панели до установки сертификата	ca/new-keys

`ca-pass`	Пароль файлового ключа ЦС	пусто
`openssl.pkcs11.enable`	Использовать ключ ЦС через PKCS#11	false
`openssl.pkcs11.ca-uri`	URI единственного ключа ЦС в PKCS#11	пусто

Относительные пути отсчитываются от рабочего каталога службы /opt/itc/minica. Установщик записывает в minica.yml абсолютные пути: /opt/itc/minica/ca/ca.key, /opt/itc/minica/ca/ca.crt и /opt/itc/minica/ca/old-keys. Если параметры `ca-pass`, `ca.new-keys` и `openssl.pkcs11` в файле не заданы, действуют значения по умолчанию. Изменения minica.yml вступают в силу после перезапуска службы (см. Управление службой (see page 138)).

При установке сертификата из Контрольной панели или командой `mclient new-ca-cert` ядро ЦС само копирует прежний сертификат (при новой ключевой паре и прежний ключ) в каталог old-keys. При ручной замене файлов ядро ЦС архив прежних ключей не создаёт: скопировать файлы в каталог old-keys должен администратор до того, как они будут заменены (шаг 1.3).

7.3.5.2 Выбор ключа по значению AKI

Значение AKI, полученное в API-запросе, нормализуется: удаляются пробелы по краям и двоеточия, символы приводятся к верхнему регистру. Затем ядро ЦС выбирает файлы:

AKI в запросе	**Какие файлы использует ядро ЦС**
Не задан	Текущие ca.key и ca.crt
Совпадает с SKI текущего ЦС	Текущие ca.key и ca.crt
Не совпадает с SKI текущего ЦС	`old-keys/<AKI>.key` и `old-keys/<AKI>.crt`

Например, для AKI AABB…CCDD используются файлы /opt/itc/minica/ca/old-keys/AABB…CCDD.key и /opt/itc/minica/ca/old-keys/AABB…CCDD.crt. Если хотя бы один из них не найден, операция завершается ошибкой ERR_UNKNOWN_AKI (код 140, см. «Коды ответов API»).

7.3.5.3 Операции, использующие прежние ключи

Отзыв сертификата. При отзыве ядро ЦС читает AKI сертификата из базы данных. Если AKI совпадает с SKI текущего ЦС, используется действующая пара, а если относится к предыдущему ЦС, то пара из каталога old-keys. Выбранные ключ и сертификат передаются OpenSSL при

выполнении `openssl ca -revoke`. Передавать AKI в запросе клиенту API не требуется: ядро ЦС получает его из записи сертификата.

Полный CRL. Эндпоинт /updcrl принимает необязательный параметр AKI (в mclient ему соответствует `updcrl -aki`). Без него создаётся CRL текущего ЦС, с ним создаётся CRL указанного ЦС. Для прежнего AKI ядро ЦС находит пару в каталоге old-keys, выбирает из базы отозванные сертификаты с этим AKI, формирует CRL, подписывает его прежним ключом ЦС и сохраняет в базе с соответствующим AKI.

Delta CRL. Эндпоинт /updeltacrl (в mclient ему соответствует `updeltacrl -aki`) работает аналогично. Он выбирает ЦС по AKI, находит последний полный CRL этого ЦС, затем последующий CRL или Delta CRL, отбирает новые записи об отзыве с тем же AKI и подписывает Delta CRL выбранным ключом ЦС.

Получение готового CRL. Эндпоинты /crl и /deltacrl прежние ключи не используют: они возвращают уже сформированный CRL из базы данных по значению AKI.

Выпуск сертификатов. Каталог old-keys не позволяет выпускать новые сертификаты от имени прежнего ЦС. Операции выпуска всегда используют текущие `ca.key` и `ca.cert`.

7.3.5.4 Ограничения механизма old-keys

Пароль прежнего ключа. Значение `ca-pass` применяется и к текущему ключу, и к файлам из каталога old-keys. Отдельный пароль для каждого прежнего ЦС конфигурацией не предусмотрен. Если прежние ключи зашифрованы разными паролями, приведите их к единому паролю `ca-pass`.

Хранение ключа в PKCS#11. При такой конфигурации OpenSSL использует ключ из единственного `ca-uri`:

```
openssl:
  pkcs11:
    enable: true
    ca-uri: <URI текущего ключа>
```

Путь `old-keys/<AKI>.key`, выбранный ядром ЦС, в этом случае не передаётся OpenSSL как файловый ключ. Механизм old-keys рассчитан прежде всего на ключи в файлах PEM: одновременный выбор нескольких прежних ключей в HSM по значению AKI не предусмотрен. Ротация из Контрольной панели (и командами `mclient new-ca-req` и `mclient new-ca-cert`) для ЦС с ключом на носителе PKCS#11 или в КриптоПро CSP недоступна: ядро ЦС возвращает ошибку. Хранение ключа ЦС на аппаратном носителе не поддерживается в бесплатной редакции (см. «Хранение ключа ЦС на HSM/Rutoken (PKCS#11)»).

7.3.6 Как mOCSP обслуживает прежний и новый ЦС

В mOCSP аналога каталога old-keys нет: прежние ключи ЦС из него сервис не получает. Один процесс mOCSP загружает:

- один закрытый ключ OCSP-ответчика;
- один сертификат OCSP-ответчика;
- один сертификат издающего ЦС.

Поэтому один процесс обслуживает ровно один ЦС, и для прежнего и нового ЦС нужны отдельные экземпляры службы (этап 3).

7.3.6.1 Параметры конфигурации mOCSP

Обслуживаемый ЦС и ключ подписи ответа задаются в /opt/itc/ocsp/conf/mocsp.yml:

```
listen-port: 8888
url-path: /ocsp
key-file: /opt/itc/ocsp/conf/mocsp.key
cert-file: /opt/itc/ocsp/conf/mocsp.crt
ca-file: /opt/itc/ocsp/conf/ca.crt
add-cert: true
hash: SHA256
nmin: 60
update-expired: true
no-trim-zeros: false
```

Процесс отвечает по сертификатам того ЦС, сертификат которого указан в `ca-file`. Ответ подписывается закрытым RSA-ключом из `key-file`, а в `cert-file` указан сертификат OCSP-ответчика. Экземпляр принимает запросы на порту `listen-port` по пути `url-path`. Параметр `hash` на выбор ЦС не влияет. Назначение и значения по умолчанию всех параметров приведены в подразделе «OCSP-респондер mOCSP» раздела Настройка вспомогательных сервисов (see page 139).

Путь к файлу конфигурации экземпляра задаёт переменная `MOCSP_CONF` в файле окружения /opt/itc/ocsp/mocsp.env, который юнит mocsp.service подключает через `EnvironmentFile`.

При запуске mOCSP вычисляет для файла `ca-file` значения IssuerKeyHash и IssuerNameHash. OCSP-запрос содержит такие же значения для ЦС, выпустившего проверяемый сертификат. Если хеши запроса не совпадают с хешами настроенного `ca-file`, mOCSP возвращает ошибку unauthorized (в выводе openssl: `Responder Error: unauthorized (6)`).

Почему нужны отдельные экземпляры

> Процесс mOCSP, настроенный на новый ЦС, не сможет обслужить запрос по сертификату прежнего ЦС, и наоборот. Отсюда и требование запускать отдельные экземпляры службы.

7.3.6.2 Чем подписывается OCSP-ответ

OCSP-ответ подписывается ключом из `key-file`, а не файлом `ca/old-keys/<SKI>.key`: каталог old-keys ядра ЦС сервису mOCSP неизвестен.

Для прежнего ЦС нужны:

- сертификат прежнего ЦС;
- сертификат прежнего OCSP-ответчика;
- соответствующий ему закрытый RSA-ключ ответчика.

Сертификат делегированного ответчика mocsp.crt должен:

- соответствовать ключу mocsp.key;
- быть выдан нужным ЦС;
- быть пригоден для подписи OCSP-ответов;
- содержать необходимые расширения, включая назначение OCSPSigning.

Такой сертификат выпускается на ЦС по шаблону ocsp. Запрос по сертификату прежнего ЦС обрабатывается так:

1. Прежний целевой сертификат содержит AKI прежнего ЦС.
2. Клиент отправляет OCSP-запрос с хешами имени и ключа прежнего ЦС.
3. Запрос принимает экземпляр mOCSP прежнего ЦС: в его ca-file указан прежний сертификат ЦС, в cert-file прежний mocsp.crt, в key-file прежний mocsp.key.
4. Клиент получает OCSP-ответ, подписанный прежним ключом ответчика.

Закрытый ключ прежнего ЦС нужен ядру ЦС для подписи CRL, но для обычной делегированной подписи ответа mOCSP он не требуется. Подписывать OCSP-ответ ключом самого ЦС можно, только если его сертификат и закрытый ключ явно указаны в `cert-file` и `key-file`. Рекомендуется схема с отдельным делегированным OCSP-ответчиком.

7.3.6.3 Учет OCSP URL в выпущенных сертификатах

Адрес OCSP-ответчика попадает в расширение AIA сертификата при выпуске, если он задан в параметре `ca.aia` ядра ЦС (см. «Адрес OCSP-ответчика в AIA» в разделе «Шаблоны, поля CDP/AIA и сроки действия»). У выпущенного сертификата изменить этот адрес нельзя. Поэтому до отключения или перенастройки прежнего OCSP-ответчика проверьте, к какому адресу обращаются ранее выпущенные сертификаты (шаг 1.5).

Надёжная схема разводит поколения сертификатов по разным адресам:

1. Сертификаты прежнего ЦС → прежний адрес OCSP → mOCSP прежнего ЦС.
2. Сертификаты нового ЦС → новый адрес OCSP → mOCSP нового ЦС.

> **Один адрес для двух поколений**
>
> Если прежние и новые сертификаты обращаются к одному адресу, запросы нельзя распределять между экземплярами mOCSP произвольно: экземпляр с неподходящим `ca-file` вернёт unauthorized. В этом случае внешний компонент должен направлять запрос к нужному ответчику с учётом издателя проверяемого сертификата.

7.3.7 Типовые проблемы

Что видно	Причина	Что сделать
Пункт «Ротация ключей» не отображается в меню	В токене пользователя нет права `rleMiniCARotation-w`	Добавьте право группе пользователя в карте `ldapGroupRolesMap`, для встроенной учётной записи добавьте его в список `customClaims.roles` (см. Назначение ролей через группы AD/LDAP (see page 196)). Перезапустите сервис OpenID, затем выйдите из Контрольной панели и войдите снова
В Контрольной панели «Сертификат отклонён проверкой ЦС»	Сертификат не прошёл проверку ядра ЦС (шаг 5 раздела «Ротация из Контрольной панели»), например выпущен по CSR, который заменён повторным формированием CSR; причины перечислены в окне	Устраните причины и загрузите сертификат, выпущенный по последнему сформированному CSR (проверка хешей открытых ключей описана в шаге 4 раздела «Ротация из Контрольной панели»)
Отзыв сертификата прежнего ЦС или формирование его CRL завершается ошибкой ERR_UNKNOWN_AKI (код 140)	В каталоге old-keys нет файла `<AKI>.key` или `<AKI>.crt` либо имя не совпадает с AKI	Проверьте, что оба файла на месте и названы по SKI прежнего ЦС без двоеточий и пробелов, в верхнем регистре
`mclient ping` после ручной замены возвращает прежний SKI	Служба minica не перезапущена: SKI текущего ЦС определяется при запуске	Перезапустите minica.service и повторите проверку
Служба minica не запускается после замены файлов	Ключ не соответствует сертификату либо ключ зашифрован паролем, отличным от `ca-pass`	Просмотрите журнал `journalctl --user -u minica.service -n 50`, сравните хеши открытых ключей (шаг 2.1), проверьте `ca-pass`

mOCSP отвечает `unauthorized (6)`	Запрос пришёл к экземпляру, чей `ca-file` относится к другому ЦС	Направьте запрос экземпляру mOCSP ЦС, выпустившего проверяемый сертификат
Второй экземпляр mOCSP не запускается	Порт уже занят первым экземпляром	Задайте в mocsp-current.yml другой `listen-port` и перезапустите службу

7.3.8 Файлы, задействованные при ротации

Файл	Сервер	Назначение
/opt/itc/minica/conf/minica.yml	Сервер ЦС	Рабочая конфигурация ядра ЦС: параметры `ca.*`, `ca-pass`, `openssl.pkcs11`
/opt/itc/minica/ca/ca.key, /opt/itc/minica/ca/ca.crt	Сервер ЦС	Ключ и сертификат текущего ЦС
/opt/itc/minica/ca/old-keys/	Сервер ЦС	Ключи и сертификаты прежних ЦС: `<SKI>.key`, `<SKI>.crt`; после ротации из Контрольной панели также копия прежней цепочки (параметр `ca.old-keys`)
/opt/itc/minica/ca/new-keys/	Сервер ЦС	Новый ключ ЦС при «Перевыпуске» из Контрольной панели до установки сертификата (параметр `ca.new-keys`)
/opt/itc/minica/ca/chain.pem	Сервер ЦС	Цепочка сертификатов ЦС (параметр `chain-file`). При ротации из Контрольной панели ядро ЦС перестраивает её само
conf/minica.example.yml	Сервер ЦС	Расширенный пример конфигурации ядра ЦС
/opt/itc/ocsp/conf/mocsp.yml	Сервер mOCSP	Рабочая конфигурация mOCSP
/opt/itc/ocsp/mocsp.env	Сервер mOCSP	Путь к конфигурации (`MOCSP_CONF`) и параметры запуска (`MOCSP_OPTIONS`)

/opt/itc/ocsp/conf/mocsp.key, mocsp.crt, ca.crt	Сервер mOCSP	Ключ и сертификат ответчика, сертификат обслуживаемого ЦС
/home/itc-svc/.config/systemd/user/mocsp.service	Сервер mOCSP	Юнит mOCSP
conf/mocsp.example.yml	Сервер mOCSP	Расширенный пример конфигурации mOCSP

7.4 Управление службами

Раздел описывает, как запускать, останавливать и проверять службы Clearway CA. Эти команды нужны при изменении конфигурации, обновлении продукта, восстановлении из резервной копии и диагностике.

Службы Clearway CA работают как пользовательские юниты systemd технологической учётной записи (по умолчанию `itc-svc`), а не как системные службы. Файлы юнитов находятся в каталоге `/home/itc-svc/.config/systemd/user/`. Службами управляют командами `systemctl --user` из сеанса технологической учётной записи, поэтому для повседневного обслуживания права root не нужны.

Юниты служб: `minica.service` (ядро ЦС), `openid.service` (OpenID), `controlPanel.service` (Контрольная панель), `publication.service` (сервис публикации), `mocsp.service` (OCSP-ответчик mOCSP), `archiver.service` (архиватор), `scepAdapter.service` (адаптер SCEP), `wstepCep.service` и `wstepCesAdapter.service` (службы CEP и CES адаптера MS-WSTEP). Порты служб и способ их публикации приведены в таблице раздела «Службы и сетевые порты» главы «Архитектура и состав системы».



Состав служб на сервере

Набор служб на сервере зависит от выбранных для него групп установки. Служба `minica.service` работает на каждом сервере корневого и выдающего ЦС, остальные службы работают на серверах, где установлены соответствующие компоненты. Фактический перечень служб выводит команда `systemctl --user list-units --type=service` в сеансе технологической учётной записи.

Прикладные порты .NET-служб принимают подключения на всех сетевых интерфейсах сервера, поэтому их закрывают межсетевым экраном для подключений извне. Правила приведены в разделе «Ограничение сетевого доступа» главы Обеспечение безопасной среды (see page 167), а полный перечень портов дан в разделе «Приложение Б. Сетевые порты».

> **Системные службы Nginx и PostgreSQL**
>
> Nginx и PostgreSQL работают как системные службы ОС. Ими управляют командой `sudo systemctl` без ключа `--user`. Юнит Nginx называется `nginx.service`, а юнит PostgreSQL `postgresql.service` (ALT Linux, Debian, Astra Linux) или `postgresql-<версия>.service` (РЕД ОС). Команды `sudo systemctl …` и `sudo journalctl -u …` без ключа `--user` службы Clearway CA не находят.

7.4.1 Условия работы пользовательских юнитов

Пользовательские юниты работают без интерактивного входа, только если выполнены два условия. Установщик настраивает их сам, но при обслуживании их стоит проверять, например если службы остановились после выхода пользователя из системы. Эти условия входят в требования к технологической учётной записи, см. главу Требования к среде (see page 38).

Режим linger. Без него systemd останавливает службы учётной записи, когда завершается её последний сеанс. Проверка и включение выполняются под администратором сервера:

```
loginctl show-user itc-svc | grep Linger   # ожидается: Linger=yes
sudo loginctl enable-linger itc-svc        # включить, если выведено Linger=no
```

Переменные окружения. В сеансе технологической учётной записи должны быть заданы переменные XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS. Установщик прописывает их в файл `~/.bashrc` учётной записи. Если переменные не заданы, команды `systemctl --user` завершаются ошибкой `Failed to connect to bus`. Задать их в текущем сеансе можно так:

```
export XDG_RUNTIME_DIR="/run/user/$(id -u)"
export DBUS_SESSION_BUS_ADDRESS="unix:path=${XDG_RUNTIME_DIR}/bus"
```

7.4.2 Команды управления

Команды выполняются на сервере службы в сеансе технологической учётной записи. Войдите в него из учётной записи администратора командой `sudo su - itc-svc` (из сеанса root командой `su - itc-svc`): ключ `-` загружает окружение учётной записи, в том числе переменные из `~/.bashrc`. Примеры приведены для `minica.service`; для других служб подставьте имя юнита из перечня выше.

```
systemctl --user status  minica.service    # состояние и последние строки журнала
systemctl --user start   minica.service    # запустить
systemctl --user stop    minica.service    # остановить: служба перестаёт обслуживать
запросы
```

```
systemctl --user restart minica.service    # перезапустить: на время перезапуска служба
недоступна
systemctl --user enable  minica.service    # включить автозапуск
```

Остановка и перезапуск служб

Пока остановлена служба `minica.service`, выпуск и отзыв сертификатов недоступны. Пока перезапускается `openid.service`, в Контрольную панель нельзя войти. Останавливайте службы в согласованное окно обслуживания.

После изменения файла юнита перечитайте определения служб, затем перезапустите службу:

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

Изменения в конфигурационных файлах (`minica.yml` , `mocsp.yml` , `config.yaml` сервиса OpenID, JSON-конфигурации .NET-служб) применяются перезапуском соответствующей службы (`systemctl --user restart <юнит>`). Команда `daemon-reload` для этого не нужна.

Проверка активности службы, например перед настройкой Контрольной панели или SCEP-адаптера:

```
systemctl --user is-active controlPanel.service   # ожидается: active
systemctl --user is-active scepAdapter.service
```

Команда выводит `active` , если служба работает. Другие значения (`inactive` , `failed` , `activating`) означают, что служба остановлена, завершилась с ошибкой или ещё запускается; в этом случае команда возвращает ненулевой код завершения.

Журнал службы выводит команда `journalctl --user -u <юнит> -n 50` . Другие средства диагностики описаны в главе «Диагностика и устранение неисправностей».

7.5 Обновление продукта

Раздел описывает порядок перехода на новую версию Clearway CA и установки исправлений: как подготовиться, обновить компоненты и проверить результат.

Обновление выполняется штатным установщиком из дистрибутива производителя. Порядок обновления отдельных сервисов без установщика описан в документе «Ручное обновление сервисов Clearway CA».

Различают два вида обновления:

- полное обновление: переход на новую версию продукта (замена исполняемых файлов служб и, при необходимости, миграция схемы БД);
- частичное обновление: установка исправлений отдельных компонентов без изменения остальных.

Перед установкой обязательно проверьте подлинность и целостность полученного дистрибутива способом, который указал производитель. Дистрибутив, не прошедший проверку, устанавливать запрещено.

Простой на время обновления

Чтобы обновить службу, её нужно остановить. Пока обновляется ядро ЦС, выпуск и отзыв сертификатов недоступны; пока обновляется mOCSP, недоступна проверка статуса сертификатов по OCSP. Согласуйте окно обслуживания и заранее уведомите пользователей.

7.5.1 Подготовка

Шаг 1. Сделайте резервную копию. Запустите вне расписания задание резервного копирования ЦС в Контрольной панели и вручную скопируйте то, что задание не охватывает: архивную БД, БД OpenID, конфигурации и сертификаты остальных серверов, а для возможности отката также каталоги приложений. Порядок описан в разделе «Резервное копирование», подраздел «Ручное резервное копирование».

