Aladdin Enterprise CA

Служба обеспечения совместимости <gost@basealt.ru>, Константин Белаш

Version 1.0, 07.12.2023

Оглавление

1. Дистрибутив ОС Альт

В данной инструкции используются следующие ОС:

- `Альт Сервер 10.1` (репозиторий p10) — Центр сертификации (**ЦС**, корневой и подчинённый), Центр валидации (**ЦВ**), контроллер домена `SambaAD`;
- `Альт Рабочая станция 10.1` (репозиторий p10) — ПК в домене `SambaAD`.

С полным списком поддерживаемых дистрибутивов семейства ОС Альт можно ознакомиться в документации `Aladdin Enterprise CA` (**AeCA**) — https://www.aladdin-rd.ru/catalog/aladdin-eca/#documentation/ (На момент написания инструкции данная информация еще не добавлена).

В данной инструкции будет показана установка и настройка **AeCA**, а также совместная работа в домене `SambaAD`. Аутентификация пользователей `SambaAD` будет осуществляться по сертификату, хранящемуся на токене `JaCarta`.

AECA будет состоять из трёх компонентов:

- корневой центр сертификации (**ЦС**);
- подчинённый (корневому) **ЦС**;
- центр валидации (**ЦВ**).

На корневом **ЦС** будет выпущен сертификат для подчинённого **ЦС**, который в свою очередь будет выпускать сертификаты для контроллеров домена `SambaAD` и его пользователей.

Корневой **ЦС**, подчинённый **ЦС** и **ЦВ** устанавливаются на отдельные сервера.

2. Обновление ОС до актуального состояния

Процедуры установки, обновления и удаления **АеСА** выполняются администратором, обладающим правами суперпользователя компьютера. Перед установкой необходимо убедиться в выполнении следующих требований:

- На компьютере установлена поддерживаемая ОС Альт;
- ОС Альт и ядро обновлены из соответствующего дистрибутиву репозитория.

Рекомендуемая процедура обновления ОС и ядра:

```
$ su-
# apt-get update
# apt-get dist-upgrade
# update-kernel
# reboot

$ su-
# remove-old-kernels
# apt-get autoremove
# apt-get clean
```

3. Подготовка ПО для работы с токенами

3.1. Удаление конфликтующего ПО

Пакеты поддержки работы токенов `openct` конфликтуют с пакетами `opensc`, поэтому их необходимо удалить.

```
# apt-get remove --purge openct libopenct pcsc-lite-openct
```

3.2. Установка ПО

Для работы с токенами в `Aladdin Enterprise CA` необходимо установить следующие пакеты:

```
# apt-get install opensc pcsc-lite-ccid pcsc-lite pcsc-tools libjcpkcs11
```

- **`libjcpksc11`** — библиотеки `PKCS#11` вендора токенов Аладдин;
- **`opensc, pcsc-lite-ccid, pcsc-lite, pcsc-tools`** — утилиты и библиотеки, необходимые для обеспечения работы интерфейсов PC/SC(+CCID) и `PKCS#11`.

Если пакета поддержки работы токенов Аладдин в репозитории нет или с ним выявлены проблемы, то необходимо установить его с сайта производителя — https://www.aladdin-rd.ru/support/downloads/jacarta_client/ — Аладдин JaCarta (`libjcPKCS11-2`) (теперь ещё и в составе «Единого Клиента JaCarta»).

3.3. Замена `TCP` порта Альтератор

AeCA и Центр управления системой ОС Альт (**Альтератор**) в своей работе используют порт `8080`. Для устранения конфликта необходимо изменить порт.