Шаг 2. Зафиксируйте текущие версии компонентов: они понадобятся для проверки результата и отката. Версии ЦС, серверной части Контрольной панели и сервиса публикации показаны на странице Контрольной панели «Настройки → Конфигурация». Версию ядра ЦС выводит также команда `mclient ping` на сервере ЦС в сеансе `itc-svc`:

```
cd /opt/itc/minica
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
./mclient ping    # в ответе: версия ЦС и SKI его ключа
```

Версии остальных компонентов определите по сопроводительному файлу версии установленного релиза. Версию mOCSP выводит также команда `/opt/itc/ocsp/mocsp -v` на сервере службы, а у архиватора, сервиса публикации и служб MS-WSTEP номер версии записан в файле `version.txt` в каталоге приложения. Состояние служб выводит `systemctl --user status <юнит>`.

Шаг 3. Проверьте подлинность и целостность дистрибутива.

7.5.2 Установка обновления

Шаг 4. Остановите обновляемые службы командой `systemctl --user stop <юнит>`. С этого момента они не обслуживают запросы.

Шаг 5. Запустите установщик и дождитесь завершения обновления компонентов. Если компоненты установлены на разных серверах, обновляйте их в рекомендованном порядке: ЦС → OpenID → Контрольная панель → сервис публикации → mOCSP → архиватор → SCEP → CDP. Не оставляйте в работе несовместимые версии ядра ЦС, Контрольной панели и OpenID.

Шаг 6. До запуска служб сверьте новые параметры конфигурации по сопроводительным материалам версии. Добавьте их в рабочие файлы вручную, не перезаписывая действующие значения.

Шаг 7. Выполните `systemctl --user daemon-reload`, запустите службы и проверьте их состояние командами `status` и `is-active` (см. «Управление службами»).

7.5.3 Проверка результата

Откройте в Контрольной панели страницу «Настройки → Конфигурация» (нужно право `rleMiniCAConfig-r`). Обновление прошло успешно, если индикаторы карточек «Конфигурация Clearway CA», «Конфигурация контрольной панели» и «Конфигурация сервиса публикации CRL» зелёные, а в полях «Версия» указаны новые версии.

Если служба после обновления не запускается, просмотрите её журнал командой `journalctl --user -u <юнит> -n 100 --no-pager` и типовые причины в главе «Диагностика и устранение неисправностей». Если причину устранить не удаётся, верните предыдущую версию из резервных копий; порядок отката описан в документе «Ручное обновление сервисов Clearway CA».

Откат теряет изменения

При откате из резервной копии теряются изменения, сделанные после её создания: выпуски и отзывы сертификатов. По возможности выполняйте откат до возобновления выпуска сертификатов.

Сведения об уязвимостях

Администратору следует отслеживать сообщения производителя о новых версиях и исправлениях безопасности, а также сведения об уязвимостях системного окружения (ОС, PostgreSQL, Nginx, OpenSSL) и своевременно применять обновления.

7.6 Резервное копирование

Резервная копия позволяет восстановить работу ЦС после сбоя, ошибки администратора или неудачного обновления. В Clearway CA 26.3.0 есть два способа резервного копирования, они дополняют друг друга:

- задания резервного копирования: сервис резервного копирования по расписанию или по команде из Контрольной панели копирует БД ЦС и файлы узлов ЦС и восстанавливает ЦС из копии;
- ручное копирование средствами PostgreSQL и архивированием каталогов: для остальных баз данных и серверов, в том числе сервера самого сервиса резервного копирования, а также как независимая копия.

7.6.1 Что копировать

В таблице перечислены данные, которые нужно сохранять: они не восстанавливаются из дистрибутива. Исключение составляет исполняемый файл ЦС: его копирует задание резервного копирования. Пути указаны для установки по умолчанию.

Данные	Где находятся	Способ копирования
БД ЦС	БД `cwica` на сервере БД ЦС	Задание резервного копирования или `pg_dump`
Ключи и сертификаты ЦС: `ca.key`, `ca.crt`, пара шифрования `encr.key` и `encr.crt`, сертификат KRA `kra.crt`, цепочка `chain.pem`, прежние ключи ЦС `old-keys/`, новый ключ ЦС до установки его сертификата `new-keys/`	Каталог `/opt/itc/minica/ca/` на сервере ЦС	Задание резервного копирования или `tar`
Конфигурация ЦС и клиента mclient (`minica.yml`, `mclient.yml`, `openssl.cnf`), TLS-ключ и сертификат ЦС, ключи JWT (`jwt.key`, `jwt-keys/`), ключи и JWT клиента mclient (`mclient.key`, `mclient.pem`, `mclient.jwt`)	Каталог `/opt/itc/minica/conf/` на сервере ЦС	Задание резервного копирования или `tar`
Файлы окружения ЦС	`/opt/itc/minica/minica.env`, `/opt/itc/minica/config-jwt.env`, `/opt/itc/minica/config-mclient.env`	Задание резервного копирования или `tar`

Исполняемый файл ЦС	`/opt/itc/minica/minica`	Задание резервного копирования (копируется при каждом запуске)
Архивная БД	БД `cwica_archive`	`pg_dump`
БД OpenID	БД `openid` на сервере OpenID	`pg_dump`
Конфигурация и TLS-сертификат OpenID	`/opt/itc/openid/config.yaml`, каталог `/opt/itc/itc/certs/` на сервере OpenID	`tar`
Конфигурации .NET-служб, включая подключения ЦС, добавленные через интерфейс Контрольной панели (`itc_api_miniCA_config.runtime.json`), и конфигурацию сервиса резервного копирования с заданиями (`itc_api_miniCA_backup_config.json`, `itc_api_miniCA_backup_config.runtime.json`)	Каталог `/opt/itc/itc/config/`	`tar`
Ключ шифрования резервных копий	Файл из параметра `Encryption.KeyFile` на сервере сервиса резервного копирования	Отдельно от копий, как ключевой материал
Копии, созданные заданиями резервного копирования	Каталог из параметра `StoragePath` на сервере сервиса резервного копирования	Перенос на отдельный носитель
Сертификаты и ключи служб: TLS-сертификат Nginx, сертификат и ключ SCEP, файлы подключения к ЦС (`<имя ЦС>.crt`, `.key`, `.jwt`)	Каталог `/opt/itc/itc/certs/`	`tar`

Настройки веб-интерфейса, конфигурации Nginx, окружение .NET-служб	Каталоги `/opt/itc/itc/env/` и `/opt/itc/itc/nginx/`, файл `/opt/itc/itc_svc_env.conf`	`tar`
Окружение и ключи Kerberos MS-WSTEP	`/opt/itc/itc_wstep_env.conf`, `/etc/krb5.keytab`	`tar`
Состояние SCEP-адаптера (SQLite)	`/opt/itc/itc/sqlite/scep.db`	`tar`
Конфигурация и ключ подписи mOCSP	`/opt/itc/ocsp/mocsp.env`, каталог `/opt/itc/ocsp/conf/` (`mocsp.yml`, `mocsp.key`, `mocsp.crt`, `ca.crt`)	`tar`
Конфигурация точки распространения CDP	`/opt/itc/web/nginx.conf`	`tar`
Определения юнитов systemd	Каталог `/home/itc-svc/.config/systemd/user/` на каждом сервере	`tar`

Закрытый ключ KRA в продукте не хранится, за его резервное копирование отвечает владелец ключа (см. раздел «Механизм KRA (резервное копирование и восстановление ключей)» главы Обеспечение безопасной среды (see page 167)).

Файл `/etc/krb5.keytab` создаётся при вводе сервера в домен. SSSD перезаписывает его, когда меняет пароль учётной записи компьютера в домене (по умолчанию, если пароль старше 30 дней). Если пароль сменился после создания копии, ключи в копии недействительны: вместо восстановления файла повторно введите сервер в домен.

7.6.2 Задания резервного копирования в Контрольной панели

Задание резервного копирования описывает, какую БД ЦС и какие узлы ЦС копировать и по какому расписанию. Узлом ЦС называется сервер, на котором работает служба ЦС `minica.service`. Задание запускается по расписанию cron или по команде администратора, и каждый запуск создаёт одну копию. Из копии ЦС восстанавливается в интерфейсе.

Задания выполняет отдельный сервис резервного копирования `ITC.Api.MiniCA.Backup` (приложение .NET). Серверная часть Контрольной панели сама ничего не копирует: запросы раздела «Задачи → Резервное копирование» она передаёт сервису вместе с токеном пользователя. Подключения к узлам ЦС по SSH, снятие и восстановление дампа БД, хранение копий и расписание находятся в сервисе.

Откуда	Куда	Порт	Назначение
Серверная часть Контрольной панели	Сервис резервного копирования	Адрес из параметра `Services.BackupUrl`, например `http://localhost:8409`	Запросы раздела «Задачи → Резервное копирование»
Сервис резервного копирования	Узлы ЦС	22/tcp (SSH), порт задаётся в задании	Копирование и восстановление файлов, остановка и запуск службы ЦС
Сервис резервного копирования	Сервер БД ЦС	5432/tcp, порт задаётся в задании	Снятие и восстановление дампа БД ЦС
Сервис резервного копирования	Сервис OpenID	443/tcp	Проверка токенов пользователей
Сервис резервного копирования	EXC, SMTP-сервер	Порты EXC и SMTP-сервера	Пароли по ссылкам EXC и письма о неуспешных операциях, если они используются

Сервис устанавливают на сервер Контрольной панели или на отдельный сервер. Если он работает на сервере Контрольной панели, подключения к узлам ЦС и к БД ЦС идут с этого сервера, но выполняет их сервис резервного копирования. Сама Контрольная панель подключается к серверам по SSH только в типовых задачах замены сертификатов. Параметры сервиса описаны в подразделе «Сервис резервного копирования» раздела Настройка вспомогательных сервисов (see page 139).

Раздел «Задачи → Резервное копирование» работает с ЦС, выбранным в Контрольной панели (карточка «Выбор Clearway CA» на странице «Настройки → Конфигурация»): новое задание относится к этому ЦС. ЦС задания показан в карточке задания.

Пошаговое описание экранов раздела приведено в Руководстве пользователя, раздел «Резервное копирование». Ниже приведены сведения, которые нужны администратору для настройки заданий.

7.6.2.1 Права

Права проверяет сервис резервного копирования по токену пользователя, который ему передаёт Контрольная панель.

Действие	Право
Открыть раздел, просмотреть задания, ход операций и копии	`rleMiniCABackup-r`
Добавить, изменить, удалить задание	`rleMiniCAConfig-w`
Запустить копирование вне расписания	`rleMiniCABackup-w`
Восстановить ЦС из копии	`rleMiniCABackupRestore-w`

Для восстановления из интерфейса пользователю нужно и право `rleMiniCABackup-r`: без него карточка задания не показывает список копий.

Пункт «Резервное копирование» находится в группе меню «Задачи», которая отображается пользователю с правом `rleMiniCACRL-w` (см. «Права и пункты меню Контрольной панели» в главе Описание ролевой модели (see page 47)). Права резервного копирования появились в версии 26.3.0. Если группа администраторов их не получает, дополните карту групп, как описано в подразделе «Добавление новых прав 26.3.0 в карту групп» раздела «Стандартные роли и матрица доступа».

Копия содержит закрытый ключ ЦС, если в задание включён каталог ключевого материала. Назначайте права резервного копирования и восстановления только администраторам ЦС.

7.6.2.2 Подготовка сервера сервиса резервного копирования

На сервере, где работает сервис резервного копирования, подготовьте:

- утилиты `pg_dump` и `pg_restore` из пакета клиента PostgreSQL той же основной версии, что сервер PostgreSQL БД ЦС: без утилит сервис не запускается, а утилиты другой версии могут не восстановить дамп;
- каталог копий (параметр `StoragePath`, например `/opt/itc/itc/backups`) на разделе с запасом места под копии за весь срок хранения; операция не начинается, если в каталоге свободно меньше `MinFreeSpaceBytes` (по умолчанию 1 ГиБ);
- доступ к каталогу копий только для учётной записи, от которой работает служба сервиса: копии содержат ключевой материал ЦС;
- файл конфигурации `/opt/itc/itc/config/itc_api_miniCA_backup_config.json` с правами 0600: при более широких правах сервис не запускается;

- сертификат TLS в формате PFX для канала askpass (параметры `SshAskPass.CertificatePath` и `SshAskPass.CertificatePassword`): по этому каналу узел ЦС получает от сервиса пароль для sudo через обратный туннель SSH;
- свой порт askpass (параметр `SshAskPass.LocalPort`, например 5226), если сервис работает на сервере Контрольной панели: серверная часть Контрольной панели тоже занимает локальный порт askpass, у обеих служб по умолчанию 5225;
- секцию `OpenId` для проверки токенов пользователей, а если пароли заданы ссылками EXC, также секцию `Ess`.

В конфигурации серверной части Контрольной панели укажите адрес сервиса в параметре `Services.BackupUrl`, например `http://localhost:8409`, если сервис работает на том же сервере. Пока адрес не задан, раздел «Резервное копирование» не работает. Параметры и пример конфигурации приведены в подразделе «Сервис резервного копирования» раздела Настройка вспомогательных сервисов (see page 139).

7.6.2.3 Подготовка узлов ЦС и сети

Что понадобится на каждом узле ЦС до создания задания:

- вход по SSH с паролем: сервис подключается по имени пользователя и паролю, вход по ключу не поддерживается;
- право sudo у пользователя SSH: sudo запрашивает его собственный пароль (сервис передаёт ему пароль SSH) или не запрашивает пароль (NOPASSWD);
- выключенный параметр `requiretty` в sudoers: сервис выполняет команды без терминала;
- разрешённый в sshd проброс TCP (параметр `AllowTcpForwarding` не равен `no`): через обратный туннель SSH узел получает пароль для sudo;
- утилиты `bash`, `curl`, GNU `find` и `base64`;
- режим linger технологической учётной записи, если служба ЦС работает как пользовательский юнит: сервис управляет службой через пользовательскую сессию systemd этой учётной записи.

Через sudo сервис выполняет на узле команды `find`, `base64`, `mkdir`, `mv`, `chmod` и `chown` от имени root. Пользовательским юнитом службы ЦС он управляет от имени технологической учётной записи (`sudo -u <учётная запись> env XDG_RUNTIME_DIR=/run/user/<uid> systemctl --user …`), системным юнитом от имени root (`sudo systemctl …`). Если права sudo ограничены списком команд, разрешите эти команды.

Что понадобится в сети и в БД:

- сетевой доступ с сервера сервиса резервного копирования к узлам ЦС по SSH (порт из задания, по умолчанию 22/tcp);
- сетевой доступ с сервера сервиса резервного копирования к серверу БД ЦС (порт из задания, по умолчанию 5432/tcp);

- в `pg_hba.conf` сервера БД ЦС разрешено подключение пользователя БД ЦС с сервера сервиса резервного копирования;
- пароль пользователя БД ЦС (пользователь по умолчанию `cwica`) либо ссылка на секрет во внешнем хранилище секретов (EXC).

Пользователь SSH обычно не совпадает с владельцем службы ЦС (поле «Пользователь сервиса»). Сервис входит на узел под учётной записью администратора, а службой ЦС управляет от имени её владельца, технологической учётной записи (по умолчанию `itc-svc`).

7.6.2.4 Создание задания

1. Откройте «Задачи → Резервное копирование» и нажмите **Добавить задание**.
2. Заполните общие параметры и блок «База данных ЦС».
3. Нажмите **Добавить узел** и заполните параметры узла; повторите для каждого узла ЦС.
4. Нажмите **Сохранить**.

Поле	Что указать	Значение для установки по умолчанию
Ключ*	Уникальный ключ задания; после создания не меняется	Например, `root-ca-daily`
Описание*	Короткий комментарий, виден только в интерфейсе	Произвольный текст
Cron*	Расписание копирования в формате cron из пяти полей. Время расписания считается в UTC	`0 2 * * *`, то есть каждый день в 02:00 UTC (подставляется по умолчанию)
База данных ЦС: Хост*, Порт	Адрес и порт сервера БД ЦС	`<адрес сервера БД ЦС>`, порт 5432
База данных ЦС: Имя БД*	Имя БД ЦС	`cwica`
База данных ЦС: Пользователь*, Пароль*	Учётная запись БД ЦС и её пароль. Вместо пароля можно указать ссылку EXC вида `mountPoint:path:key` и включить «Пароль из EXC»	Пользователь `cwica`
Имя узла*	Произвольное обозначение узла, уникальное в задании	Например, `node1`

Бинарник*	Полный путь к исполняемому файлу службы ЦС на узле. Файл входит в копию и восстанавливается вместе с файлами конфигурации	`/opt/itc/minica/minica`
systemd-юнит	Имя юнита службы ЦС	`minica` (подставляется по умолчанию)
Пользователь сервиса	Владелец пользовательского юнита службы ЦС	`itc-svc` (подставляется по умолчанию)
systemctl --user	Служба ЦС запущена как пользовательский юнит. Для системного юнита выключите	Включено (по умолчанию)
SSH: Хост*, Порт	Адрес узла ЦС и порт SSH	`<адрес сервера ЦС>`, порт 22
SSH: Пользователь*, Пароль*	Учётная запись с правом sudo и её пароль. Пароль используется и для sudo. Вместо пароля можно указать ссылку EXC и включить «Пароль из EXC»	Учётная запись администратора узла
Файлы конфигурации*	Полные пути к файлам и каталогам узла, по одному в строке. Каталоги копируются со всем содержимым, символьные ссылки не копируются. Каталог ключевого материала укажите раньше каталога настроек: в этом порядке файлы восстанавливаются	См. пример ниже

БД ЦС задаётся один раз на задание и общая для всех его узлов. Пример поля «Файлы конфигурации» для установки ЦС в каталог `/opt/itc/minica`:

```
/opt/itc/minica/ca
/opt/itc/minica/conf
/opt/itc/minica/minica.env
/opt/itc/minica/config-jwt.env
/opt/itc/minica/config-mclient.env
```

Задание сохранено, если появилось уведомление «Настройки сохранены» и задание есть в списке. Пока не добавлен хотя бы один узел, задание не сохраняется. При изменении задания ключ заблокирован, а пустые поля паролей и пользователя БД сохраняют прежние значения. Если переключить «Пароль из EXC», пароль или ссылку EXC нужно ввести заново.

Сервис сохраняет задание в файле `itc_api_miniCA_backup_config.runtime.json` на своём сервере. Пароли в ответах сервиса не возвращаются, поэтому в окне изменения поля паролей пустые. Пароль, заданный ссылкой EXC, в файле не хранится: в нём остаётся только ссылка, а пароль сервис получает из EXC в момент операции.

7.6.2.5 Состав копии и порядок копирования

Шаги по БД выполняются один раз за запуск, шаги по узлам выполняются для каждого узла задания по очереди:

Шаг	Где выполняется	Что делает
Копирование базы данных	Сервер сервиса резервного копирования	Утилита `pg_dump` снимает дамп БД ЦС в формате custom
Подсчёт записей	Сервер сервиса резервного копирования	Считает записи в таблицах сертификатов (`cert`), запросов (`csr`) и событий (`evt`) БД ЦС. Эти числа показываются при подтверждении восстановления
Копирование файлов конфигурации	Узел ЦС, через sudo по SSH	Забирает файлы и каталоги из поля «Файлы конфигурации» вместе с правами и владельцем
Копирование бинарника сервиса	Узел ЦС, через sudo по SSH	Забирает исполняемый файл ЦС из поля «Бинарник» вместе с правами и владельцем

После шагов сервис упаковывает данные в архив `tar.gz` , при включённом шифровании шифрует его и проверяет пригодность копии: целостность каждого файла по контрольной сумме SHA-256. Затем он удаляет копии с истёкшим сроком хранения (см. «Хранение копий» ниже). Эти действия отдельными шагами не показываются.

Особенности копирования:

- служба ЦС на узлах при копировании не останавливается;
- символьные ссылки не копируются, поэтому в поле «Файлы конфигурации» указывайте реальные пути;
- перед началом сервис проверяет свободное место в каталоге копий: если свободно меньше `MinFreeSpaceBytes` , копирование не начинается;
- обработка одного узла ограничена временем `OperationTimeoutMinutes` (по умолчанию 60 минут).

В копию входят только БД ЦС, файлы и каталоги из поля «Файлы конфигурации» и исполняемый файл ЦС. Архивная БД, БД OpenID и конфигурации сервера служб, в том числе конфигурация самого сервиса резервного копирования и ключ шифрования копий, в задание не входят. Их копирует администратор средствами СУБД и ОС (см. «Ручное резервное копирование»).

7.6.2.6 Запуск и контроль

Задание запускается по расписанию cron, время расписания считается в UTC. Чтобы запустить задание вне расписания, откройте карточку задания, нажмите **Запустить копирование** и подтвердите запуск (право `rleMiniCABackup-w`). Пока операция выполняется, вместо этой кнопки отображается неактивная кнопка **Выполняется**.