Определите, занят ли порт 8080 веб-интерфейсом Альтератора (https://www.altlinux.org/Первое_знакомство_с_альтератором/):

```
# ss -ntulp | grep 8080
tcp   LISTEN 0      100                *:8080           *:*
users:(("ahttpd",pid=2670,fd=16))
```

Если вывод предыдущей команды не пустой, то порт 8080 занят. Отредактируйте конфигурационный файл приложения `alterator-fbi`, чтобы заменить порт (в примере меняем на 8085):

```
# sed -iE 's:\(server-port\s*\)8080:\18085:' /etc/ahttpd/ahttpd.conf
```

Перезапустите службу `ahttpd`:

```
# systemctl restart ahttpd.service
```

4. Установка AeCA

Шаги, выполняемые в этом разделе, одинаковы для корневого **ЦС**, подчинённого **ЦС** и **ЦВ**. О различиях в настройке будет указываться дополнительно.

4.1. Установка вспомогательных пакетов

```
# apt-get install git wget ant psmisc bc patch tar unzip sudo
```

4.2. Настройка sudo

Необходимо настроить `sudo`, так как скрипты установки `AeCA` используют это приложение. Разрешите членам группы `wheel` использовать `sudo`.

```
# sed -iE 's/# WHEEL_USERS = %wheel/WHEEL_USERS = %wheel/' /etc/sudoers
# sed -iE 's/# WHEEL_USERS ALL=(ALL:ALL) ALL/WHEEL_USERS ALL=(ALL:ALL) ALL/' \
    /etc/sudoers
```

4.3. Установка OpenJDK 11

Проверьте установленную версию `Java`:

```
# java -version
openjdk version "17.0.8" 2023-07-18
OpenJDK Runtime Environment (Red_Hat-17.0.8.0.7-alt1) (build 17.0.8+7)
OpenJDK 64-Bit Server VM (Red_Hat-17.0.8.0.7-alt1) (build 17.0.8+7, mixed mode, sharing)
```

Если получаете сообщение `java: команда не найдена`, то OpenJDK не установлен.

Если версия `OpenJDK` отличается от версии 11, то удалите её:

```
# apt-get remove java-17-openjdk-headless
```

Установите `OpenJDK` 11:

```
# apt-get install java-11-openjdk-devel
```

Убедитесь, что установлена и будет использоваться `OpenJDK` версии 11:

```
# java -version
openjdk version "11.0.19.1" 2023-04-18
```

```
OpenJDK Runtime Environment 18.9 (build 11.0.19.1+1)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.19.1+1, mixed mode, sharing)
```

4.4. Установка СУБД PostgreSQL 12

```
# apt-get install postgresql12-server
```

```
# /etc/init.d/postgresql initdb
```

Установите метод аутентификации `password` вместо `trust` для локальных сетевых соединений:

```
# sed -iE 's/\(host.*all.*all.*127.*\)trust$/\1password/g' \
    /var/lib/pgsql/data/pg_hba.conf
# sed -iE 's/\(host.*all.*all.*128.*\)trust$/\1password/g' \
    /var/lib/pgsql/data/pg_hba.conf
```

```
# cat /var/lib/pgsql/data/pg_hba.conf | grep 'host.*password'
host    all             all             127.0.0.1/32            password
host    all             all             ::1/128                 password
```

Включите автоматический запуск службы `postgresql`:

```
# systemctl enable --now postgresql
```

4.5. Настройка СУБД PostgreSQL 12

Создайте пользователя базы данных, саму базу данных и установите привилегии для созданного пользователя:

```
# psql -U postgres
psql (12.16)
Введите "help", чтобы получить справку.

postgres=# CREATE USER aeca;
CREATE ROLE
postgres=# ALTER USER aeca WITH PASSWORD 'aeca';
ALTER ROLE
postgres=# CREATE DATABASE aecatest;
CREATE DATABASE
postgres=# ALTER DATABASE aecatest OWNER TO aeca;
ALTER DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE aecatest TO aeca;
```

```
GRANT
postgres=# ALTER USER aeca SUPERUSER;
ALTER ROLE
postgres=# \q
```

4.6. Установка JC-WebClient

`JC-WebClient` используется для доступа к токенам `JaCarta` из веб-интерфейса.