Для одного ЦС одновременно выполняется одна операция: копирование или восстановление. Если в момент запуска по расписанию по заданию уже идёт операция, этот запуск пропускается. Восстановление сервис выполняет только одно за раз, даже для разных ЦС.

В карточке задания:

- блок «Текущая операция» показывает шаги по БД и по каждому узлу, их статусы и ошибки; сведения обновляются автоматически;
- блок «Последний запуск» показывает статус, время начала и результат последнего копирования и последнего восстановления; время указано в UTC;
- таблица «Копии» перечисляет созданные копии и их пригодность: «Пригодна» или «Непригодна» с причиной.

Статусы шагов и операций: «Ожидает», «Выполняется», «Завершено», «Неуспех» (рядом выводится текст ошибки), «Возвращено к сохранённому состоянию», «Пропущено».

Уведомления по почте сервис отправляет только о неуспешном копировании или восстановлении. Их включает секция `Mailing` конфигурации сервиса резервного копирования, а не Контрольной панели. Ход каждой операции сервис записывает в свой журнал (см. главу Журналирование и мониторинг (see page 256)).

Проверка результата: после первого запуска последний запуск должен завершиться со статусом «Завершено», а новая копия в таблице «Копии» должна быть помечена «Пригодна». Если шаг завершился со статусом «Неуспех», прочитайте текст ошибки и проверьте требования из подразделов «Подготовка сервера сервиса резервного копирования» и «Подготовка узлов ЦС и сети»: доступность узла по SSH и сервера БД с сервера сервиса, пароль, право sudo, разрешение в `pg_hba.conf`. Типовые ошибки разобраны в разделе «Резервное копирование и восстановление» главы «Диагностика и устранение неисправностей».

7.6.2.7 Хранение копий

Копии хранятся на сервере сервиса резервного копирования, в каталоге из параметра `StoragePath`. Каждая копия состоит из архива `<StandId>_<ключ задания>_<время создания UTC>.tar.gz` (при шифровании `.tar.gz.enc`) и файла сведений `<имя`

`архива>.meta.json` с результатом проверки пригодности. В подкаталоге `state` сервис хранит историю операций, в подкаталоге `work` временные файлы выполняемой операции.

Скачать копию через интерфейс нельзя, её забирают с диска сервера сервиса. Чтобы копии сохранились при отказе этого сервера, регулярно переносите их на отдельный носитель.

Срок хранения. После каждого успешного копирования сервис удаляет копии этого задания старше `RetentionDays` дней (по умолчанию 30). Последняя пригодная копия сохраняется всегда. Если копирование завершилось неуспехом, старые копии не удаляются.

Шифрование. По умолчанию копии не шифруются. Копия содержит ключевой материал ЦС, поэтому незашифрованную копию защищайте так же, как ключ ЦС: доступ к каталогу копий и к перенесённым копиям только у администраторов ЦС. Шифрование AES-256-GCM включается так.

Шаг 1. На сервере сервиса от имени учётной записи, под которой работает его служба, создайте файл ключа (32 случайных байта в base64) вне каталога копий и закройте к нему доступ:

```
openssl rand -base64 32 > /opt/itc/itc/certs/backup-aes.key
chmod 600 /opt/itc/itc/certs/backup-aes.key
```

Шаг 2. В файле `itc_api_miniCA_backup_config.json` задайте `"Encryption": { "Enabled": true, "KeyFile": "/opt/itc/itc/certs/backup-aes.key" }` и перезапустите службу сервиса резервного копирования.

Зашифрованные копии получают расширение `.enc`, при проверке и восстановлении сервис расшифровывает их сам. Без файла ключа зашифрованную копию восстановить нельзя, поэтому храните копию ключа отдельно от копий (см. «Что копировать»). После смены ключа сохраняйте прежний ключ, пока нужны копии, зашифрованные им.

7.6.2.8 Восстановление ЦС из копии

Восстановление заменяет данные

При восстановлении служба ЦС на выбранных узлах останавливается, а БД ЦС и файлы выбранных узлов, включая исполняемый файл ЦС, заменяются данными копии. Записи, появившиеся после создания копии, удаляются. Сертификаты, отозванные после создания копии, после восстановления снова числятся действительными. Отзовите их повторно. Вместе с исполняемым файлом возвращается версия ядра ЦС на момент создания копии.

Перед запуском пользователь подтверждает восстановление, а сервис показывает, сколько записей сертификатов, запросов и событий будет потеряно: разницу между текущей БД ЦС и копией. Восстановить можно только копию, которая прошла проверку пригодности, и перед восстановлением сервис проверяет её повторно. Если набор узлов в задании отличается от набора узлов в копии, сервис сообщает о расхождении и требует повторного подтверждения.

БД ЦС одна на все узлы задания и восстанавливается при любом выборе узлов. От выбора узлов зависит, на каких узлах останавливается служба ЦС и заменяются файлы.

Восстановление выполняется по шагам:

Шаг	Где выполняется	Что делает
Снимок текущей БД	Сервер сервиса резервного копирования	Снимает дамп текущей БД ЦС для отката
Остановка сервиса ЦС	Каждый выбранный узел	Останавливает службу ЦС
Проверка останова сервиса	Каждый выбранный узел	Проверяет, что служба остановлена
Снимок файлов узла	Каждый выбранный узел	Сохраняет текущие файлы конфигурации и исполняемый файл ЦС для отката
Восстановление базы данных	Сервер сервиса резервного копирования	Заменяет данные БД ЦС данными копии командой `pg_restore --clean --if-exists`
Восстановление файлов узла	Каждый выбранный узел	Восстанавливает файлы в порядке поля «Файлы конфигурации», затем исполняемый файл ЦС, с исходными правами и владельцем
Запуск сервиса ЦС	Каждый выбранный узел	Запускает службу ЦС
Проверка запуска сервиса	Каждый выбранный узел	Ждёт до 30 секунд, пока служба перейдёт в состояние `active`

Сначала сервис снимает снимок текущей БД. Затем на всех выбранных узлах по очереди останавливается служба и сохраняется снимок файлов, после этого восстанавливается БД, а затем на каждом узле по очереди восстанавливаются файлы и запускается служба. Если шаг завершился ошибкой, сервис откатывает восстановление: возвращает БД и файлы узлов из снимков и снова запускает службы ЦС. Узел, возвращённый к снимку, получает статус «Возвращено к сохранённому состоянию», а операция завершается со статусом «Неуспех».

Что учесть после восстановления:

- команду `daemon-reload` сервис не выполняет: если среди путей задания есть файл юнита, выполните на узле `systemctl --user daemon-reload` (для системного юнита `sudo systemctl daemon-reload`) и перезапустите службу ЦС;

- файлы на узле, которых нет в копии, не удаляются;
- на Astra Linux команды установки мандатных меток (MAC LABEL), на которые у пользователя БД нет прав, пропускаются: восстановление завершается успешно, а в журнал сервиса записывается предупреждение.

Порядок действий (право `rleMiniCABackupRestore-w`):

1. В карточке задания найдите копию с отметкой «Пригодна» в таблице «Копии» и нажмите **Восстановить** в её строке.
2. В окне «Восстановление копии» оставьте отмеченными нужные узлы; в списке только узлы, которые есть и в задании, и в копии.
3. Проверьте таблицу «Число записей, которые будут удалены при восстановлении» (столбцы «Таблица», «Сейчас», «В копии», «Будет удалено»).
4. Отметьте «Подтверждаю: данные выбранных узлов будут заменены данными копии» и нажмите **Восстановить**.

Кнопка **Восстановить** видна, только пока по заданию не выполняется другая операция. Если сервер сообщает о расхождении, в окне появляется блок «Расхождение: …» с описанием и флажок «Продолжить несмотря на расхождение». Чтобы всё равно восстановить копию, отметьте флажок и снова нажмите **Восстановить**; чтобы отказаться, нажмите **Отмена**.

Проверка результата:

- в блоке «Последний запуск» восстановление завершилось со статусом «Завершено» (после завершения операции блок «Текущая операция» пуст);
- на каждом восстановленном узле `systemctl --user is-active minica.service` выводит `active`, а `mclient ping` отвечает версией ЦС и SKI его ключа; версия соответствует моменту создания копии;
- на странице «Настройки → Конфигурация» индикатор карточки «Конфигурация Clearway CA» зелёный.

7.6.3 Ручное резервное копирование

Ручное копирование нужно для данных, которые задания резервного копирования не охватывают: архивной БД, БД OpenID и конфигураций сервера служб, в том числе конфигурации сервиса резервного копирования. Кроме того, оно даёт независимую копию ЦС. Процедуру рекомендуется автоматизировать (cron) и хранить копии на отдельном носителе. Каталог `/backup` в примерах условный, используйте свой.

7.6.3.1 Базы данных

Для рабочей БД ЦС (по умолчанию `cwica`), архивной БД (`cwica_archive`) и БД OpenID (`openid`) используются утилиты PostgreSQL. Команды выполняются на любом сервере с клиентом PostgreSQL и сетевым доступом к серверу БД; вместо `<host>` укажите адрес сервера

соответствующей БД. Ключ `-Fc` создаёт копию в формате, который восстанавливает `pg_restore`.

```
pg_dump -h <host> -U cwica -Fc cwica > /backup/cwica_$(date +%F).dump
pg_dump -h <host> -U cwica -Fc cwica_archive > /backup/cwica_archive_$(date +%F).dump
pg_dump -h <host> -U cwi -Fc openid > /backup/openid_$(date +%F).dump
```

В командах указаны владельцы БД `cwica` и `cwi`, которых установщик создаёт по умолчанию. Если при установке заданы другие имена, подставьте их. Для запуска по расписанию храните пароли в файле `~/.pgpass` с правами 0600.

Выгрузка всего кластера PostgreSQL, включая роли и права:

```
pg_dumpall -h <host> -U postgres > /backup/cluster_$(date +%F).sql
```

Ниже приведён порядок восстановления БД ЦС из копии, созданной `pg_dump` с ключом `-Fc`. Восстановление заменяет текущие данные БД, а на его время ЦС недоступен.

Шаг 1. На каждом узле ЦС в сеансе `itc-svc` остановите службу ЦС: `systemctl --user stop minica.service`.

Шаг 2. Восстановите БД. Вместо `<дата>` подставьте дату копии из имени файла.

```
pg_restore -h <host> -U cwica -d cwica --clean /backup/cwica_<дата>.dump
```

Шаг 3. Запустите службу ЦС (`systemctl --user start minica.service`) и проверьте её ответ командой `mclient ping`.

Копия кластера, созданная `pg_dumpall`, восстанавливается командой `psql -h <host> -U postgres -f /backup/cluster_<дата>.sql postgres`. Команда рассчитана на новый (пустой) кластер PostgreSQL. Чтобы восстановить отдельную БД в работающем кластере, используйте копию `pg_dump` и `pg_restore`.

> **Хранение копий БД**
>
> Резервные копии БД содержат сведения о выпущенных сертификатах и служебные данные. Храните их в защищённом месте с ограниченным доступом.

7.6.3.2 Каталоги ключей и конфигураций

Файлы из таблицы «Что копировать» архивируются командой `tar` на каждом сервере. Команды выполняются с правами sudo: часть файлов доступна для чтения только технологической учётной записи. Если какого-то компонента на сервере нет, уберите его пути из команды.

Сервер ЦС:

```
sudo tar czf /backup/minica_conf_$(date +%F).tgz \
  /opt/itc/minica/ca /opt/itc/minica/conf \
  /opt/itc/minica/minica.env /opt/itc/minica/config-jwt.env /opt/itc/minica/config-
mclient.env \
  /home/itc-svc/.config/systemd/user
```

Сервер служб (Контрольная панель, публикация, mOCSP, архиватор, SCEP, CDP, MS-WSTEP, сервис резервного копирования). Файлы `/opt/itc/itc_wstep_env.conf` и `/etc/krb5.keytab` нужны, только если установлен MS-WSTEP. Каталог `/opt/itc/itc/config` содержит и конфигурацию сервиса резервного копирования с заданиями, а ключ шифрования копий архивируйте отдельно от копий, которые он защищает:

```
sudo tar czf /backup/services_conf_$(date +%F).tgz \
  /opt/itc/itc/config /opt/itc/itc/certs /opt/itc/itc/env /opt/itc/itc/nginx /opt/itc/
itc/sqlite \
  /opt/itc/itc_svc_env.conf /opt/itc/ocsp/mocsp.env /opt/itc/ocsp/conf /opt/itc/web/
nginx.conf \
  /opt/itc/itc_wstep_env.conf /etc/krb5.keytab \
  /home/itc-svc/.config/systemd/user
```

Сервер OpenID:

```
sudo tar czf /backup/openid_conf_$(date +%F).tgz \
  /opt/itc/openid/config.yaml /opt/itc/itc/certs /home/itc-svc/.config/systemd/user
```

Перед обновлением продукта для возможности отката дополнительно архивируйте каталоги приложений. Команда перечисляет каталоги всех компонентов; уберите из неё пути, которых на сервере нет:

```
sudo tar czf /backup/apps_$(date +%F).tgz \
  /opt/itc/itc.minica.* /opt/itc/itc.acm.* \
  /opt/itc/ocsp/mocsp /opt/itc/openid/openid \
  /opt/itc/minica/minica /opt/itc/minica/mclient /opt/itc/minica/bin
```

Для восстановления остановите службы сервера и распакуйте архив в корень файловой системы. Ключ `p` сохраняет права доступа файлов. Команда перезаписывает текущие файлы.

```
sudo tar xzpf /backup/minica_conf_<дата>.tgz -C /
```

Затем в сеансе `itc-svc` выполните `systemctl --user daemon-reload`, запустите службы сервера и проверьте их командой `systemctl --user is-active <юнит>` (ожидается `active`). На сервере ЦС проверьте также ответ ЦС командой `mclient ping`. Если восстановлены конфигурации Nginx, примените их командой `sudo nginx -t && sudo systemctl reload nginx`.

> **Ключи ЦС и секреты в копиях**
>
> Закрытые ключи ЦС (`ca.key`, `encr.key`) составляют самый критичный актив. Храните их резервные копии в зашифрованном виде и отдельно от резервных копий БД. Без `encr.key` данные сертификатов, зашифрованные в БД, после восстановления не расшифровать. Копии конфигураций содержат пароли БД и другие секреты. Ограничьте доступ к ним так же, как к копиям ключей.

7.6.3.3 Журналы

Журналы служб включают в резервные копии для последующего анализа:

- ядро ЦС: `/opt/itc/minica/logs/` (`minica.log`, `security.log`);
- mOCSP: `/opt/itc/ocsp/logs/mocsp.log`;
- .NET-службы: `/opt/itc/itc/logs/<служба>/` (`itc_api_miniCA_controlPanel`, `itc_api_miniCA_publication`, `itc_api_miniCA_archiver`, `itc_api_miniCA_backup`, `itc_acm_scepAdapter_clearwayCa`);
- MS-WSTEP: `/opt/itc/itc/logs/itc_acm_wstep_cep/`, `/opt/itc/itc/logs/itc_acm_wstep_cesAdapter_clearwayCa/`.

Чтобы включить журналы в копию, добавьте их каталоги в команду `tar` своего сервера, например `/opt/itc/minica/logs` на сервере ЦС, `/opt/itc/ocsp/logs /opt/itc/itc/logs` на сервере служб.

Ротацию и число хранимых файлов журнала задают параметры секции журналирования в конфигурационном файле службы, см. главу Журналирование и мониторинг (see page 256).

7.7 Архивирование истекших сертификатов

Раздел описывает, как работает архиватор, как изменить его расписание и как проверить, что архивирование выполняется.

Архиватор (служба `archiver.service`, приложение `/opt/itc/itc.minica.archiver/ITC.Api.MiniCA.Archiver`, порт 9507) по расписанию переносит истекшие сертификаты из рабочей БД ЦС в отдельную архивную БД. Вместе с сертификатами переносятся запросы и события. Так рабочая БД не разрастается и операции ЦС выполняются быстрее, а история выпуска сохраняется для последующего обращения.

По умолчанию задание `ClearwayCA` раз в сутки в 00:00 UTC (`@daily`) переносит данные из БД `cwica` в архивную БД `cwica_archive`. Архивную БД установщик создаёт тем же компонентом «База данных», что и рабочую. Параметры задания описаны в разделе Настройка вспомогательных сервисов (see page 139), подраздел «Архиватор».

Архиватору нужен сетевой доступ к серверу рабочей БД ЦС и к серверу архивной БД (по умолчанию 5432/tcp).

7.7.1 Изменение расписания и подключения к БД

Шаг 1. На сервере архиватора в сеансе `itc-svc` откройте файл `/opt/itc/itc/config/itc_api_miniCA_archiver_config.json` и измените параметры задания. `Cron` задаёт расписание в формате cron (время считается в UTC), `Source` и `Target` задают строки подключения к рабочей и архивной БД в формате Npgsql. Пример файла приведён в шаге 4 раздела «Развертывание Архиватора», а параметры задания описаны в подразделе «Архиватор» раздела Настройка вспомогательных сервисов (see page 139).

Файл содержит пароли двух БД. Требования к правам доступа на такие файлы приведены в разделе Ограничение прав доступа к файлам (see page 178).

Шаг 2. Перезапустите службу и проверьте её состояние:

```
systemctl --user restart archiver.service
systemctl --user status  archiver.service
```

Служба работает, если в выводе `status` есть строка `Active: active (running)`.

7.7.2 Проверка работы

В начале и в конце каждого прогона архиватор записывает в журнал сообщения с ключом задания и числом перенесённых записей. Найти их можно в сеансе `itc-svc`:

```
grep -r "архивации" /opt/itc/itc/logs/itc_api_miniCA_archiver/
```

Архивирование выполняется, если в журнале есть записи с текстом «Начало архивации» и «Конец архивации», ключом задания `ClearwayCA` и числом записей.

Резервное копирование архивной БД

Архивная БД хранит историю выпуска и копируется так же, как рабочая, командой `pg_dump` (см. «Резервное копирование»). Задание резервного копирования Контрольной панели копирует только БД ЦС.

7.8 Периодический контроль целостности исполняемых файлов

Раздел описывает, как вовремя обнаружить несанкционированное изменение или повреждение программных компонентов. Для этого контрольные суммы исполняемых файлов служб периодически сверяют с эталонными значениями, зафиксированными сразу после установки или обновления продукта.

Контроль выполняется на каждом сервере для компонентов, которые на нём установлены:

Компонент	Что контролировать
Ядро ЦС	`/opt/itc/minica/minica`, `/opt/itc/minica/mclient`, `/opt/itc/minica/bin/mjwt`
OCSP-ответчик mOCSP	`/opt/itc/ocsp/mocsp`
Сервис OpenID	`/opt/itc/openid/openid`
Контрольная панель	Каталоги `/opt/itc/itc.minica.controlPanel/` (серверная часть) и `/opt/itc/itc.minica.spa/` (веб-интерфейс)
Сервис публикации	Каталог `/opt/itc/itc.minica.publication/`
Архиватор	Каталог `/opt/itc/itc.minica.archiver/`
Адаптер SCEP	Каталог `/opt/itc/itc.acm.scepAdapter.clearwayCa/`
Адаптер MS-WSTEP	Каталоги `/opt/itc/itc.acm.wstep.cep/` и `/opt/itc/itc.acm.wstep.cesAdapter.clearwayCa/`

.NET-службы поставляются самодостаточными сборками: кроме основного исполняемого файла, в каталоге приложения лежат библиотеки. Поэтому для них контролируется весь каталог.

Шаг 1. Сразу после установки или обновления сформируйте на сервере файл эталонных контрольных сумм. Перечислите в команде пути компонентов этого сервера из таблицы. Команду выполняйте от учётной записи администратора сервера (или `itc-svc`) в её домашнем каталоге: файл эталона создаётся в текущем каталоге. Пример для сервера ЦС:

```
find /opt/itc/minica/minica /opt/itc/minica/mclient /opt/itc/minica/bin/mjwt \
  -type f -exec sha256sum {} + | sort -k 2 > checksums-$(hostname -s).sha256
```

Пример для сервера служб:

```
find /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 \
  -type f -exec sha256sum {} + | sort -k 2 > checksums-$(hostname -s).sha256
```

Шаг 2. Перенесите файл эталона в защищённое хранилище вне сервера, например вместе с резервными копиями. Не храните единственный экземпляр эталона рядом с контролируемыми файлами.

Шаг 3. Периодически, например по расписанию cron, проверяйте файлы по эталону. Скопируйте эталон на сервер и выполните:

```
sha256sum -c --quiet checksums-<сервер>.sha256
```

С ключом `--quiet` команда выводит только расхождения. Пустой вывод и код завершения 0 означают, что файлы не изменились. Строка со статусом `FAILED` означает, что файл изменён, а `FAILED open or read` сообщает, что файла нет.