Перейдите в каталог, содержащий дистрибутив **JC-WebClient** (в примере `/var/tmp`), и произведите установку:

```
# cd /var/tmp
# apt-get install ./JC-WebClient_4.3.3.1528_Alt_Linux_x64.rpm
```

4.7. Установка основных пакетов `AeCA`

ЦС (корневой и подчинённый), а также **ЦВ**, должны устанавливаться на разные сервера.

Перейдите в каталог, содержащий дистрибутив **AeCA** (в примере `/var/tmp`), и произведите установку **ЦС** (корневого или подчинённого):

```
# cd /var/tmp
# apt-get install ./aeca-ca-1.2.0.356_ro_x64_ru.rpm
```

Для установки **ЦВ** выполните следующую команду:

```
# cd /var/tmp
# apt-get install ./aeca-va_1.2.0.356_ro_x64_ru.rpm
```

5. Настройка основных компонентов AeCA

5.1. Развертывание AeCA

Для корневого и подчинённого **ЦС** запустите следующий скрипт:

```
# bash /opt/aecaCa/scripts/install.sh
```

Для **ЦВ** запустите следующий скрипт:

```
# bash /opt/aecaVa/scripts/install.sh
```

Далее в процессе развертывания необходимо ответить на вопросы установщика. Для **ЦВ** развертывание аналогично, только пути к файлам будут отличаться.

На вопрос, нужно ли заменить шаблоны конфигурационных файлов по умолчанию, отвечаем **Да**:

```
To install the Product correctly, you need to have correct EJBCA configuration files in
/opt/aecaCa/scripts/../properties/
Type [Yes] if you want to overwrite default template config files in
/opt/aecaCa/scripts/../dist/properties/ by values from /opt/aecaCa/scripts/config.sh
Type [No] if you already have correct config files in
/opt/aecaCa/scripts/../dist/properties/ and no need to overwrite it
Type [Cancel] to cancel this installation
1) Yes
2) No
3) Cancel
#? 1
```

На вопрос, нужно ли установить **WildFly Application Server**, отвечаем **Да**:

```
To install the Product correctly, you need to have installed WildFly Application Server
Type [Yes] if you want to install WildFly to /opt/aeca/wildfly
Type [No] if you already have installed and configured WildFly in /opt/aeca/wildfly and
want to save it
Type [Cancel] to cancel this installation
1) Yes
2) No
3) Cancel
#? 1
```

На вопрос, нужно ли установить **EJBCA**, отвечаем **Да**:

```
To install the Product correctly, you need to have installed EJBCA
Type [Yes] if you want to install EJBCA to /opt/aeca/ejbca
Type [No] if you already have configured EJBCA in /opt/aeca/ejbca and want to save it
Type [Cancel] to cancel this installation
1) Yes
2) No
3) Cancel
#? 1
```

На вопрос, нужно ли установить **AeCA**, отвечаем **Да**:

```
Do you really want to install AECA?
Type [Yes] if you want to install AECA to /opt/aeca/ejbca
Type [No] if you want to use only ejbca_ce_7_4_3_2 without any AECA possibilities
Type [Cancel] to cancel this installation
1) Yes
2) No
3) Cancel
#? 1
```

В итоге развертывания должны получить следующее сообщение:

```
=========================================
[SUCCESS] AECA successfully installed
=================================================
SUCCESS
restarting aecaca.service
INSTALLATION COMPLETED!
=================================================
You can now install the keystore, from /opt/aeca/p12, in your web browser, using the
password  9cdf44c0f6bd62e592e15dc6b1352ace99ad4e5b
You can find all the generated passwords in the file /opt/aeca/generated_passwords.txt
```

Другие возможные варианты установки **AeCA** описаны в официальной документации.