Проверка по эталону не обнаруживает файлы, появившиеся в каталогах после его создания. Чтобы найти и их, сформируйте текущий список той же командой, что в шаге 1, но в файл `current.sha256`, и сравните его с эталоном: `diff checksums-<сервер>.sha256 current.sha256`. Пустой вывод `diff` означает отсутствие изменений.

Любое расхождение контрольных сумм требует расследования: сверьте файл с дистрибутивом, при необходимости восстановите его из проверенного источника или переустановите компонент из проверенного дистрибутива.

Обновление эталона

После каждого штатного обновления продукта формируйте эталон заново.

7.9 Типовые задачи замены сертификатов

Раздел описывает, как включить в Контрольной панели типовые задачи и что подготовить на серверах, где заменяются сертификаты.

Типовыми задачами называются пошаговые мастера замены TLS-сертификата на сервере nginx или PostgreSQL. Контрольная панель подключается к серверу по SSH и находит файлы сертификата и ключа. Затем она выпускает в ЦС новый сертификат по выбранному шаблону, записывает его в эти файлы и при необходимости перезапускает сервис. Мастер удобен для плановой замены: действия каждый раз выполняются одинаково.

7.9.1 Включение функции

Функцию включает параметр `Features.TypicalTask` в конфигурации серверной части Контрольной панели `/opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json`.

Параметр	Назначение	Значение по умолчанию
`Features.TypicalTask`	Включает типовые задачи. При `true` пункт меню «Задачи → Типовые» и мастера доступны, при `false` функция выключена	`true` (записывает установщик). Если секции `Features` в файле нет, функция выключена

Если функция выключена, пункта меню нет, а страница мастера по прямому адресу показывает «Функционал недоступен».

Порядок включения:

Шаг 1. На сервере Контрольной панели в сеансе `itc-svc` сохраните копию конфигурации и задайте параметр. Утилита `jq` установлена на сервере Контрольной панели установщиком. Перенаправление `>` перезаписывает содержимое файла, сохраняя его владельца и права доступа.

```
cd /opt/itc/itc/config
cp -p itc_api_miniCA_controlPanel_config.json
itc_api_miniCA_controlPanel_config.json.bak
jq '.Features.TypicalTask = true' itc_api_miniCA_controlPanel_config.json.bak >
itc_api_miniCA_controlPanel_config.json
jq .Features itc_api_miniCA_controlPanel_config.json   # ожидается: "TypicalTask": true
```

Шаг 2. Перезапустите серверную часть Контрольной панели. Пока служба перезапускается, Контрольная панель недоступна.

```
systemctl --user restart controlPanel.service
systemctl --user is-active controlPanel.service   # ожидается: active
```

Шаг 3. Обновите страницу Контрольной панели в браузере. Пользователю с нужными правами (см. ниже) в меню «Задачи → Типовые» доступны пункты «Замена сертификата nginx» и «Замена сертификата postgres».

Чтобы выключить функцию, задайте значение `false` тем же способом и перезапустите службу.

7.9.2 Требования

Права пользователя. Мастер доступен пользователю с правом `rleMiniCACertReq-w`. Пункт «Типовые» находится в группе меню «Задачи», которая отображается при праве `rleMiniCACRL-w` (см. «Права и пункты меню Контрольной панели» в главе Описание ролевой модели (see page 47)). Кроме того, у пользователя должен быть доступ к шаблону, по которому выпускается сертификат.

Учётная запись SSH на целевом сервере. Её имя и пароль вводят в мастере при каждой замене. Учётной записи нужны:

- право изменять файлы сертификата и ключа, а если они не существуют, создавать их по указанным путям;
- право перезапускать сервис nginx или PostgreSQL, если замена выполняется с перезапуском;
- вход по SSH с паролем: мастер подключается по имени пользователя и паролю.

Используйте отдельную учётную запись только с этими правами. Требования повторены на странице мастера в блоке «Требуются разрешения».

Сетевой доступ. С сервера Контрольной панели должен быть доступен целевой сервер по SSH на порт 22/tcp: отдельного поля для порта в мастере нет.

7.9.3 Порядок работы

Мастер проходит три шага:

1. «Подключение к серверу»: адрес сервера, учётная запись и пароль SSH; Контрольная панель проверяет, установлен ли nginx или PostgreSQL, и ищет сертификаты.
2. «Пути к сертификату и ключу»: выбор найденного сертификата или ручной ввод путей с проверкой наличия файлов.
3. «Параметры выпуска»: режим замены, шаблон сертификата и флажок перезапуска сервиса (отмечен по умолчанию).

Режимы замены: в режиме «Выпустить с новым ключом» (по умолчанию) создаётся новый закрытый ключ; в режиме «Обновить с тем же ключом» сертификат выпускается для существующего ключа, и этот режим доступен, только если файл ключа найден. При успехе

появляется уведомление «Сертификат успешно заменён. Серийный номер: …». Экраны мастера подробно описаны в Руководстве пользователя, раздел «Типовые задачи».

7.9.4 Проверка результата

Убедитесь, что сервер отдаёт новый сертификат: сравните его серийный номер с номером из уведомления. Команда `openssl` выводит серийный номер в шестнадцатеричном виде.

Сервер nginx:

```
openssl s_client -connect <адрес сервера>:443 -servername <fqdn> </dev/null 2>/dev/null
 | openssl x509 -noout -serial -enddate
```

Сервер PostgreSQL (OpenSSL 1.1.1 и новее):

```
openssl s_client -starttls postgres -connect <адрес сервера>:5432 </dev/null 2>/dev/
null | openssl x509 -noout -serial -enddate
```

Если флажок перезапуска был снят, сервер отдаёт прежний сертификат до перезапуска или перечитывания конфигурации сервиса.

7.9.5 Типовые проблемы

Что видно	Причина	Что сделать
В меню нет пункта «Задачи»	У пользователя нет права `rleMiniCACRL-w`	Назначьте право `rleMiniCACRL-w` через карту групп; учтите, что оно также разрешает обновлять CRL и запускать задачи публикации (см. «Права и пункты меню Контрольной панели»)
В группе «Задачи» нет пункта «Типовые»	Нет права `rleMiniCACertReq-w` или функция выключена	Проверьте права пользователя и параметр `Features.TypicalTask`
Страница мастера показывает «Функционал недоступен»	Функция выключена	Включите её, как описано в подразделе «Включение функции»

«Ошибка подключения к серверу» или текст ошибки SSH	Сервер недоступен по SSH на порт 22, неверные имя или пароль, вход по паролю запрещён	Проверьте сетевой доступ с сервера Контрольной панели, учётные данные и настройки SSH целевого сервера
«Ошибка проверки путей»	Не удалось проверить файлы на сервере	Проверьте подключение, пути к файлам и права учётной записи на каталоги
«Ошибка замены сертификата» или текст ошибки	Нет прав на запись файлов или перезапуск сервиса, нет доступа к шаблону	Проверьте права учётной записи SSH и доступ пользователя к шаблону

8 Журналирование и мониторинг

Глава помогает найти журналы Clearway CA, настроить их формат, объём и защиту от подделки, а также регулярно проверять, что ЦС и службы работают. Команды главы выполняются на сервере нужного компонента в сеансе технологической учётной записи itc-svc, если не указано иное. Из сеанса root в него входят командой `su - itc-svc`, а администратор с правом sudo входит командой `sudo su - itc-svc`.

Clearway CA ведёт журналы двух видов:

- **Журнал событий в базе данных ЦС**: таблица evt в БД ЦС с записями об обращениях к HTTP API центра сертификации.
- **Текстовые журналы служб**: записи о работе каждой службы в системном журнале journald, а у большинства служб также в файлах на сервере службы.

В журнал событий попадают выпуск, отзыв, проверка статуса и получение сертификатов, операции с CRL и Delta CRL, управление шаблонами и токенами, одобрение и отклонение запросов на выпуск и другие операции. Записи доступны в Контрольной панели, в разделе «События» (право `rleMiniCAEventList-r`), и через API ЦС. Типы событий и столбцы журнала описаны в Руководстве пользователя, в главе «Журнал событий».

Архиватор по расписанию переносит записи журнала событий вместе с запросами и сертификатами в архивную БД (по умолчанию cwica_archive), см. Настройка вспомогательных сервисов (see page 139).

У ядра ЦС два текстовых журнала: журнал службы minica.log и отдельный журнал событий безопасности security.log. Где искать текстовые журналы компонентов:

Компонент и юнит systemd	**Файловый журнал**	**Идентификатор в journald (SyslogIdentifier)**
Ядро ЦС `minica.service`	`/opt/itc/minica/logs/minica.log` (журнал службы) `/opt/itc/minica/logs/security.log` (события безопасности)	Не задан, записи выбираются по имени юнита
OCSP-ответчик mOCSP `mocsp.service`	`/opt/itc/ocsp/logs/mocsp.log`	Не задан, записи выбираются по имени юнита
Контрольная панель `controlPanel.service`	Каталог `/opt/itc/itc/logs/itc_api_miniCA_controlPanel/`	`itc_api_minica_controlPanel`

Сервис публикации `publication.service`	Каталог `/opt/itc/itc/logs/itc_api_miniCA_publication/`	`itc_api_minica_publication`
Архиватор `archiver.service`	Каталог `/opt/itc/itc/logs/itc_api_miniCA_archiver/`	`itc_api_minica_archiver`
Сервис резервного копирования служба сервиса резервного копирования	Каталог `/opt/itc/itc/logs/itc_api_miniCA_backup/`, имя сервиса в журнале `itc.api.miniCA.backup`	Задаётся в юните службы
SCEP-адаптер `scepAdapter.service`	Каталог `/opt/itc/itc/logs/itc_acm_scepAdapter_clearwayCa/`	`itc_scep`
MS-WSTEP, служба CEP `wstepCep.service`	Каталог `/opt/itc/itc/logs/itc_acm_wstep_cep/`	`itc_wstep_cep`
MS-WSTEP, служба CES `wstepCesAdapter.service`	Каталог `/opt/itc/itc/logs/itc_acm_wstep_cesAdapter_clearwayCa/`	`itc_wstep_ces`
Сервис аутентификации OpenID `openid.service`	Нет	`itc_api_openid`

Текстовые журналы ядра ЦС и OCSP-ответчика (службы, которая отвечает на запросы о статусе сертификатов) настраиваются параметрами, описанными в разделе «Параметры журналирования (секции log и rotate)». Изменённые параметры вступают в силу после перезапуска службы. Пример для ядра ЦС, на сервере ЦС:

```
su - itc-svc
systemctl --user restart minica.service
systemctl --user status minica.service
```

Для OCSP-ответчика так же перезапустите `mocsp.service` на его сервере. Управление службой ядра ЦС подробнее описано в разделе Управление службой (see page 138).

Перезапуск службы

На время перезапуска служба не обрабатывает запросы. Перезапускайте её в период обслуживания и после перезапуска проверяйте состояние командой `systemctl --user status <юнит>`.

8.1 Параметры журналирования (секции log и rotate)

Раздел описывает, куда служба пишет текстовый журнал, в каком формате и сколько его хранить. Параметры ядра ЦС задаются в секции `log` файла `/opt/itc/minica/conf/minica.yml`, а параметры ротации находятся во вложенной секции `log.rotate`. Ротацией называется замена заполненного файла журнала новым: старые файлы хранятся ограниченное время и в ограниченном числе. Такие же секции есть в файле `/opt/itc/ocsp/conf/mocsp.yml` OCSP-ответчика, их отличия приведены в таблице.

Служба выводит журнал двумя способами:

- в стандартный поток (stdout или stderr): записи принимает системный журнал journald, а сроки и объём хранения определяют настройки ОС (journald, logrotate);
- в файл на диске: объём файлов ограничивает встроенная ротация (секция `log.rotate`).

Способы можно использовать одновременно. В поставляемой конфигурации ядра ЦС включены оба: записи выводятся в stdout и пишутся в файл `/opt/itc/minica/logs/minica.log`.

Параметр	Назначение	По умолчанию в minica.yml	По умолчанию в mocsp.yml
`log.pipe`	Стандартный поток для записей: `stdout` или `stderr`; при значении `null` или пустой строке журнал в поток не выводится	`stdout`	`stdout`
`log.file`	Путь к файлу журнала; при пустой строке журнал в файл не записывается	`/opt/itc/minica/logs/minica.log`	`/opt/itc/ocsp/logs/mocsp.log`
`log.fileMode` (в mocsp.yml `log.file-mode`)	Права на создаваемый файл журнала в восьмеричной записи	`0600`	`0640`

`log.format`	Формат записей. В minica.yml: `json`, `logfmt`, `tinted`, `default`. В mocsp.yml: `std`, `text`, `json`, `tint`	`json`	`tint`
`log.level`	Минимальный уровень регистрируемых событий: `flood`, `trace`, `debug`, `info`, `notice`, `warn`, `error`, `crit`, `fatal`	`info`	`info`
`log.addKey` / `log.addValue`	Имя и значение поля, которое добавляется в каждую запись журнала	`app` / `minica`	Не заданы
`log.rotate.enable`	Включает ротацию файла журнала	`true`	`true`
`log.rotate.maxSize`	Размер файла в мегабайтах, при достижении которого выполняется ротация	`10`	`10`
`log.rotate.maxAge`	Срок хранения старых файлов журнала, дней	`365`	`10`
`log.rotate.maxBackups`	Наибольшее число хранимых старых файлов журнала	`100`	`100`
`log.rotate.localTime`	Использовать при ротации локальное время	`true`	`true`
`log.rotate.compress`	Сжимать старые файлы журнала (gzip)	`true`	`true`

Фрагмент секции `log` в minica.yml со значениями по умолчанию. Параметры контроля целостности описаны в разделе «Контроль целостности записей журнала», а вложенная секция `log.secLog` в разделе «Отказоустойчивый журнал событий безопасности (secLog)».

```
log:
```

```
  pipe: "stdout"                          # stdout | stderr | null | "" (не выводить в
поток)
  file: "/opt/itc/minica/logs/minica.log" # файл журнала; "" отключает запись в файл
  fileMode: "0600"                        # права на файл журнала
  format: "json"                          # json | logfmt | tinted | default
  level: "info"                           # flood | trace | debug | info | notice |
warn | error | crit | fatal
  rotate:
    enable: true                          # включить ротацию
    maxSize: 10                           # размер файла для ротации, МБ
    maxAge: 365                           # срок хранения старых файлов, дней
    maxBackups: 100                       # число хранимых старых файлов
    localTime: true                       # локальное время при ротации
    compress: true                        # сжимать старые файлы (gzip)
```

Как временно повысить уровень журнала для разбора инцидента, описано в главе «Диагностика и устранение неисправностей», в разделе «Статус служб и журналы».

8.1.1 Параметры журнала в командной строке службы

Уровень, формат, поток и файл журнала можно задать также параметрами командной строки `-log-level`, `-log-format`, `-log-pipe` и `-log-file`. Юнит передаёт их службе из переменной в файле окружения:

Служба	Файл окружения	Переменная
Ядро ЦС	`/opt/itc/minica/minica.env`	`MINICA_OPTIONS`
OCSP-ответчик	`/opt/itc/ocsp/mocsp.env`	`MOCSP_OPTIONS`

Уровень и формат журнала могут быть заданы и в конфигурационном файле, и в этих переменных, поэтому при настройке журнала проверяйте оба места. Для эксплуатации OCSP-ответчика достаточно уровня `info`. Файл окружения правят на сервере OCSP-ответчика в сеансе itc-svc; строка в `/opt/itc/ocsp/mocsp.env`:

```
MOCSP_OPTIONS="-log-level=info -log-format=tint"
```

После изменения файла окружения перезапустите службу и проверьте её состояние. На время перезапуска OCSP-ответчик не отвечает на запросы.

```
systemctl --user restart mocsp.service
systemctl --user status mocsp.service
```

8.1.2 Просмотр журналов

Службы работают как пользовательские юниты учётной записи itc-svc, поэтому записи в journald читайте из её сеанса командой `journalctl` с ключом `--user`. Команды с ключом `-f` выводят новые записи, пока их не прервут клавишами Ctrl+C, поэтому выполняйте их по одной.

На сервере ЦС, в сеансе itc-svc:

```
journalctl --user -u minica.service          # журнал службы ядра ЦС
journalctl --user -u minica.service -f       # новые записи в реальном времени; выход:
Ctrl+C
tail -f /opt/itc/minica/logs/minica.log      # файловый журнал ядра ЦС; выход: Ctrl+C
```

Последние 100 записей SCEP-адаптера выводит команда на сервере служб, в сеансе itc-svc:

```
journalctl --user -u scepAdapter.service -n 100
```

Файл minica.log создаётся с правами 0600, поэтому читайте его в сеансе itc-svc или через sudo. Другие приёмы работы с journalctl для диагностики (фильтр по уровню, записи с момента загрузки) приведены в главе «Диагностика и устранение неисправностей».

8.1.2.1 Кто получил одноразовый пароль SCEP

Одноразовый пароль (challenge password) клиент SCEP указывает в запросе на сертификат, и по нему SCEP-адаптер подтверждает право на выпуск. Пароль выдаёт SCEP-адаптер пользователю, вошедшему через сервис OpenID. Выдачу пароля адаптер отмечает в своём файловом журнале записью аудита (сообщение ScepAudit): в ней указаны время, имя пользователя, путь запроса `challenge-password` и код ответа HTTP (200 означает, что пароль выдан).

Журнал адаптера хранится на сервере служб в файлах `*.ndjson` каталога `/opt/itc/itc/logs/itc_acm_scepAdapter_clearwayCa/`, по одной записи JSON в строке. Записи о запросах паролей выбирает команда, выполняемая на сервере служб в сеансе itc-svc:

```
grep ScepAudit /opt/itc/itc/logs/itc_acm_scepAdapter_clearwayCa/*.ndjson | grep
challenge-password
```

По этим записям определяют, какой пользователь и когда получил пароль для выпуска сертификата по SCEP.

8.1.2.2 Журнал сервиса резервного копирования

Сервис резервного копирования ведёт файловый журнал в каталоге `/opt/itc/itc/logs/itc_api_miniCA_backup/` своего сервера. В журнал попадают:

- начало и завершение каждого шага копирования и восстановления с длительностью и текстом ошибки;
- итог операции и её инициатор: имя пользователя или расписание;
- командная строка запуска `pg_dump` и `pg_restore` без пароля;
- удаление копий с истёкшим сроком хранения;
- предупреждения, например о пропущенных командах мандатных меток (MAC LABEL) при восстановлении на Astra Linux.

Кроме того, сервис регистрирует события безопасности: начало, успешное и неуспешное завершение копирования и восстановления, остановку и запуск службы ЦС на узле при восстановлении, а также проверку токена пользователя и отказ в доступе при нехватке прав.

Ход и результат операций видны также в карточке задания в Контрольной панели, а письма о неуспешных операциях отправляются, если включена секция `Mailing` конфигурации сервиса (см. «Сервис резервного копирования» в разделе Настройка вспомогательных сервисов (see page 139)).

8.2 Контроль целостности записей журнала

Служебные поля записей позволяют обнаружить искажение, изъятие или подмену записей текстового журнала. К каждой записи могут добавляться два поля:

- `logId` : уникальный идентификатор записи в формате UUID v7 (RFC 9562); он однозначно определяет запись и упорядочивает записи по времени;
- `logSum` : контрольная сумма записи по алгоритму CRC16; если при проверке сумма не совпадает, содержимое записи искажено.

Контрольная сумма сохраняется либо в младших битах `logId` , либо отдельным полем `logSum` в шестнадцатеричном виде. В mocsp.yml способ выбирает параметр `log.sumAlone` .

В цепочном режиме (`log.sumChain`) контрольная сумма каждой записи вычисляется с учётом суммы предыдущей записи, и записи образуют связанную последовательность. Изъятие, добавление или перестановка записей нарушает цепочку и выявляется при проверке. Так обнаруживается систематическая потеря или подмена записей, которую независимые суммы отдельных записей не показывают.

Параметры задаются в секции `log` файлов minica.yml и mocsp.yml:

Параметр	Назначение	По умолчанию в minica.yml	По умолчанию в mocsp.yml
`log.idOff`	Отключает поле `logId` : при `true` идентификатор не формируется	`false` (logId формируется)	`false`

`log.sumOff`	Отключает расчёт контрольной суммы записи	`false` (сумма формируется)	`false`
`log.sumFull`	Вычислять контрольную сумму по всем атрибутам записи рекурсивно (более сложный алгоритм)	`false`	`false`
`log.sumChain`	Цепочный режим: сумма записи учитывает сумму предыдущей записи	`true`	`false`
`log.sumAlone`	Сохранять контрольную сумму отдельным полем `logSum`; при `false` сумма хранится в младших битах `logId`	Не задан	`false`

Пример записи в формате JSON, когда контрольная сумма сохраняется отдельным полем `logSum` (значения условные):

```
{
  "time": "2026-07-09T12:00:00Z",
  "level": "INFO",
  "msg": "certificate issued",
  "app": "minica",
  "logId": "0190e2a1-7c3d-7f8a-b2c1-9f4e6d5a1b20",
  "logSum": "a1b2"
}
```

Если контрольная сумма хранится в младших битах `logId`, отдельного поля `logSum` в записи нет.