5.2. Настройка доступа к веб-интерфейсу AeCA

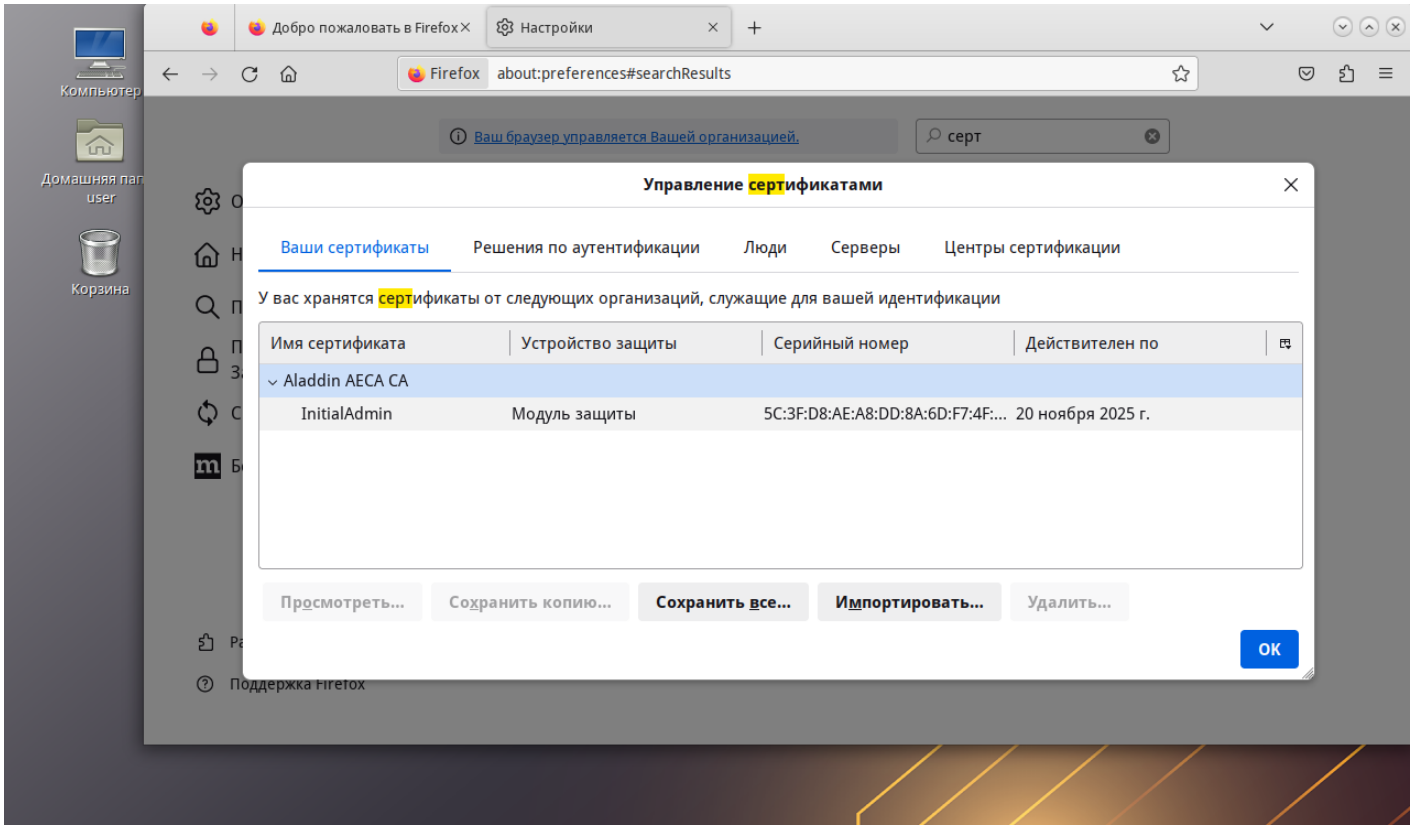
В процессе развертывания на каждом сервере **ЦС** и **ЦВ** создаётся контейнер с сертификатом для доступа к веб-интерфейсу управления **AeCA** — `/opt/aeca/p12/superadmin.p12`. Пароль доступа к контейнеру хранится в `/opt/aeca/generated_passwords.txt`.

Сертификат необходимо добавить в браузер (в примере Firefox) на каждом сервере **ЦС** и **ЦВ**. Для установки сертификата в браузере:

- **Настройки** –> **Приватность и Защита** –> **Сертификаты**. Нажмите кнопку **Просмотр сертификатов**

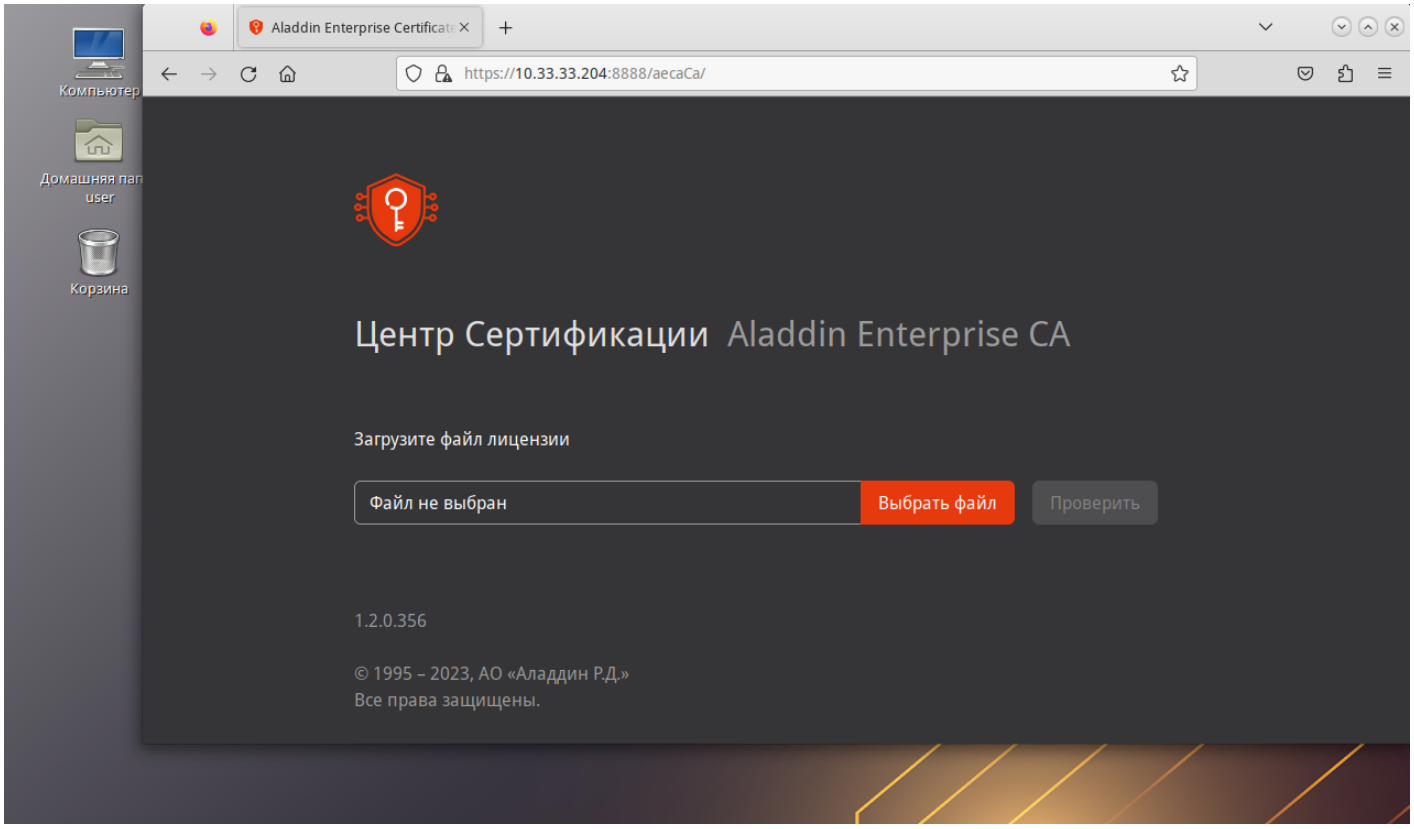
- Выберите вкладку **Ваши сертификаты**, и далее нажмите кнопку **Импортировать**