Рекомендации

Цепочный режим включайте для журналов аудита с повышенными требованиями к защите от подделки. Чтобы сохранить прежние записи, включите каталоги журналов (см. таблицу в начале главы) в общий регламент резервного копирования (раздел «Резервное копирование»). При выводе журнала в стандартные потоки хранение обеспечивают средства ОС (journald, logrotate).

8.3 Отказоустойчивый журнал событий безопасности (secLog)

Ядро ЦС записывает события безопасности в отдельный файл `/opt/itc/minica/logs/security.log`. Механизм secLog сохраняет эти события даже при нехватке места на диске: заранее предупреждает о заполнении диска, удерживает события в памяти при сбое записи и оповещает администратора. Предупреждения о заполнении диска и оповещения работают, когда включены контроль занятости диска (`watchdogEnabled`) и уведомления (`notifyUiEnabled`, `notifyEmailEnabled`); в поставляемой конфигурации они выключены.

Параметры задаются во вложенной секции `log.secLog` файла `/opt/itc/minica/conf/minica.yml`; в таблицах раздела имена параметров указаны относительно неё. Механизм включается, когда задан путь `log.secLog.path`. В поставляемой конфигурации путь задан. Изменённые параметры применяются после перезапуска `minica.service` (на время перезапуска ЦС недоступен).

8.3.1 Общие параметры

Параметр	Назначение	Значение по умолчанию
`path`	Путь к файлу журнала безопасности. Если путь задан, механизм включён	`/opt/itc/minica/logs/security.log`
`failOnError`	Передавать ошибку записи журнала безопасности выполняемой операции. При `false` ошибка записи не приводит к сбою операции	`false`
`notifyDebug`	Расширенная отладка отправки уведомлений (вывод в stderr)	`false`
`serviceName` / `env`	Имя службы и имя среды, которые подставляются в уведомления	`minica` / пустая строка

8.3.2 Контроль занятости диска (watchdog)

Сторожевой механизм (watchdog) периодически проверяет занятость диска в каталоге журналов и заранее предупреждает о её исчерпании.

Параметр	Назначение	Значение по умолчанию

`watchdogEnabled`	Включает контроль занятости диска	`false`
`watchdogProbeInterval`	Период проверки, секунд	`20`
`watchdogWarnPercent`	Порог предупреждения (Warning) по занятости диска, %	`85`
`watchdogCritPercent`	Критический порог (Critical) по занятости диска, %	`95`

Порядок действий при срабатывании порогов описан в главе «Диагностика и устранение неисправностей», в разделе «Переполнение диска под журналы (watchdog журнала безопасности)».

8.3.3 Уведомления

При сбоях записи журнала и достижении критических порогов механизм оповещает администратора по двум независимым каналам: в веб-интерфейсе Clearway CA и по электронной почте.

Параметр	Назначение	Значение по умолчанию
`notifyUiEnabled`	Включает уведомления в веб-интерфейсе Clearway CA	`false`
`notifyUiEndpoint`	Адрес приёмника уведомлений для веб-интерфейса	Пустая строка
`notifyEmailEnabled`	Включает уведомления по электронной почте (SMTP)	`false`
`notifyEmailSmtp`	Параметры SMTP-сервера одной строкой в формате `host:port;username;password;recipient;authType`	Пустая строка
`notifyEmailFrom`	Адрес отправителя. Если не задан, берётся из `username`	Пустая строка
`notifyEmailTls`	Подключаться к SMTP-серверу по TLS	`false`

`notifyEmailSkipVerify`	Не проверять сертификат SMTP-сервера. Чтобы сертификат проверялся, задайте `false`	`true`
`notifyMinInterval`	Минимальный интервал между повторными уведомлениями, секунд; защищает от потока повторных сообщений	`30`

Пример: контроль занятости диска и уведомления по почте. Замените значения в угловых скобках своими. Строка `notifyEmailSmtp` содержит пароль почтовой учётной записи, поэтому доступ к minica.yml должен быть только у itc-svc (права 0600, см. Ограничение прав доступа к файлам (see page 178)).

```
log:
  secLog:
    path: "/opt/itc/minica/logs/security.log"
    watchdogEnabled: true
    notifyEmailEnabled: true
    notifyEmailSmtp: "<smtp-сервер>:<порт>;<пользователь>;<пароль>;<адрес
получателя>;<тип аутентификации>"
    notifyEmailFrom: "<адрес отправителя>"
    notifyEmailTls: true
    notifyEmailSkipVerify: false
```

8.3.4 Буфер событий в оперативной памяти

Если записать событие на диск не удаётся, оно не теряется: события временно удерживаются в кольцевом буфере в оперативной памяти и выгружаются в файл, когда запись восстановится.

Параметр	Назначение	Значение по умолчанию
`bufferMaxEvents`	Наибольшее число событий, удерживаемых в памяти при сбое диска	`50000`
`bufferMaxBytes`	Ограничение объёма буфера, байт	`52428800` (50 МБ)
`flushBackoffInitial` / `flushBackoffMax`	Начальный и наибольший интервал между попытками выгрузить буфер в файл, секунд	`60` / `60`
`preallocSizeMb`	Резервирование места под основной файл журнала при старте службы (fallocate), МБ; при `0` место не резервируется	`100`

8.3.5 Фильтры событий

Фильтры определяют, какие события безопасности регистрируются. Списки идентификаторов принимают отдельные значения (`"1005"`) и диапазоны (`"1000-1100"`).

Параметр	Назначение	Значение по умолчанию
`allowedEventIds`	Разрешающий список идентификаторов событий; при пустом списке регистрируются все события	`[]`
`deniedEventIds`	Запрещающий список идентификаторов; при пустом списке исключений нет	`[]`
`allowedCategories` / `deniedCategories`	Разрешающий и запрещающий списки категорий событий	`[]`
`criticalCategories`	Категории, которые регистрируются даже при деградации журнала (нехватке места на диске)	`[]`

> **Журнал безопасности есть только у ядра ЦС**
>
> Механизм secLog работает только в ядре ЦС (minica). У OCSP-ответчика mOCSP есть контроль целостности записей (параметры `log.idOff` , `log.sumOff` , `log.sumChain` , `log.sumAlone`), но нет отдельного файла security.log, контроля занятости диска и буфера событий.

8.4 Мониторинг и самодиагностика

Раздел описывает, как убедиться, что ЦС и службы работают: проверить доступность ЦС, получить показатели его работы, проверить состояние служб и настроить оповещения.

8.4.1 Проверка доступности ЦС

Утилита mclient проверяет, что служба ЦС запущена и принимает запросы. Выполните команды на сервере ЦС в сеансе itc-svc: этой учётной записи доступны ключ и токен клиента (`conf/mclient.key` , `conf/mclient.jwt`).

```
su - itc-svc
cd /opt/itc/minica
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
```

```
./mclient ping
```

При успехе команда завершается с кодом 0 и выводит версию ЦС и идентификатор ключа ЦС (SKI). Этой же командой установщик проверяет готовность ЦС после запуска службы. Если ping завершается ошибкой, порядок поиска причины описан в главе «Диагностика и устранение неисправностей».

8.4.2 Показатели работы ЦС

Показатели работы ЦС (сколько сертификатов и списков отзыва выпущено, сколько сертификатов, запросов и шаблонов хранится в БД ЦС) доступны двумя способами:

Способ	Когда рассчитываются показатели	Кому доступен
Страница «Статистика ЦС» в Контрольной панели	В фоне по расписанию; страница показывает результат последнего расчёта	Пользователю с правом `rleMiniCAConfig-r`
Команда `mclient stat`	В момент вызова	Учётной записи с доступом к ключу и токену mclient (itc-svc)

Если после расчёта по расписанию выпускались или отзывались сертификаты, значения на странице и в выводе команды могут различаться.

Страница «Статистика ЦС»:

- показывает показатели ЦС, выбранного на странице «Настройки → Конфигурация»;
- при открытии не запускает новый расчёт, а выводит результат последнего расчёта;
- внизу виджета «Статистика сертификатов» указывает время последнего расчёта («Данные собраны») и расписание расчёта («Обновление по расписанию»);
- в блоке «Информация о ЦС» выводит идентификатор ключа ЦС, версию ЦС и статус службы ЦС (UP, если служба работает, или DOWN, если не работает); эти сведения приходят вместе с результатом расчёта, поэтому текущее состояние службы проверяйте командой `mclient ping`;
- выводит «Статистика ещё не собрана», пока расчёта не было: показатели появятся после ближайшего расчёта по расписанию;
- выводит «Ошибка загрузки данных», если Контрольной панели не удалось получить показатели.

Подробное описание страницы и соответствие её показателей полям `mclient stat` приведены в Руководстве пользователя, в главе «Самодиагностика».

Команда `mclient stat` выполняется в том же сеансе, что и `mclient ping` (каталог /opt/itc/minica, задана переменная MCLIENT_CONF):

```
./mclient stat
```

Команда выводит одну строку с показателями в виде пар имя=значение. Пример:

```
2026/09/28 14:25:42.594 INFO minica stat errno=0 validCnt=11 revokedCnt=3 expiredCnt=0
unknownCnt=0 AllIssuedCertificates=14 csrCnt=14 crlCnt=21 templCnt=5 jwtCnt=3
evtCnt=465 dt=12.102487ms
```

Поле	Что означает
`errno`	Результат: `0` означает, что показатели получены без ошибок
`AllIssuedCertificates`	Сколько сертификатов ЦС выпустил за всё время работы
`validCnt` / `revokedCnt` / `expiredCnt`	Действующие, отозванные и истекшие сертификаты в БД ЦС
`unknownCnt`	Сертификаты со статусом UNKNOWN, не относящиеся к трём группам выше
`csrCnt`	Запросы на выпуск сертификатов в БД ЦС
`crlCnt`	Выпущенные списки отзыва CRL
`templCnt`	Шаблоны сертификатов
`evtCnt`	Записи журнала событий
`jwtCnt`	Токены JWT программных клиентов, учтённые в БД ЦС
`dt`	Время подсчёта показателей

Для обработки вывода программами, например системой мониторинга, используйте формат JSON: `./mclient -log-format json stat`. В этом формате время подсчёта `dt` указывается в наносекундах.

8.4.3 Состояние служб

Службы Clearway CA работают как пользовательские юниты systemd учётной записи itc-svc. Для учётной записи включён режим linger, поэтому службы запускаются при загрузке сервера и работают без входа пользователя в систему. Режим включает администратор с правом sudo командой `sudo loginctl enable-linger itc-svc` (см. Управление службой (see page 138)). Проверка на сервере службы:

```
su - itc-svc
systemctl --user status minica.service       # подробное состояние и последние записи
журнала
systemctl --user is-active minica.service    # краткий ответ для сценариев проверки
```

Команда `is-active` выводит `active` и завершается с кодом 0, если служба работает. Иначе она выводит другое состояние, например `inactive` или `failed`, и завершается с ненулевым кодом. Имена юнитов остальных служб приведены в таблице в начале главы.

В Контрольной панели состояние компонентов показывает страница «Настройки → Конфигурация» (право `rleMiniCAConfig-r`). Карточки «Конфигурация Clearway CA», «Конфигурация контрольной панели» и «Конфигурация сервиса публикации CRL» показывают версию компонента и индикатор: зелёный, если компонент отвечает, и красный, если не отвечает.

8.4.4 Проверка сервиса публикации и архиватора по HTTP

Сервис публикации и архиватор отвечают на запрос проверки работоспособности по пути `/api/health`. Выполните команды на сервере, где установлена служба:

```
curl -s -o /dev/null -w '%{http_code}\n' http://localhost:9407/api/health   # сервис
публикации
curl -s -o /dev/null -w '%{http_code}\n' http://localhost:9507/api/health   # архиватор
```

Сервис резервного копирования отвечает на тот же запрос на своём порту, например `curl -s -o /dev/null -w '%{http_code}\n' http://localhost:8409/api/health` на его сервере.

Код `200` означает, что служба запущена и отвечает на запросы. Если служба остановлена, curl не получает ответа и выводит `000`. При другом коде проверьте состояние службы и её журнал (таблица в начале главы). Через Nginx проверка сервиса публикации доступна по адресу `https://<fqdn>/api/publication/health`, где `<fqdn>` обозначает имя сервера служб.

8.4.5 Оповещения по электронной почте

Clearway CA отправляет по электронной почте оповещения трёх видов:

- о событиях жизненного цикла сертификатов (выпуске, отзыве и приближении срока истечения действия); адреса и события пользователи задают при выпуске сертификата и на вкладке «Подписки» карточки сертификата (см. Руководство пользователя);
- о сбоях журнала событий безопасности и заполнении диска (см. «Уведомления» в разделе «Отказоустойчивый журнал событий безопасности (secLog)»);
- о неуспешном резервном копировании и восстановлении ЦС; получателей и SMTP-сервер задаёт секция `Mailing` конфигурации сервиса резервного копирования (см. «Сервис резервного копирования» в разделе Настройка вспомогательных сервисов (see page 139)).

Все каналы однонаправленные: система только отправляет сообщения на внешний почтовый сервер. Для их работы нужен настроенный и доступный SMTP-сервер; если он не настроен или недоступен, оповещения не отправляются.

9 Диагностика и устранение неисправностей

Глава помогает найти и устранить причину сбоя: какими средствами проверить компоненты Clearway CA, как расшифровать код ответа API и что делать при типовых неисправностях.

Примеры приведены для установки по умолчанию:

- каталог ядра ЦС /opt/itc/minica, файл конфигурации /opt/itc/minica/conf/minica.yml;
- службы Clearway CA работают как пользовательские юниты systemd технологической учётной записи (по умолчанию itc-svc), ими управляют командами `systemctl --user`;
- команды без `sudo` выполняются на сервере компонента в сеансе itc-svc: из сеанса root в него входят командой `su - itc-svc`, из сеанса администратора с правом sudo командой `sudo su - itc-svc`;
- команды с `sudo` выполняет администратор из своего сеанса.

9.1 Средства диагностики

Начинайте проверку с ядра ЦС: к нему обращаются Контрольная панель, сервис публикации и адаптер SCEP. Затем проверьте службу компонента, в котором проявилась ошибка, и её журнал.

9.1.1 Проверка доступности ЦС (mclient)

Основным инструментом проверки ядра ЦС служит консольный клиент mclient (/opt/itc/minica/mclient). Он обращается к ЦС так же, как другие компоненты: предъявляет JWT (токен с ролью клиента) и подписывает запрос своим ключом. Запускайте его на сервере ЦС в сеансе itc-svc: этой учётной записи доступны ключ и токен клиента (conf/mclient.key, conf/mclient.jwt). Путь к конфигурации клиента задаёт переменная MCLIENT_CONF.

```
su - itc-svc
cd /opt/itc/minica
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
# связь с ЦС: TLS, проверка JWT, подписи запроса и времени
./mclient ping
# статистика ЦС: число выпущенных, отозванных и истекших сертификатов
./mclient stat
```

Успешный `mclient ping` завершается без ошибки и выводит версию ЦС и идентификатор его ключа (SKI). Это значит, что служба minica запущена, слушает свой порт (9443), а токен, подпись и время клиента приняты. Ошибка ping сужает поиск до одного уровня: сеть и порт, TLS, подпись и JWT или рассогласование времени. Определить уровень помогут разделы «Коды ответов API» и «Типовые проблемы и их устранение».

`mclient stat` выводит одну строку с показателями; значение `errno=0` означает, что показатели получены без ошибок. Пример вывода и назначение всех полей приведены в разделе «Показатели работы ЦС» главы Журналирование и мониторинг (see page 256).

9.1.2 Проверка из Контрольной панели

Часть проверок доступна в Контрольной панели, без входа на серверы:

- «Настройки → Конфигурация» (право `rleMiniCAConfig-r`): карточки «Конфигурация Clearway CA», «Конфигурация контрольной панели» и «Конфигурация сервиса публикации CRL» показывают индикатор состояния компонента: зелёный, если компонент отвечает (выводится номер его версии), и красный, если компонент не отвечает.
- «Статистика ЦС» (право `rleMiniCAConfig-r`): блок «Информация о ЦС» показывает статус сервиса ЦС: UP (работает) или DOWN (не работает).
- «События» (право `rleMiniCAEventList-r`): столбец «Код ошибки» показывает код результата операции ЦС, 0 означает успех (расшифровку см. в разделе «Коды ответов API»).

Сведения на странице «Статистика ЦС» обновляются вместе с расчётом статистики по расписанию. Текущее состояние ЦС проверяйте командой `mclient ping`.

9.1.3 Статус служб и журналы

Каждый компонент Clearway CA работает как отдельный пользовательский юнит systemd. Записи юнита попадают в системный журнал journald учётной записи itc-svc; у служб .NET и OpenID они помечены идентификатором SyslogIdentifier. Кроме того, большинство компонентов ведёт файловый журнал.

Юниты, идентификаторы журналов и файловые журналы всех компонентов, включая службы MS-WSTEP, перечислены в таблице в начале главы Журналирование и мониторинг (see page 256).

Команды просмотра выполняются на сервере компонента в сеансе itc-svc. В примерах указан юнит ядра ЦС; для другого компонента подставьте его юнит.

```
# состояние службы и последние строки журнала
systemctl --user status minica.service
# весь журнал службы
journalctl --user -u minica.service
# новые записи в реальном времени, только текст сообщений; выход: Ctrl+C
journalctl --user -u minica.service -f -o cat
# ошибки и более важные записи с момента последней загрузки узла
journalctl --user -u minica.service -b -p err
# последние 100 записей по идентификатору, например адаптера SCEP
journalctl --user -t itc_scep -n 100 -o cat
```

Состояние службы показывает строка `Active:` в выводе `systemctl --user status`:

Значение	Что означает

`active (running)`	Служба работает
`activating (auto-restart)`	Процесс завершился, и systemd перезапустит его через 10 секунд (параметр юнита RestartSec=10). Если состояние повторяется, причину ищите в журнале службы
`inactive (dead)`	Служба остановлена
`failed`	Служба завершилась с ошибкой и не перезапускается

Ключ `-o cat` выводит только текст сообщений, без меток времени и имени узла. Ключ `-p err` оставляет записи с приоритетом err и выше.

Файловый журнал ядра ЦС

Прикладной журнал ЦС дополнительно пишется в файл /opt/itc/minica/logs/minica.log. Уровень детализации задаёт параметр `log.level` в minica.yml: flood, trace, debug, info, notice, warn, error, crit, fatal; по умолчанию info. Для разбора инцидента временно поднимите уровень до debug или trace и перезапустите службу, а после разбора верните info. Остальные параметры журналов описаны в главе Журналирование и мониторинг (see page 256).

9.2 Коды ответов API

Ответ ЦС на вызов API содержит числовой код результата: 0 означает успех, ненулевое значение указывает конкретную причину отказа. Код результата сопровождается соответствующим статусом HTTP. В Контрольной панели код операции виден в журнале «События», в столбце «Код ошибки».

В таблице столбец «Область» показывает подсистему, в которой возник отказ, а столбец «Уровень» показывает, насколько серьёзна ошибка: Notice, Warn или Error.

Код	Идентификатор	Значение	Когда возникает	Область	Уровень
5	`ERR_UNAUTHORIZED`	Неверный JWT	JWT отсутствует, недействителен, просрочен, отозван или не подходит для запрошенного ресурса	Аутентификация	Warn

10	`ERR_BAD_SIGN`	Неверная подпись	Криптографическая подпись запроса не прошла проверку	Аутентификация	Warn
15	`ERR_NO_PRIVILEGES`	Нет привилегий для шаблона	В JWT нет разрешения на использование выбранного шаблона ЦС	Авторизация	Warn
20	`ERR_BAD_CSR`	Неверный CSR	Запрос на выпуск сертификата повреждён или содержит недопустимые данные	PKI / CSR	Warn
25	`ERR_BAD_SUBJECT`	Subject не соответствует политике	Поле Subject запроса не удовлетворяет политике шаблона	PKI / Политика	Warn
30	`ERR_CERT_NOT_FOUND`	Сертификат не найден	Сертификат отсутствует в базе ЦС	PKI / Сертификаты	Warn
40	`ERR_CERT_EXPIRED`	Срок действия сертификата истёк	Операция невозможна: срок действия сертификата уже истёк	PKI / Сертификаты	Warn
50	`ERR_CERT_REVOKED`	Сертификат уже отозван	Повторная попытка отозвать сертификат	PKI / Сертификаты	Warn
60	`ERR_CERT_NOT_REVOKED`	Сертификат не отозван	Восстановление невозможно: сертификат не находится в отозванном состоянии	PKI / Сертификаты	Warn
70	`ERR_CRL_NOT_FOUND`	CRL не найден	Полный список отозванных сертификатов отсутствует	PKI / CRL	Warn
80	`ERR_DCRL_NOT_FOUND`	Delta CRL не найден	Разностный список отозванных сертификатов отсутствует	PKI / CRL	Warn
90	`ERR_CSR_NOT_FOUND`	CSR не найден	Запрос на выпуск сертификата отсутствует в базе ЦС	PKI / CSR	Warn

100	`ERR_BAD_REQ_TIME`	Неверное время запроса	Время запроса выходит за допустимый интервал (параметр `max-dt-sec`)	Валидация запроса	Warn
110	`ERR_OPENSSL`	Ошибка OpenSSL	Команда или криптографическая операция OpenSSL завершилась ошибкой	Криптография / OpenSSL	Error
120	`ERR_OPENSSL_TO`	Тайм-аут OpenSSL	Превышено время ожидания операции OpenSSL	Криптография / OpenSSL	Error
130	`ERR_INTERNAL`	Внутренняя ошибка	Внутренняя операция сервиса, например работа с файлами, завершилась ошибкой	Система	Error
140	`ERR_UNKNOWN_AKI`	Неизвестный AKI	Не найден ЦС, чей SKI соответствует указанному AKI	PKI / ЦС	Warn
150	`ERR_TEMPL_NOT_FOUND`	Шаблон не найден	Запрошенный шаблон сертификата отсутствует	Шаблоны	Warn
160	`ERR_TEMPL_DELETED`	Шаблон уже удалён	Повторная попытка удалить шаблон	Шаблоны	Warn
170	`ERR_BAD_KEY_TYPE`	Недопустимый тип ключа	Тип открытого ключа CSR не разрешён выбранным шаблоном	PKI / CSR	Warn
180	`ERR_SHORT_KEY_SIZE`	Недостаточная длина ключа	Длина открытого ключа CSR меньше установленного шаблоном минимума	PKI / CSR	Warn
190	`ERR_DEMO_VERSION_LIMIT`	Ограничение бесплатной редакции	Достигнут лимит количества сертификатов для бесплатной редакции (см. «Бесплатная и платная редакции продукта»)	Лицензирование	Notice

300	`ERR_JWT_NOT_FOUND`	JWT не найден	JWT отсутствует в базе данных	JWT	Warn
310	`ERR_JWT_REVOKED`	JWT уже отозван	Повторная попытка отозвать JWT	JWT	Warn
320	`ERR_JWT_NOT_REVOKED`	JWT не отозван	Восстановление невозможно: JWT не находится в отозванном состоянии	JWT	Warn
400	`ERR_BAD_REQUEST`	Неверный запрос	Переданы некорректные, неполные или неподдерживаемые параметры запроса	Валидация запроса	Warn
404	`ERR_NOT_FOUND`	Ресурс не найден	Запрошенный ресурс или маршрут API отсутствует	HTTP API	Warn
500	`ERR_DATABASE`	Ошибка базы данных	Ошибка обращения к СУБД или выполнения операции с данными	База данных	Error
520	`ERR_UNKNOWN`	Неизвестная ошибка	Ошибка не сопоставлена с более точным кодом	Система	Error
540	`ERR_BAD_PEM`	Некорректный PEM	Не удалось разобрать данные в формате PEM	PKI / PEM	Warn
580	`ERR_BAD_PEM_TYPE`	Некорректный тип PEM	Тип PEM-блока не соответствует ожидаемому	PKI / PEM	Warn
600	`ERR_REQUSTER_NOT_FOUND`	Инициатор запроса не найден	В запросе отсутствует обязательный заголовок инициатора	HTTP API	Warn
700	`ERR_TEMPL_NODB_APPR`	Несовместимые параметры шаблона	В шаблоне одновременно включены выпуск без сохранения в БД (nodb) и выпуск с одобрением (approval)	Шаблоны	Warn

800	`ERR_CRYPTOPTO`	Ошибка КриптоПро CSP	Операция криптопровайдера КриптоПро CSP завершилась ошибкой	Криптография / КриптоПро CSP	Error
900	`ERR_RATE_LIMIT`	Слишком много запросов	В бесплатной редакции превышен лимит запросов на выпуск сертификатов: 10 в минуту	Ограничение запросов	Warn

Помимо кода результата, причину отказа подсказывают сообщения в журнале службы. Категория сообщения указывает узел отказа: Конфигурация, СУБД, Криптография, HTTP клиент, Клиент (JWT), PKI операции, Валидация.

Сообщение в журнале	**Что означает**
`bad request time`	Рассинхронизация времени клиента и ЦС
`bad signature`	Подпись токена не прошла проверку
`JWT expired`, `JWT already revoked`	Проблема с токеном: истёк срок действия или токен отозван
`cert already revoked`	Повторный отзыв сертификата
`certificate not found`	Сертификат отсутствует
`can't access to database`, `database error`	База данных недоступна

9.3 Типовые проблемы и их устранение

Для каждой неисправности ниже приведены признаки, причина, что проверить и как исправить. Команды проверки приведены рядом с таблицей.

Если в столбце «Как исправить» указан перезапуск службы, учтите: на время перезапуска компонент не принимает запросы.

9.3.1 Служба ЦС не запускается

Что видно	Причина	Что проверить	Как исправить
`systemctl --user` выводит «Failed to connect to bus»	В сеансе не заданы XDG_RUNTIME_DIR и DBUS_SESSION_BUS_ADDRESS: команда выполнена без входа в сеанс itc-svc, например через `sudo -u itc-svc`	Значения переменных в сеансе (шаг 2 ниже)	Войдите в полноценный сеанс: `su - itc-svc`. Если переменных нет в ~/.bashrc, задайте их по шагу 4 раздела Общая подготовка сервера (see page 69)
Служба не работает после перезагрузки узла или останавливается после выхода из системы	Не включён режим linger: без него пользовательские юниты не запускаются без входа учётной записи в систему и останавливаются, когда завершается её последний сеанс	`loginctl show-user itc-svc` (шаг 2 ниже)	`sudo loginctl enable-linger itc-svc`
Служба в состоянии `activating (auto-restart)`, в журнале «can't parse database URL» или «bad CA configuration»	Ошибка в файле конфигурации minica.yml	Путь к файлу, его наличие и ключевые параметры (шаг 3 ниже)	Исправьте файл и перезапустите службу
Служба не запускается	Каталог ЦС или ключевые файлы недоступны службе: принадлежат другой учётной записи или не читаются учётной записью itc-svc	Владелец и права файлов (шаг 4 ниже)	Верните каталогу ЦС и ключевым файлам владельца itc-svc. Отдельно, как требование безопасности: закрытым ключам и файлам с паролями нужны права 0600 (см. Ограничение прав доступа к файлам (see page 178))
В журнале «create Postgres DAO», «database error», «can't access to database»	Нет доступа к базе данных ЦС	См. раздел «Нет доступа к базе данных (код 500, can't access to database)»	См. раздел «Нет доступа к базе данных (код 500, can't access to database)»

Шаг 1. Посмотрите состояние службы и последние записи журналов. Первые строки вывода `status` показывают состояние службы, а журнал содержит сообщение, с которым процесс завершился.

```
systemctl --user status minica.service
journalctl --user -u minica.service -n 50 -o cat
tail -n 50 /opt/itc/minica/logs/minica.log
```

Шаг 2. Проверьте окружение сеанса и режим linger. Первая команда должна вывести `/run/user/<UID> unix:path=/run/user/<UID>/bus`, где `<UID>` означает числовой идентификатор itc-svc; вторая должна вывести `Linger=yes`.

```
echo $XDG_RUNTIME_DIR $DBUS_SESSION_BUS_ADDRESS
loginctl show-user itc-svc | grep Linger
```

Если выводится `Linger=no`, администратор с правом sudo включает режим linger:

```
sudo loginctl enable-linger itc-svc
```

Шаг 3. Проверьте файл конфигурации. Путь к нему задаёт переменная MINICA_CONF в файле окружения службы /opt/itc/minica/minica.env; ожидается /opt/itc/minica/conf/minica.yml. Файл должен существовать и принадлежать itc-svc. Третья команда выводит номера строк ключевых параметров, не показывая пароль БД.

```
grep MINICA_CONF /opt/itc/minica/minica.env
ls -l /opt/itc/minica/conf/minica.yml
grep -n -E '^(max-dt-sec|database):|allow-without-tls' /opt/itc/minica/conf/minica.yml
```

Шаг 4. Проверьте владельца и права. Служба должна иметь доступ к этим файлам на чтение. По требованиям безопасности для каждого файла ожидается `600 itc-svc:itc-svc`, а каталог должен принадлежать itc-svc.

```
ls -ld /opt/itc/minica
stat -c '%a %U:%G %n' /opt/itc/minica/ca/ca.key /opt/itc/minica/conf/*.key \
  /opt/itc/minica/conf/mclient.jwt /opt/itc/minica/conf/minica.yml
```

Шаг 5. После исправления перезапустите службу и проверьте ЦС командой `mclient ping` (см. «Проверка доступности ЦС (mclient)»). На время перезапуска ЦС не принимает запросы.

```
systemctl --user restart minica.service
```

9.3.2 Ошибка проверки времени (код 100, bad request time)

ЦС отклоняет запрос, если время клиента и сервера расходится больше, чем на окно `max-dt-sec`. По умолчанию окно равно 600 с (±10 минут).

Что видно	Причина	Что проверить	Как исправить
Код 100 `ERR_BAD_REQ_TIME`, в журнале ЦС «bad request time»	Часы клиента и сервера ЦС расходятся больше, чем на `max-dt-sec`	Синхронизацию времени на узлах ЦС, сервере СУБД и узле клиента: `timedatectl status`	Настройте синхронизацию времени по NTP (systemd-timesyncd, chrony или ntpd) по документации ОС; см. также «Синхронизация времени и защита от повторов» в главе Обеспечение безопасной среды (see page 167)

Проверьте синхронизацию на каждом узле:

```
timedatectl status
```

В выводе должны быть строки `System clock synchronized: yes` и `NTP service: active`.

Окно задаётся в файле /opt/itc/minica/conf/minica.yml:

```
max-dt-sec: 600   # допустимое рассогласование времени, с
```

Изменение `max-dt-sec` вступает в силу после перезапуска службы. На время перезапуска ЦС не принимает запросы.

```
systemctl --user restart minica.service
```

> ⚠ **Не расширяйте окно без необходимости**

> Увеличение `max-dt-sec` расширяет окно приёма запросов и ослабляет защиту от повторного воспроизведения. Устраняйте причину (расхождение часов), а не наращивайте окно.

9.3.3 Нет доступа к базе данных (код 500, can't access to database)

Ядро ЦС подключается к PostgreSQL по строке подключения `database.url` в minica.yml; поддерживаются версии PostgreSQL 11–18. Требование TLS к этому подключению задаёт параметр `database.allow-without-tls`:

- `true` (значение установщика по умолчанию): подключение допускается и без TLS;
- `false` (безопасное состояние): если сервер БД не поддерживает TLS, служба завершает работу.

Что видно	Причина	Что проверить	Как исправить
Код 500 `ERR_DATABASE`; в журнале «can't access to database», «database error», «create Postgres DAO» или «can't get cert stat»	Сервер СУБД недоступен или неверна строка подключения: сервер, порт 5432, имя БД, учётные данные	Подключение к БД с сервера ЦС (команда ниже); значение `database.url`	Исправьте `database.url` или откройте доступ к СУБД: listen_addresses, pg_hba.conf, межсетевой экран (шаг 4 раздела Развертывание базы данных ЦС (see page 71)). Затем перезапустите службу
Служба завершает работу сразу после запуска, в minica.yml `allow-without-tls: false`	На сервере БД не включён TLS	Настройки TLS сервера PostgreSQL	Включите TLS на сервере PostgreSQL (см. «Защита подключений к СУБД и каталогу LDAP» в главе Обеспечение безопасной среды (see page 167))

Чтобы служба проверяла сертификат сервера БД, добавьте в `database.url` параметр `sslmode=verify-full` и используйте корректный корневой сертификат. Без этого параметра служба доверяет любому TLS-сертификату сервера БД.

Проверьте подключение к БД с сервера ЦС. Команде нужен клиент PostgreSQL (psql). Ядру ЦС этот клиент не нужен, поэтому на сервере ЦС его может не быть: в этом случае установите пакет клиента PostgreSQL или проверьте хотя бы доступность порта 5432 (команда ниже).

Заполнители замените значениями из `database.url` (строка вида `postgres://<пользователь>:<пароль>@<сервер БД>:<порт>/<имя БД>`). По умолчанию пользователь и БД называются cwica. Ожидается вывод `1`.

```
PGPASSWORD='<пароль>' psql -U <пользователь> -h <сервер БД> -p 5432 -d <имя БД> -tAc
"SELECT 1"
```

Команда сохраняет пароль в истории оболочки. После проверки удалите эту строку из истории (`history -d <номер>`).

Если клиента psql нет, проверьте с сервера ЦС доступность порта сервера БД. Вывод `порт доступен` означает, что сервер БД принимает TCP-подключения. Сообщение об ошибке (`Connection refused`, `No route to host`) или отсутствие вывода в течение 5 секунд означает, что порт недоступен.

```
timeout 5 bash -c '</dev/tcp/<сервер БД>/5432' && echo "порт доступен"
```

Канал к СУБД и подключения к ЦС

Параметр `allow-without-tls` управляет только каналом «служба ЦС → БД». Приём защищённых подключений к самому ЦС настраивается в секции `tls` файла minica.yml: `enable: true`, `verify: true` для взаимной аутентификации, порт 9443.

9.3.4 Ошибки TLS, JWT и подписи (коды 5 и 10)

Признаки: отказ `mclient ping`, ошибки установления соединения с ЦС, коды 5 и 10 в ответах ЦС.

Что видно	Причина	Что проверить	Как исправить
Клиент не устанавливает TLS-соединение с ЦС	Истёк срок TLS-сертификата ЦС, имя в адресе не совпадает с именем в сертификате или клиент не доверяет ЦС	Срок действия, субъект и издатель сертификата /opt/itc/minica/conf/tls.crt; адрес `server-url` в mclient.yml (команды ниже)	Обращайтесь к ЦС по имени из сертификата; при истёкшем сроке выпустите TLS-сертификат заново (см. «Включение TLS» в разделе «Развертывание ядра ЦС (minica)»)

При включённой взаимной аутентификации (`tls.verify: true`) ЦС отклоняет соединение	Клиентский сертификат выпущен ЦС, которому сервер не доверяет, или отозван	Параметры `tls.ca-cert` и `tls.ca-certs` в minica.yml; статус клиентского сертификата по CRL или OCSP	Выпустите клиенту сертификат доверенным ЦС или добавьте сертификат его ЦС в `tls.ca-certs` ; перезапустите службу
Код 5 `ERR_UNAUTHORIZED` или 10 `ERR_BAD_SIGN`	JWT клиента отсутствует, недействителен, просрочен или отозван, либо подпись запроса не прошла проверку	Файлы токена и ключа подписи клиента: для mclient параметры `jwt-file` и `key-file` в mclient.yml (conf/mclient.jwt, conf/mclient.key); для Контрольной панели, сервиса публикации и SCEP параметры `Auth.Jwt` и `Auth.Sign` в блоке MiniCa	Укажите верные файлы; при истёкшем или отозванном токене выпустите новый JWT: для mclient см. «Ключи mclient, запуск и проверка», для сервисов см. «Выпуск JWT для сервиса»

Проверьте TLS-сертификат ЦС и адрес, по которому к ЦС обращается mclient:

```
openssl x509 -in /opt/itc/minica/conf/tls.crt -noout -dates -subject -issuer
grep -n '^server-url' /opt/itc/minica/conf/mclient.yml
```

В выводе первой команды `notBefore` и `notAfter` показывают срок действия сертификата, `subject` содержит имя сервера ЦС, `issuer` указывает ЦС, выпустивший сертификат. Адрес `server-url` должен содержать то же имя сервера и порт 9443.

Дополнительно проверьте:

- для шифрования данных используется самостоятельная пара `ca.encryption` (ca/encr.key, ca/encr.crt), а не ключ подписи ЦС;
- путь к сертификату KRA указан как ca/kra.crt (параметр `ca.kra.cert`);
- после замены сертификатов или правки секции `tls` служба перезапущена командой `systemctl --user restart minica.service` ; на время перезапуска ЦС не принимает запросы.

9.3.5 OCSP-ответчик mOCSP

OCSP-ответчик mOCSP работает как служба и отвечает клиентам на запросы о статусе сертификатов: действителен, отозван или неизвестен. Ответчик принимает запросы напрямую,

без Nginx, по адресу `http://<сервер служб>:8888/ocsp`: порт задаёт параметр `listen-port`, путь задаёт параметр `url-path` в /opt/itc/ocsp/conf/mocsp.yml. Запросы принимаются по HTTP методами GET и POST. Проверьте ответчик командой openssl на любом узле с доступом к порту 8888:

```
openssl ocsp -issuer <сертификат выдающего ЦС> -cert <проверяемый сертификат> \
  -url "http://<сервер служб>:8888/ocsp" -CAfile <корневой сертификат ЦС>
```

Строка `Response verify OK` означает, что подпись ответа проверена. Следующая строка показывает статус сертификата: `good` (не отозван), `revoked` (отозван) или `unknown` (ЦС не знает сертификата с таким серийным номером).

Что видно	Причина	Что проверить	Как исправить
`Error connecting BIO` с `Connection refused`, `Connection timed out` или `No route to host`	`Connection refused` означает, что служба mocsp.service не запущена или не слушает порт 8888. Истечение времени ожидания или `No route to host` означает, что порт 8888 закрыт межсетевым экраном	Состояние службы и прослушиваемый порт (команды ниже); правила межсетевого экрана на сервере служб	Запустите службу; откройте порт 8888/tcp для клиентов OCSP
`Error querying OCSP responder … Code=404`	Неверный путь в адресе ответчика	Параметр `url-path` (по умолчанию `/ocsp`)	Исправьте адрес ответчика у клиента

Журнал и состояние ответчика проверяются на сервере служб в сеансе itc-svc. Файловый журнал: /opt/itc/ocsp/logs/mocsp.log.

```
systemctl --user status mocsp.service
ss -ltn | grep ':8888'
journalctl --user -u mocsp.service -n 100 -o cat
```

9.3.5.1 Получение статуса из CRL

В режиме crl-only БД не используется, поэтому проверка её доступности не входит в диагностику этого режима. В режиме db-first проверяйте и ошибки запросов к БД, и готовность резервного CRL.