Выберите файл сертификата `/opt/aeca/p12/superadmin.p12`, созданный на этапе установки **AeCA**.



Перейдите в браузере на страницу `https://ip_address:8888/aecaCa/` (для **ЦВ** — `https://ip_address:8888/aecaVa/`). При появлении окна **Предупреждение: Вероятная угроза безопасности** выбираем **Дополнительно**, затем **Принять риск и продолжить**. Затем выбираем сертификат **InitialAdmin_ca**.

Если все настройки выполнены корректно, то в веб-интерфейсе **AeCA** будет предложено загрузить файл лицензии.

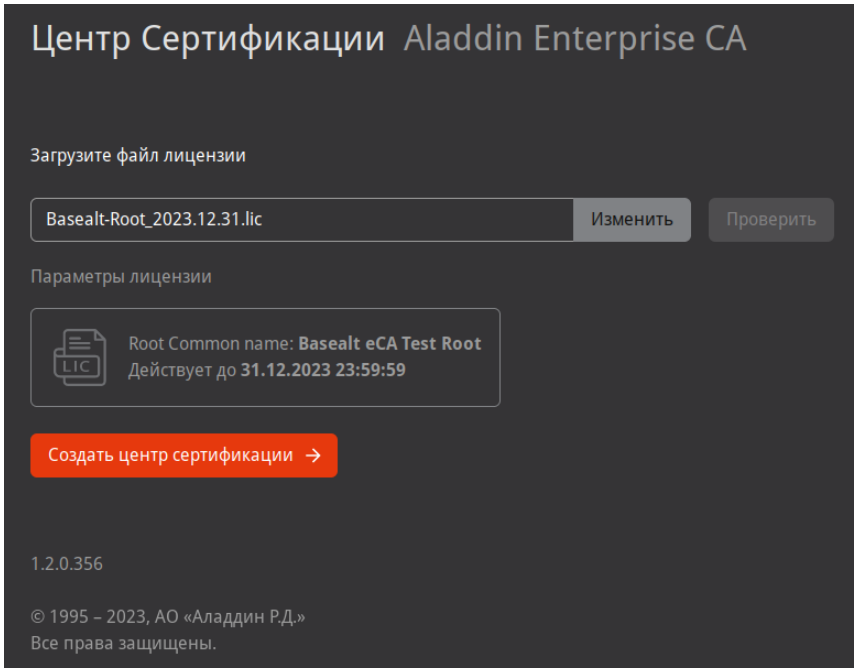


Если на сервере **ЦС** или **ЦВ** отсутствует графический интерфейс, то получить доступ можно с любого АРМ с графическим интерфейсом, предварительно добавив в браузер сертификаты (`superadmin.p12`) соответствующих серверов и установив `JC-WebClient`.