Что видно	Причина	Что проверить	Как исправить
В OCSP-ответе указан tryLater	CRL не загружен, ещё не вступил в действие либо просрочен при `fail-on-expired: true`	`crl-path`, права чтения файла, ThisUpdate и NextUpdate CRL, системное время, `fail-on-expired`	Обеспечьте по пути `crl-path` действующий CRL, доступный службе на чтение. Признак исправности: CRL загружен, запрос возвращает ожидаемый статус сертификата
После изменения файла используется прежний CRL	Автоперезагрузка отключена либо новый файл не прошёл чтение, проверку подписи или разбор	Параметр `reload-enabled` и сообщения журнала «crl reload failed, old crl will be used» (метод poll) или «crl reload by inotify failed, old crl cache will be used» (метод inotify)	Включите `reload-enabled` и замените файл корректным CRL. Признак исправности: в журнале «crl reload completed» (poll) или «crl reload by inotify completed» (inotify)
На системе, отличной от Linux, файл не перезагружается	Выбран метод inotify	Параметры `crl.source.watch.method` и `poll-interval`	Установите `crl.source.watch.method: poll`. Изменение файла обнаруживается при очередной проверке
В режиме db-first CRL не используется для отсутствующей записи БД	БД вернула результат «unknown» без технической ошибки	Результат запроса к БД: переход на CRL выполняется только при ошибке запроса	Это поведение режима db-first. При технической ошибке БД в журнале появляется сообщение о переходе на CRL
Сервис использует только БД	Неизвестное или ошибочно написанное значение `status-source` преобразовано в db-only	Точное значение `status-source`: db-only, db-first или crl-only	Исправьте значение и перезапустите mocsp.service; на время перезапуска ответчик не принимает запросы. Признак исправности: используется выбранный администратором источник

9.3.6 Адаптер SCEP

Точная причина отклонения запроса SCEP записывается в журнал адаптера на сервере, где работает scepAdapter.service. Журнал просматривается в сеансе itc-svc; файловые журналы находятся в каталоге /opt/itc/itc/logs/itc_acm_scepAdapter_clearwayCa.

```
journalctl --user -u scepAdapter.service -n 100 -o cat
ls -lt /opt/itc/itc/logs/itc_acm_scepAdapter_clearwayCa
```

Что видно	Причина	Что проверить	Как исправить
certmonger: `CA_UNREACHABLE` с сообщением `PKCS7_verify:no content`	Адаптер отклонил запрос: одноразовый пароль уже использован или не передан	Журнал адаптера: «Challenge password уже был использован» или «For the initial issuance of a certificate, a ChallengePassword is required.»	Получите новый одноразовый пароль (он подходит для одного запроса и по умолчанию действует 60 минут), запишите его в файл пароля и выполните `getcert resubmit -i <имя задачи>`
certmonger: `NEED_CSR_GEN_PIN`, `stuck: yes`	Файл пароля пуст	Содержимое файла пароля, указанного в задаче certmonger	Запишите в файл новый пароль и выполните `getcert resubmit -i <имя задачи>`
Запрос одноразового пароля (`<адрес SCEP>/challenge-password`) возвращает 401	Не передан токен доступа OpenID	Заголовок `Authorization: Bearer <токен доступа>` в запросе	Получите токен и сразу запросите пароль: по умолчанию токен доступа действует 2 минуты
Запрос к адресу SCEP возвращает HTML-страницу Контрольной панели	Путь в адресе не совпадает с блоком location адаптера в Nginx (путь чувствителен к регистру), и Nginx передаёт запрос Контрольной панели	Блок /opt/itc/itc/nginx/scep.conf	Укажите путь точно как в блоке location, по умолчанию `/api/v1/scep`

Кто инициировал выпуск по SCEP

Для сертификатов, выпущенных по SCEP, в поле Requester журнала событий ЦС записывается субъект устройства. Кто получил одноразовый пароль, видно в файловом журнале адаптера: запись аудита ScepAudit о запросе `challenge-password` содержит имя пользователя (см. главу Журналирование и мониторинг (see page 256)).

9.3.7 MS-WSTEP

Адаптер MS-WSTEP выпускает сертификаты для компьютеров Windows и состоит из двух служб: служба политик регистрации CEP сообщает компьютеру доступные шаблоны, служба регистрации CES принимает запросы и передаёт их в ЦС. Обе службы читают описания шаблонов и служб регистрации из Active Directory. Юниты служб называются wstepCep.service (CEP) и wstepCesAdapter.service (CES); идентификаторы в журнале (itc_wstep_cep, itc_wstep_ces) и каталоги файловых журналов приведены в таблице в начале главы Журналирование и мониторинг (see page 256).

Типовая неисправность:

Что видно	Причина	Что проверить	Как исправить
В окне запроса сертификата Windows нет нужного шаблона, или certreq завершается ошибкой	Описание шаблона или службы регистрации не импортировано в Active Directory, либо службы CEP и CES не перечитали каталог после импорта	Импорт файлов template_tls_export.ldf и enrollment_services.ldf командой `ldifde -i` выполнен учётной записью с правами на раздел конфигурации AD; журналы служб	Импортируйте файлы, затем перезапустите wstepCep.service и wstepCesAdapter.service, чтобы службы перечитали каталог

```
journalctl --user -t itc_wstep_cep -n 100 -o cat
journalctl --user -t itc_wstep_ces -n 100 -o cat
# перезапуск служб; на время перезапуска выпуск по MS-WSTEP недоступен
systemctl --user restart wstepCep.service wstepCesAdapter.service
```

9.3.8 Контрольная панель

Серверная часть Контрольной панели (controlPanel.service) проверяется на её сервере в сеансе itc-svc.

Что видно	Причина	Что проверить	Как исправить

Уведомление «Нет доступа к данному ресурсу»	Сервер отклонил запрос: у пользователя нет нужного права	Права пользователя в разделе «Профиль» (блок «Права») и карту групп `ldapGroupRolesMap` в /opt/itc/openid/config.yaml	Включите пользователя в группу каталога с нужными правами (права пунктов меню см. «Права и пункты меню Контрольной панели» в главе Описание ролевой модели (see page 47)) или дополните карту и перезапустите openid.service на сервере OpenID (см. Назначение ролей через группы AD/LDAP (see page 196)); пользователь должен выйти и войти снова
Нет пунктов «Задачи → Резервное копирование» и «Настройки → Ротация ключей» даже у администратора	В карте групп нет прав, добавленных в 26.3.0: `rleMiniCABackup-r`, `rleMiniCABackup-w`, `rleMiniCABackupRestore-w`, `rleMiniCARotation-w`	Блок «Права» в профиле администратора	Дополните карту по «Добавление новых прав 26.3.0 в карту групп» (глава Описание ролевой модели (see page 47))
Нет группы меню «Задачи»	У пользователя нет права `rleMiniCACRL-w`	Блок «Права» в профиле пользователя	Назначьте право `rleMiniCACRL-w`
Нет пункта «Задачи → Типовые», по прямому адресу выводится «Функционал недоступен»	Функция типовых задач выключена параметром `Features.TypicalTask`	Параметр в /opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json (команда ниже); у пользователя должны быть права `rleMiniCACertReq-w` и `rleMiniCACRL-w` (без него скрыта вся группа «Задачи»)	Задайте `"TypicalTask": true` в секции Features и перезапустите controlPanel.service

Уведомление «Не удалось подключиться к серверу авторизации»	Сервис аутентификации OpenID недоступен	Состояние openid.service на сервере OpenID; доступность адреса OpenID (`https://<fqdn OpenID>`) с компьютера пользователя и с сервера панели	Запустите службу или устраните сетевую недоступность
На странице «Настройки → Конфигурация» у карточки компонента красный индикатор, версия не выводится	Панель не получила ответ компонента: служба остановлена, неверен адрес или закрыт порт	Для ЦС параметр `Url` в блоке MiniCa (файл itc_api_miniCA_controlPanel_config.json, а для ЦС, подключённых через интерфейс, файл itc_api_miniCA_config.runtime.json) и порт 9443; для сервиса публикации параметр `Services.PublicationCrlUrl` и порт 9407; журнал controlPanel.service	Запустите службу компонента, исправьте адрес и перезапустите controlPanel.service
На странице «Статистика ЦС» сообщение «Статистика ещё не собрана»	Расчёт статистики по расписанию ещё не выполнялся	Расписание, указанное под сообщением	Дождитесь ближайшего расчёта; текущие показатели выводит `mclient stat`
На странице «Статистика ЦС» сообщение «Ошибка загрузки данных»	Контрольной панели не удалось получить показатели	Журнал controlPanel.service	Устраните причину по журналу и обновите страницу

```
systemctl --user status controlPanel.service
journalctl --user -u controlPanel.service -n 100 -o cat
# ожидается строка "TypicalTask": true
grep -n -A2 '"Features"' /opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json
# после правки конфигурации; на время перезапуска панель недоступна
systemctl --user restart controlPanel.service
```

9.3.9 Резервное копирование и восстановление

Задания резервного копирования выполняет сервис резервного копирования, а Контрольная панель передаёт ему запросы по адресу из `Services.BackupUrl`. Ошибки шагов выводятся в карточке задания (блок «Текущая операция», статус «Неуспех» с текстом ошибки) и записываются в журнал сервиса (см. таблицу в начале главы Журналирование и мониторинг (see

page 256)). Требования к серверу сервиса и узлам ЦС приведены в разделе «Резервное копирование», подразделы «Подготовка сервера сервиса резервного копирования» и «Подготовка узлов ЦС и сети».

Что видно	Причина	Что проверить	Как исправить
Раздел «Задачи → Резервное копирование» не загружает задания, выводится ошибка	Не задан параметр `Services.BackupUrl` серверной части Контрольной панели или сервис резервного копирования не отвечает по этому адресу	Параметр в /opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json; при пустом адресе в журнале controlPanel.service есть ошибка `Invalid URI: The URI is empty`. Ответ сервиса на сервере панели: `curl http://<адрес сервиса>/api/health`	Укажите адрес сервиса и перезапустите controlPanel.service; запустите службу сервиса резервного копирования или откройте его порт для сервера панели
Служба сервиса резервного копирования не запускается, в журнале сообщение, что инструмент `pg_dump` или `pg_restore` не найден, с советом установить пакет postgresql-client	На сервере сервиса нет утилит клиента PostgreSQL или неверен путь к ним	Команды `pg_dump --version` и `pg_restore --version`; параметры `Tools.PgDumpPath` и `Tools.PgRestorePath`	Установите пакет клиента PostgreSQL той же основной версии, что сервер БД ЦС, или укажите полный путь к утилитам в секции `Tools`
Служба сервиса резервного копирования не запускается, в журнале ошибка прав на файл конфигурации	Права на itc_api_miniCA_backup_config.json шире «чтение и запись только владельцу»	`stat -c '%a %U' /opt/itc/itc/config/itc_api_miniCA_backup_config.json`	Задайте права 600, владельцем сделайте учётную запись, от имени которой работает служба, затем запустите её
Шаг завершился ошибкой «Не удалось подключиться к узлу … по SSH»	Узел недоступен с сервера сервиса по SSH, неверные адрес, порт, имя или пароль, вход по паролю запрещён	Доступность порта SSH узла с сервера сервиса; параметр `PasswordAuthentication` в sshd узла; учётные данные в задании	Откройте доступ, исправьте учётные данные в задании или разрешите вход по паролю

Шаг завершился ошибкой «Ошибка команды на узле … (код N)»	sudo отказал: у пользователя SSH нет права sudo, sudo ждёт другой пароль, в sudoers включён `requiretty`, в sshd запрещён проброс TCP, на сервере сервиса не задан сертификат канала askpass или на узле нет нужных утилит	Текст ошибки после кода; на узле `sudo -l` от имени пользователя SSH, `Defaults requiretty` в sudoers, `AllowTcpForwarding` в sshd, наличие `bash`, `curl`, `find` и `base64`; параметр `SshAskPass.CertificatePath` сервиса	Приведите узел и сервис в соответствие с требованиями раздела «Резервное копирование»
Шаг завершился ошибкой «На узле … не найден путь … из состава файлов конфигурации»	Путь из поля «Файлы конфигурации» не существует на узле или указан через символьную ссылку	`sudo ls -ld <путь>` на узле	Укажите в задании реальный путь к файлу или каталогу
Шаг «Копирование базы данных» или «Восстановление базы данных» завершился ошибкой «pg_dump завершился с кодом …» или «pg_restore завершился с кодом …»	Нет доступа к БД ЦС с сервера сервиса, неверный пароль, версия утилит не совпадает с версией сервера БД	Текст ошибки; `pg_hba.conf` и межсетевой экран сервера БД; `pg_dump --version` на сервере сервиса и версия сервера PostgreSQL	Разрешите подключение с сервера сервиса, исправьте пароль в задании, установите утилиты нужной версии
Операция не начинается: «Свободный объём в папке хранения … ниже порога …»	В каталоге копий свободно меньше `MinFreeSpaceBytes`	`df -h <StoragePath>` на сервере сервиса	Освободите место: перенесите старые копии на отдельный носитель, уменьшите `RetentionDays` или расширьте раздел
Копия отмечена «Непригодна», восстановление запрещено с ошибкой «Резервная копия … не прошла проверку пригодности»	Архив повреждён или изменён, нет файла архива, для зашифрованной копии задан другой ключ	Причина в столбце «Пригодность»; наличие архива в каталоге копий; параметр `Encryption.KeyFile`	Восстановите ЦС из другой пригодной копии; для зашифрованной копии верните ключ, которым она создана

Копирование не завершено: «Файл ключа шифрования не задан или отсутствует» или «Ключ шифрования … должен иметь длину 32 байта»	Шифрование включено, а файл ключа не найден или ключ неверной длины	Параметр `Encryption.KeyFile` и файл ключа	Укажите путь к файлу с ключом 32 байта в base64 или hex и перезапустите службу сервиса
Запуск отклонён: «Для ЦС … уже выполняется операция» или сообщение, что уже выполняется восстановление другого ЦС	Для этого ЦС уже идёт копирование или восстановление, либо сервис восстанавливает другой ЦС: восстановления выполняются по одному	Блок «Текущая операция» в карточках заданий	Дождитесь завершения операции. Запуск по расписанию во время другой операции пропускается, дождитесь следующего
Операция прервана: «превышена предельная длительность (N мин)»	Обработка узла заняла больше `OperationTimeoutMinutes`	Размер каталогов в поле «Файлы конфигурации», скорость сети до узла	Увеличьте `OperationTimeoutMinutes` в конфигурации сервиса и перезапустите службу
Восстановление завершилось ошибкой «Сервис ЦС на узле … не остановился» или «… не запустился после восстановления», узел в статусе «Возвращено к сохранённому состоянию»	Неверно указаны юнит, режим systemctl --user или пользователь сервиса; у технологической учётной записи не работает пользовательская сессия systemd; служба ЦС не запускается с восстановленной конфигурацией	Поля задания «systemd-юнит», «systemctl --user», «Пользователь сервиса»; `loginctl show-user itc-svc` на узле; журнал minica.service	Исправьте задание или включите linger; если служба не запускается с данными копии, разберите причину по журналу ЦС (см. «Служба ЦС не запускается»)

```
# на сервере Контрольной панели: адрес сервиса резервного копирования
grep -n -A3 '"Services"' /opt/itc/itc/config/itc_api_miniCA_controlPanel_config.json
# на сервере сервиса резервного копирования: утилиты, права конфигурации, место под копии
pg_dump --version; pg_restore --version
stat -c '%a %U' /opt/itc/itc/config/itc_api_miniCA_backup_config.json
df -h <StoragePath>
```

9.3.10 Переполнение диска под журналы (watchdog журнала безопасности)

Журнал безопасности ЦС (`log.secLog`) может следить за свободным местом на разделе с журналом (механизм watchdog). По умолчанию он выключен (`watchdogEnabled: false`). Когда механизм включён, служба периодически проверяет занятость раздела и при достижении порогов отправляет уведомления в интерфейс или по электронной почте, если эти каналы включены. Параметры описаны в разделе «Отказоустойчивый журнал событий безопасности (secLog)» главы Журналирование и мониторинг (see page 256).

```
# /opt/itc/minica/conf/minica.yml
log:
  secLog:
    path: "/opt/itc/minica/logs/security.log"
    watchdogEnabled: true
    watchdogProbeInterval: 20   # период проверки, с
    watchdogWarnPercent: 85     # предупреждение при 85% занятости диска
    watchdogCritPercent: 95     # критический порог при 95%
```

Что видно	Причина	Что проверить	Как исправить
Уведомление watchdog: занятость диска достигла 85% (предупреждение) или 95% (критический порог)	На разделе с журналами заканчивается место	Занятость раздела и объём каталога журналов (команды ниже)	Освободите место: архивируйте и вынесите старые журналы, настройте ротацию (секция `log.rotate`)

```
df -h /opt/itc
du -sh /opt/itc/minica/logs
```

В выводе `df` столбец `Use%` показывает занятость раздела; сравните её с порогами 85% и 95%. Команда `du` показывает, сколько места занимают журналы ЦС.

При достижении критического порога не отключайте механизм, а устраните нехватку места: иначе возможна потеря событий безопасности.

> **Буфер событий безопасности**
>
> Отказоустойчивый журнал безопасности буферизует события в ОЗУ: по умолчанию до 50 000 событий и не более 50 МБ. При длительном переполнении диска и исчерпании буфера часть событий может не записаться на диск. Реагируйте на предупреждение (85%) до достижения критического порога (95%).

9.4 Быстрый чек-лист диагностики

1. `./mclient ping`: ЦС отвечает, связь, токен, подпись и время в норме?
2. `systemctl --user status <юнит>`: служба в состоянии active (running), linger включён?
3. `journalctl --user -u <юнит> -p err`: какие ошибки и в какой категории?
4. Код ответа API: найдите его в таблице «Коды ответов API» и определите узел отказа.
5. `timedatectl status`: время синхронизировано на всех узлах ЦС и на сервере СУБД?
6. База данных: доступна с сервера ЦС, верны `database.url` и `allow-without-tls`?
7. `df -h /opt/itc`: достаточно ли места под журналы?
8. «Настройки → Конфигурация» в Контрольной панели: у карточек компонентов зелёные индикаторы?

10 Приложения

Приложения содержат краткие справочники для эксплуатации: основные параметры ядра ЦС, сетевые порты для правил межсетевого экрана, сводка по ролям и правам и команды консольного клиента mclient. Значения приведены для установки по умолчанию. Каждая тема подробно описана в разделах, на которые ссылается приложение.

10.1 Приложение А. Ключевые параметры minica.yml

Сводка параметров файла конфигурации ядра ЦС `/opt/itc/minica/conf/minica.yml`, которые чаще всего нужны при эксплуатации. Полное описание, порядок изменения и примеры приведены в разделе Настройка ядра ЦС (minica.yml) (see page 126), а параметры журналирования описаны в главе Журналирование и мониторинг (see page 256).

Пути в таблицах указаны относительно каталога ЦС `/opt/itc/minica`. В файле они записаны полностью: например, вместо `conf/tls.key` в файле стоит `/opt/itc/minica/conf/tls.key`. Изменения вступают в силу после перезапуска службы `minica.service` (см. Управление службой (see page 138)).

10.1.1 Сеть, TLS и контроль времени

Подробнее см. разделы «Сеть и HTTP/HTTPS-сервер», «TLS и взаимная аутентификация (mTLS)» и «Контроль рассогласования времени (max-dt-sec)».

Параметр	Назначение	Значение по умолчанию
`listen-ip`	IP-адрес, на котором ЦС принимает соединения. Пустая строка означает все интерфейсы	`""`
`http-port`	Порт HTTP, используется при выключенном TLS	9080
`https-port`	Порт HTTPS, используется при включённом TLS	9443
`tls.enable`	Включение TLS: ЦС принимает подключения на порту `https-port`	`false` в шаблоне; `true` после установки
`tls.verify`	Взаимная аутентификация (mTLS): ЦС требует от клиента доверенный сертификат	`false`

`tls.key` `tls.cert`	Закрытый ключ и сертификат сервера ЦС	`conf/tls.key` `conf/tls.crt`
`timeout-ms`	Тайм-аут обработки запроса, мс	60000
`max-dt-sec`	Допустимое рассогласование времени при проверке подписи запроса и токена доступа JWT, с	600 (окно ±10 минут)

10.1.2 Ключ ЦС, сроки действия и расширения

Подробнее см. разделы «Ключ и сертификат ЦС», «Шаблоны, поля CDP/AIA и сроки действия», «Шифрование данных ЦС (ca.encryption)» и «Восстановление ключей субъектов (KRA)».

Параметр	Назначение	Значение по умолчанию
`ca.key` `ca.cert`	Закрытый ключ и сертификат ЦС	`ca/ca.key` `ca/ca.crt`
`ca.old-keys`	Каталог прежних ключей и сертификатов ЦС, используется после ротации ключа ЦС	`ca/old-keys`
`ca.new-keys`	Каталог нового ключа ЦС: ключ, созданный при «Перевыпуске», хранится здесь до установки сертификата ЦС	`ca/new-keys`
`chain-file`	Файл цепочки сертификатов ЦС (PEM)	`ca/chain.pem`
`openssl.cnf`	Базовый конфигурационный файл OpenSSL	`conf/openssl.cnf`
`ca.start-sn` `ca.start-crl`	Стартовые серийные номера сертификатов и списков отзыва (CRL). `ca.start-sn` в бесплатной редакции не действует	`10000000000000000000`
`ca.days.cert`	Срок действия выпускаемых сертификатов, дней	1460 (≈4 года)
`ca.days.crl`	Срок действия CRL, дней	365

`ca.overlap`	Смещение начала срока действия сертификата в прошлое, мин	10
`ca.cdp`	Адреса точки распространения CRL (CDP) в сертификатах	`["http://<CDP_SERVER_NAME>/cdp/<CN>.crl"]`
`ca.aia`	Элементы расширения AIA: адреса в сертификате, по которым клиент получает сертификат издателя и находит OCSP-ответчик. Чтобы в сертификаты записывался адрес OCSP-ответчика, добавьте элемент `OCSP;URI:` (см. «Адрес OCSP-ответчика в AIA»)	`["caIssuers;URI:http://<CDP_SERVER_NAME>/aia/<CN>.crt"]`
`ca.freshest-crl`	Адреса разностного списка отзыва Delta CRL (суффикс `+` в имени файла)	Выдающий ЦС: `["http://<CDP_SERVER_NAME>/cdp/<CN>+.crl"]` Корневой ЦС: не задан
`ca.encryption.key` `ca.encryption.cert`	Отдельная пара ключей для шифрования данных ЦС; не совпадает с ключом подписи ЦС	`ca/encr.key` `ca/encr.crt`
`ca.kra.cert`	Сертификат агента восстановления ключей (KRA)	`ca/kra.crt`

`<CDP_SERVER_NAME>` обозначает FQDN сервера точек распространения, который указывается при установке ЦС. `<CN>` обозначает значение CN из субъекта сертификата ЦС, в котором пробелы заменены на «_».