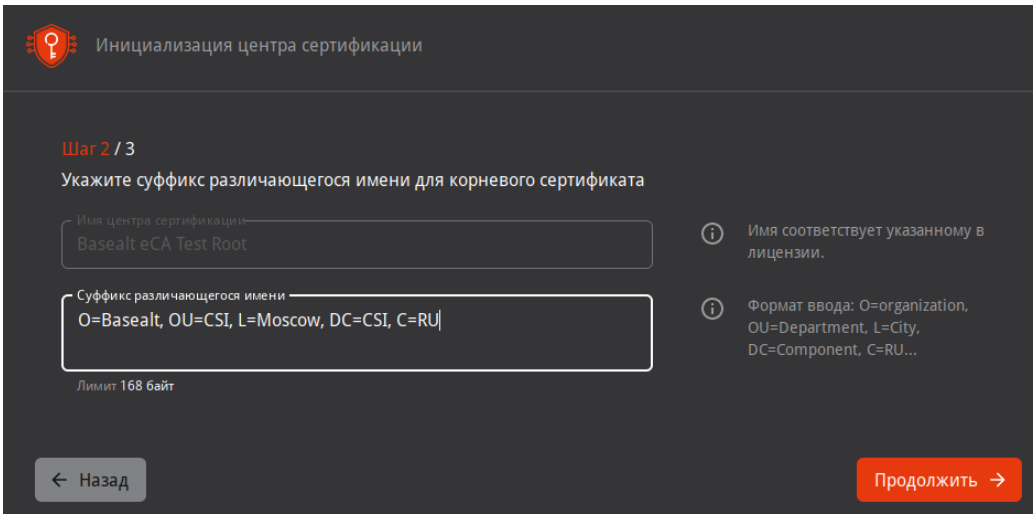
5.3. Добавление лицензии и активация корневого и подчинённого ЦС

На данном этапе необходимо добавить лицензию для корневого и подчинённого **ЦС**. Для **ЦВ** ввод лицензии не требуется.

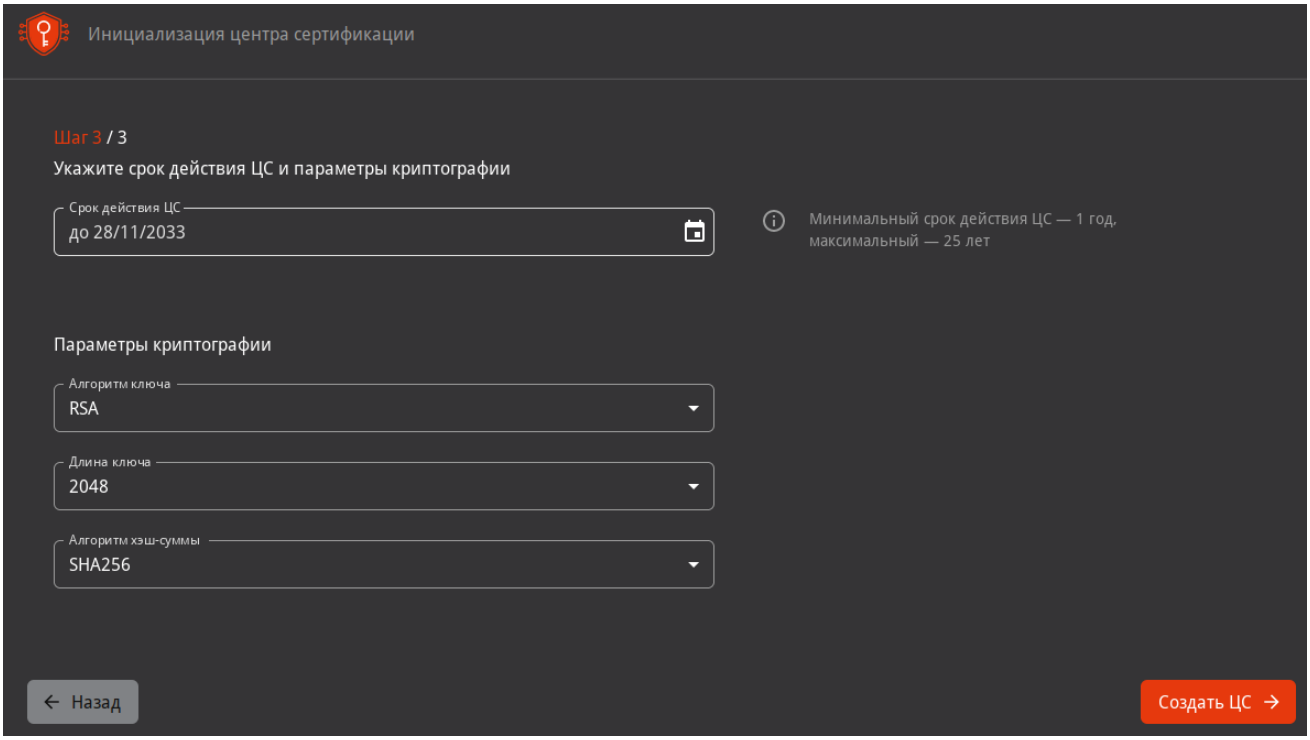
Нажмите **Изменить** и добавьте файл лицензии. Затем нажмите **Проверить**. В зависимости от загруженной лицензии **ЦС** становится либо корневым, либо подчинённым.