10.1.3 JWT и подпись запросов

JWT служит токеном доступа, который выпускает ЦС. В токене указана сервисная роль клиента, и ЦС выполняет только те функции API, которые разрешены этой роли. Кроме того, клиент подписывает каждый запрос своим ключом.

В таблице приведены роли и параметры, которые отключают проверки. Ключи подписи и проверки JWT и запросов (`jwt.key` , `jwt.keys` , `jwt.cache-minutes` , `master-key`) и состав каждой роли описаны в разделе «Сервисные роли JWT (ролевая модель)».

Параметр	Назначение	Значение по умолчанию
`jwt.roles`	Сервисные роли JWT: имя роли и перечень разрешённых функций API	7 ролей: `admin`, `causer`, `crluser`, `archuser`, `exporter`, `templadmin`, `jwtadmin`
`jwt.no-check sign-off`	Не проверять JWT клиента; отключить проверку подписи запросов. В промышленной эксплуатации оба параметра равны `false`	`false` `false`

10.1.4 База данных

Подробнее см. подраздел «База данных» раздела Настройка ядра ЦС (minica.yml) (see page 126), где описаны также тип СУБД и время хранения шаблонов в кеше, и раздел «Защита подключений к СУБД и каталогу LDAP» главы Обеспечение безопасной среды (see page 167).

Параметр	Назначение	Значение по умолчанию
`database.url`	Строка подключения к БД ЦС: `postgres://<пользователь>:<пароль>@<сервер БД>:<порт>/<имя БД>`	Заполняет установщик; по умолчанию пользователь и БД `cwica`, порт 5432
`database.allow-without-tls`	Разрешить подключение ЦС к СУБД без TLS. Для безопасного состояния задаётся `false`	`true`

Поддерживаются PostgreSQL 11–18. Параметр `database.allow-without-tls` управляет каналом «ЦС → СУБД», а не подключениями клиентов к самому ЦС (за них отвечает секция `tls`). Чтобы ЦС проверял сертификат сервера БД, добавьте в строку подключения `sslmode=verify-full`.

10.1.5 Журналирование

Подробнее см. главу Журналирование и мониторинг (see page 256), в том числе раздел «Отказоустойчивый журнал событий безопасности (secLog)».

Параметр	Назначение	Значение по умолчанию

`log.level`	Минимальный уровень регистрируемых событий: от `flood` до `fatal`	`info`
`log.format`	Формат записей журнала: `json`, `logfmt`, `tinted`, `default`	`json`
`log.file`	Файл журнала службы	`logs/minica.log`
`log.fileMode`	Права на файл журнала	`0600`
`log.rotate.maxSize` `log.rotate.maxAge`	Ротация журнала: размер файла, МБ, и срок хранения старых файлов, дней	10 365
`log.sumChain`	Цепочка контрольных сумм записей: изъятие, добавление или подмена записей обнаруживается при проверке	`true`
`log.secLog.path`	Файл отказоустойчивого журнала событий безопасности. Если путь задан, механизм включён	`logs/security.log`

10.2 Приложение Б. Сетевые порты

Сводный перечень портов Clearway CA для правил межсетевого экрана при установке по умолчанию. Какие службы слушают порты и как они публикуются, описано в разделе «Службы и сетевые порты» главы «Архитектура и состав системы», а требования к сети при подготовке серверов приведены в главе Требования к среде (see page 38). Порядок закрытия служебных портов и проверка описаны в разделе «Ограничение сетевого доступа» главы Обеспечение безопасной среды (see page 167).

10.2.1 Порты для клиентов

Эти порты должны быть доступны пользователям и приложениям, которые работают с ЦС.

Порт	Служба	Кто подключается	Назначение

443/tcp (HTTPS)	Nginx на сервере служб	Браузеры пользователей, клиенты SCEP и MS-WSTEP	Контрольная панель, сервис публикации (путь `/api/publication`), адаптер SCEP (путь `/api/v1/scep`), службы CEP и CES адаптера MS-WSTEP (пути `/CEP_Kerberos/` и `/CES_Kerberos/`)
443/tcp (HTTPS)	Сервис аутентификации OpenID	Браузеры пользователей; серверные части Контрольной панели, сервиса публикации и адаптера SCEP; сервис резервного копирования	Вход пользователей, выдача и проверка токенов
80/tcp (HTTP)	Nginx (точка распространения CDP)	Приложения, которые проверяют сертификаты	Загрузка CRL и Delta CRL (`http://<fqdn>/cdp/`) и сертификатов ЦС (`http://<fqdn>/aia/`)
8888/tcp (HTTP)	OCSP-ответчик mOCSP	Клиенты OCSP	Запросы статуса сертификата по адресу `http://<fqdn>:8888/ocsp`, без Nginx

10.2.2 Порты для взаимодействия компонентов

Эти порты открывают только для перечисленных серверов и рабочих мест.

Порт	**Служба**	**Кто подключается**	**Правило межсетевого экрана**
9443/tcp (HTTPS)	Ядро ЦС	Серверная часть Контрольной панели, сервис публикации, адаптер SCEP, службы MS-WSTEP; mclient на сервере ЦС	Разрешить только с серверов этих компонентов

5432/tcp	PostgreSQL	Ядро ЦС и mOCSP (к БД ЦС); архиватор (к БД ЦС и архивной БД); OpenID (к БД OpenID); сервер сервиса резервного копирования (к БД ЦС для заданий резервного копирования, порт задаётся в задании); АРМ администраторов БД	Разрешить только с перечисленных узлов
389/tcp, 636/tcp	Каталог LDAP или Active Directory	OpenID (коннектор LDAP); сервис публикации (задачи публикации в LDAP, по умолчанию порт 389); службы MS-WSTEP (контроллер домена, порт 389)	Разрешить исходящие подключения с этих серверов. После установки OpenID подключается к каталогу по порту 389 без шифрования; рекомендуется перейти на LDAPS (636)
22/tcp (SSH)	SSH (OpenSSH)	АРМ администраторов и рабочее место с установщиком → все серверы Clearway CA	Разрешить только с этих узлов
22/tcp (SSH)	SSH (OpenSSH)	Сервер сервиса резервного копирования → узлы ЦС (задания резервного копирования)	Разрешить только с сервера сервиса резервного копирования. Порт SSH задаётся в задании, по умолчанию 22
22/tcp (SSH)	SSH (OpenSSH)	Сервер Контрольной панели → серверы nginx и PostgreSQL (типовые задачи замены сертификата)	Разрешить только с сервера Контрольной панели
22/tcp (SSH)	SSH (OpenSSH)	Сервис публикации → серверы-получатели SCP (задачи публикации)	Разрешить исходящие подключения с сервера публикации. Порт SSH задаётся в задаче публикации, по умолчанию 22
Порт из `log.secLog.notifyEmailSmtp`	SMTP-сервер	Ядро ЦС, если включены уведомления журнала безопасности по почте (`log.secLog.notifyEmailEnabled`)	Разрешить исходящие подключения с сервера ЦС

Порт из `Mailing.Port` (по умолчанию 25)	SMTP-сервер	Сервис резервного копирования, если включены письма о неуспешных операциях (`Mailing.Enabled`)	Разрешить исходящие подключения с сервера сервиса резервного копирования

10.2.3 Служебные порты

Серверная часть Контрольной панели (8407), сервис публикации (9407), адаптер SCEP (8440) и архиватор (9507) запускаются с параметром `--urls="http://0.0.0.0:<порт>"` и принимают подключения по HTTP без TLS на всех сетевых интерфейсах сервера. Сервис резервного копирования принимает подключения по HTTP на порту из своего параметра `--urls` (в примерах 8409). Службы CEP и CES адаптера MS-WSTEP (8441, 8442) принимают подключения по HTTPS, тоже на всех интерфейсах. К веб-службам обращается Nginx на том же сервере, к сервису резервного копирования обращается серверная часть Контрольной панели, а архиватор клиентам не публикуется, поэтому эти порты закрывают межсетевым экраном. В таблице также порт HTTP ядра ЦС, служебный интерфейс OpenID и веб-интерфейс установщика.

Порт	**Служба**	**Кто подключается**	**Правило межсетевого экрана**
8407/tcp (HTTP)	Серверная часть Контрольной панели	Nginx на том же сервере (путь `/api/minica`)	Закрыть для подключений извне
9407/tcp (HTTP)	Сервис публикации	Nginx на том же сервере (путь `/api/publication`); серверная часть Контрольной панели по адресу из `Services.PublicationCrlUrl` (по умолчанию `http://localhost:9407`)	Закрыть для подключений извне. Если Контрольная панель работает на другом сервере, разрешить подключение только с него и указать адрес сервиса публикации в `Services.PublicationCrlUrl` конфигурации Контрольной панели
8440/tcp (HTTP)	Адаптер SCEP	Nginx на том же сервере (путь `/api/v1/scep`)	Закрыть для подключений извне

8441/tcp, 8442/tcp (HTTPS)	Службы CEP и CES адаптера MS-WSTEP	Nginx на том же сервере (пути `/CEP_Kerberos/` и `/CES_Kerberos/`)	Закрыть для подключений извне
9507/tcp (HTTP)	Архиватор	Клиентам не публикуется	Закрыть для подключений извне
8409/tcp (HTTP), порт задаётся параметром `--urls`	Сервис резервного копирования	Серверная часть Контрольной панели по адресу из `Services.BackupUrl` (например `http://localhost:8409`)	Закрыть для подключений извне. Если Контрольная панель работает на другом сервере, разрешить подключение только с него
5225/tcp на адресе 127.0.0.1 (параметр `SshAskPass.LocalPort`)	Канал askpass сервиса резервного копирования	Узлы ЦС через туннель SSH, открытый сервисом	Порт слушает только локальный адрес, отдельное правило не нужно
9080/tcp (HTTP)	Ядро ЦС при выключенном TLS	После установки не используется: установщик включает TLS	Держать закрытым
5559/tcp (HTTP)	Служебный интерфейс администрирования OpenID (параметр `adminServer.http`)	Клиентам не публикуется	Клиентам не открывать; при необходимости разрешить подключение только с АРМ администраторов
5224/tcp (HTTPS)	Веб-интерфейс установщика на рабочем месте оператора	Браузер на том же рабочем месте или через SSH-туннель	Разрешить только с АРМ администратора и только на время установки

10.3 Приложение B. Матрица ролей и прав

Права пользователей, стандартные роли и матрица доступа описаны в главе Описание ролевой модели (see page 47). Приложение даёт краткую сводку и показывает, где искать подробности.

Основное о правах:

- В Clearway CA 26.3.0 определено 17 прав с идентификаторами вида `rleMiniCA<Право>`: суффиксы `-r` и `-w` обозначают соответственно просмотр и изменение.
- Права `rleMiniCABackup-r`, `rleMiniCABackup-w`, `rleMiniCABackupRestore-w` и `rleMiniCARotation-w` добавлены в версии 26.3.0.
- Права назначаются членством в группах каталога AD/LDAP по карте `ldapGroupRolesMap` в файле `/opt/itc/openid/config.yaml` и передаются в токене, в claim `roles`.
- Изменение членства в группах и карты `ldapGroupRolesMap` начинает действовать, когда пользователь выйдет из Контрольной панели и войдёт снова. После правки карты также перезапустите сервис OpenID (`openid.service`).

Стандартные роли (карта групп `ldapGroupRolesMap` из поставки):

- Администратор ЦС: группа `CWCA_Administrator`, полный набор прав. В 26.3.0 их 17; если в карте перечислено 13, дополните её (см. «Добавление новых прав 26.3.0 в карту групп»).
- Аудитор: группа `CWCA_Auditor`, 6 прав на просмотр.
- Пользователь: группа `CWCA_User`, 3 права (просмотр сертификатов и шаблонов, создание запросов на выпуск).

Что нужно узнать	Где описано
Что разрешает каждое право	Раздел «Перечень прав»
Какие права нужны, чтобы увидеть пункт меню Контрольной панели	Раздел «Права и пункты меню Контрольной панели»
Какие права входят в стандартные роли (матрица доступа)	Раздел «Стандартные роли и матрица доступа» (п. 4.3)
Как добавить права версии 26.3.0 группе администраторов	Раздел «Добавление новых прав 26.3.0 в карту групп»
Как назначить роль пользователю и изменить карту групп	Назначение ролей через группы AD/LDAP (see page 196)
Для чего маркеры `rleMiniCAOperator`, `rleMiniCAAuditor`, `rleMiniCAMaster`	Раздел «Составные роли-маркеры»
Как ограничить доступ к отдельному шаблону сертификата	Раздел «Разграничение доступа к шаблонам»

Какие функции API ЦС доступны сервисам и mclient	Раздел «Сервисные роли JWT (ролевая модель)»

10.4 Приложение Г. Основные команды mclient

Консольный клиент mclient управляет ЦС из командной строки. Он устанавливается вместе с ядром ЦС в каталог `/opt/itc/minica` и обращается к API ЦС так же, как другие компоненты: с JWT и подписью запросов. Доступные команды определяет роль в JWT клиента (параметр `jwt.roles`, см. «Сервисные роли JWT (ролевая модель)»); роль `admin` разрешает все 27 команд.

10.4.1 Подготовка к работе

Команды выполняются на сервере ЦС от имени технологической учётной записи `itc-svc` из каталога ЦС. Перед вызовом укажите путь к конфигурации клиента в переменной `MCLIENT_CONF` и проверьте связь с ЦС:

```
su - itc-svc
cd /opt/itc/minica
export MCLIENT_CONF=/opt/itc/minica/conf/mclient.yml
./mclient ping
```

При успехе команда выводит версию ЦС и SKI (идентификатор ключа ЦС). Если команда завершается ошибкой, см. главу «Диагностика и устранение неисправностей».

В файле `/opt/itc/minica/conf/mclient.yml` заданы адрес ЦС (`server-url`, после установки `https://<fqdn>:9443`), ключ подписи запросов (`key-file`, после установки `/opt/itc/minica/conf/mclient.key`), JWT клиента (`jwt-file`, после установки `/opt/itc/minica/conf/mclient.jwt`) и параметры TLS клиента (секция `tls`).

Общий вид вызова: `./mclient [флаги] <команда> [параметры]`. Параметры подключения к ЦС можно также задать флагами, которые указываются перед командой, или переменными окружения:

Флаг	Переменная окружения	Назначение
`-conf <файл>`	`MCLIENT_CONF`	Файл конфигурации клиента (YAML или JSON)
`-url <URL>`	`MINICA_URL`	Адрес ЦС

`-key <файл>`	`MCLIENT_KEY`	Закрытый ключ для подписи запросов (PEM)
`-pass <пароль>`	`MCLIENT_PASS`	Пароль зашифрованного закрытого ключа
`-jwt <файл>`	`MCLIENT_JWT`	Файл JWT клиента
`-k`	Нет	Подключаться к ЦС по HTTPS без проверки его сертификата. Только для отладки

Пароль, переданный флагом `-pass`, виден другим пользователям сервера в списке процессов. Передавайте его переменной `MCLIENT_PASS`.

Локальные команды выполняются без обращения к ЦС:

- `help`: полная справка по командам и параметрам;
- `-h`: краткая справка по флагам;
- `version`: версия mclient;
- `showconf`: вывести действующую конфигурацию клиента в формате JSON;
- `mkconf [файл]`: создать файл конфигурации по умолчанию в формате JSON.

10.4.2 Команды

В таблицах приведены назначение команд и их основные параметры. Сертификат указывается серийным номером (`-serial`), отпечатком (`-fingerprint`) или идентификатором в БД (`-id`). JWT указывается значением `-jwt-sub` (SKI ключа клиента, 20 байт в HEX) или `-jwt-jti` (идентификатор токена, UUID). Параметр `-out` задаёт файл результата, а в команде `export` указывает каталог.

Проверка и статистика:

Команда	Назначение	Основные параметры
`ping`	Проверка связи с ЦС: выводит версию ЦС и SKI его ключа	Нет
`stat`	Показатели ЦС на момент вызова: число действительных, отозванных и истёкших сертификатов, запросов, CRL, шаблонов, событий и JWT	Нет

Те же показатели показывает страница «Статистика ЦС» Контрольной панели: она выводит результат последнего расчёта, который выполняется по расписанию.

Сертификаты:

Команда	Назначение	Основные параметры
`ca`	Выпуск сертификата по CSR	`-csr`, `-template`, `-days`, `-out`; `-nodb` (выпуск без сохранения в БД)
`status`	Статус сертификата: действителен, отозван, истёк, неизвестен	`-serial`, `-fingerprint`, `-id`
`get`	Получение сертификата из БД в формате PEM	`-serial`, `-fingerprint`, `-id`, `-out`
`csr`	Получение CSR, по которому выпущен сертификат	`-serial`, `-fingerprint`, `-id`, `-out`
`find`	Поиск сертификатов по критериям с постраничной выборкой	`-ski`, `-subj`, `-cn`, `-num`, `-offset`
`export`	Выгрузка сертификатов по фильтру в каталог	`-status` (VALID, REVOKED, EXPIRED, UNKNOWN), `-num`, `-offset`, `-out`
`revoke`	Отзыв сертификата с указанием причины и комментария	`-serial`, `-fingerprint`, `-id`, `-reason`, `-comment`
`repair`	Восстановление сертификата, отозванного с причиной CertificateHold	`-serial`, `-fingerprint`, `-id`, `-comment`
`delete`	Удаление сертификата и связанного CSR из БД без возможности восстановления	`-serial`, `-fingerprint`
`chain`	Цепочка сертификатов от корневого ЦС до текущего и CN текущего ЦС	`-out`

Списки отзыва:

Команда	Назначение	Основные параметры
`updcrl`	Принудительное обновление полного CRL	`-aki`, `-next_update`, `-next_publish`
`crl`	Получение актуального полного CRL в файл	`-aki`
`updeltacrl`	Обновление разностного списка отзыва (Delta CRL)	`-aki`, `-next_update`, `-next_publish`
`deltacrl`	Получение актуального Delta CRL в файл	`-aki`

Параметр `-aki` выбирает ЦС по идентификатору ключа. В сертификатах и CRL этот идентификатор записан как AKI (идентификатор ключа издателя) и совпадает с SKI ключа ЦС, который их подписал. Без параметра команды работают со списками отзыва текущего ЦС. С параметром они работают со списками ЦС, SKI которого равен указанному значению, например прежнего ключа ЦС после ротации (см. «Выбор ключа по значению AKI» в разделе «Ротация ключа ЦС»).

Шаблоны:

Команда	Назначение	Основные параметры
`templates`	Список шаблонов сертификатов	Нет
`gettempl`	Полная конфигурация шаблона	`-name` и `-ver` или `-id`
`addtempl`	Создание шаблона выпуска	`-name`, `-desc`, `-days`, `-exts`, `-bc`, `-ku`, `-eku`, `-ext`, `-key-type`, `-key-size`
`deltempl`	Удаление шаблона	`-name` и `-ver` или `-id`

JWT:

Команда	Назначение	Основные параметры

`mkjwt`	Выпуск и подпись JWT	`-jwt-name`, `-jwt-role`, `-jwt-key`, `-jwt-exp`, `-out`
`getjwt`	Сведения о JWT из БД	`-jwt-sub` или `-jwt-jti`
`revokejwt`	Отзыв JWT	`-jwt-sub` или `-jwt-jti`
`repairjwt`	Восстановление отозванного JWT	`-jwt-sub` или `-jwt-jti`
`renewjwt`	Продление срока действия JWT	`-jwt-key`, `-jwt-exp`
`revivejwt`	Повторная активация истёкшего JWT	`-jwt-key`, `-jwt-exp`
`deljwt`	Удаление JWT из БД без возможности восстановления	`-jwt-sub` или `-jwt-jti`

Пример: выпуск сертификата по шаблону tls из файла запроса `request.csr` в файл `cert.crt`.

```
./mclient ca -csr request.csr -template tls -out cert.crt
```

Команды, которые меняют данные ЦС

Команды `ca`, `revoke`, `repair`, `delete`, `updcrl`, `updeltacrl`, `addtempl`, `deltempl`, `mkjwt`, `revokejwt`, `repairjwt`, `renewjwt`, `revivejwt` и `deljwt` изменяют данные ЦС. Команды `delete` и `deljwt` удаляют данные из БД без возможности восстановления. Сертификат, выпущенный командой `ca` с параметром `-nodb`, не сохраняется в БД и не учитывается при отзыве и формировании CRL.