Нажмите **Создать центр сертификации**. На **Шаг 2** введите суффикс различающегося имени (в примере `O=Basealt, OU=CSI, L=Moscow, DC=CSI, C=RU`) и нажмите **Продолжить**.

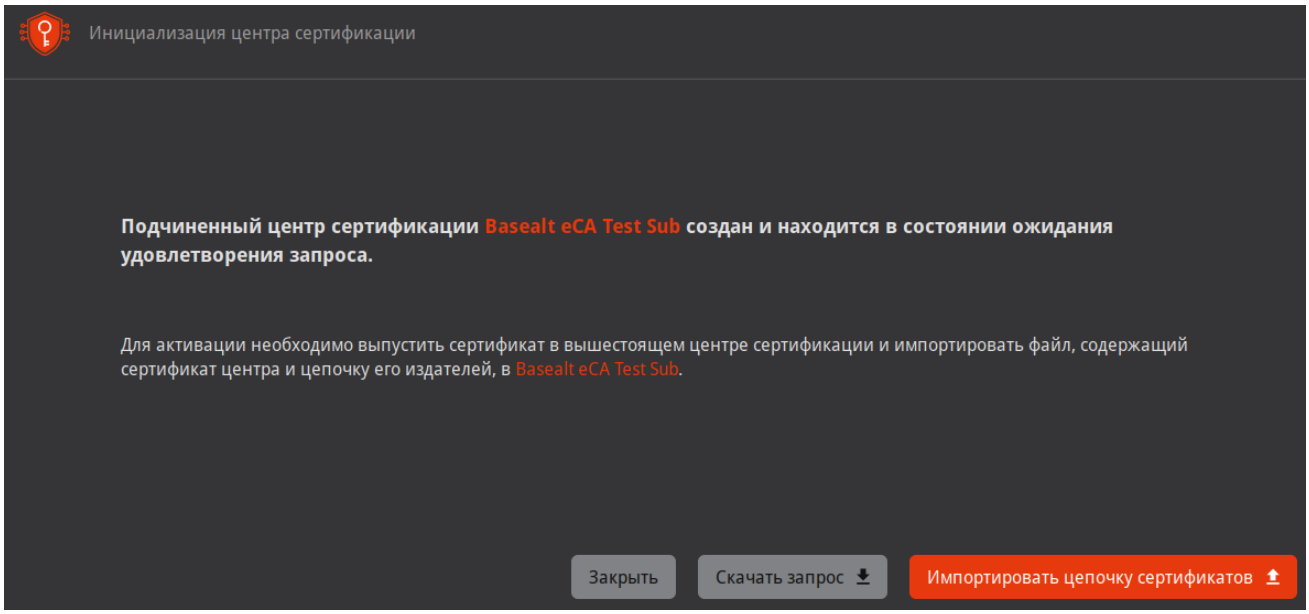


На **Шаг 3** заполните **Параметры криптографии** и нажмите **Создать ЦС**:



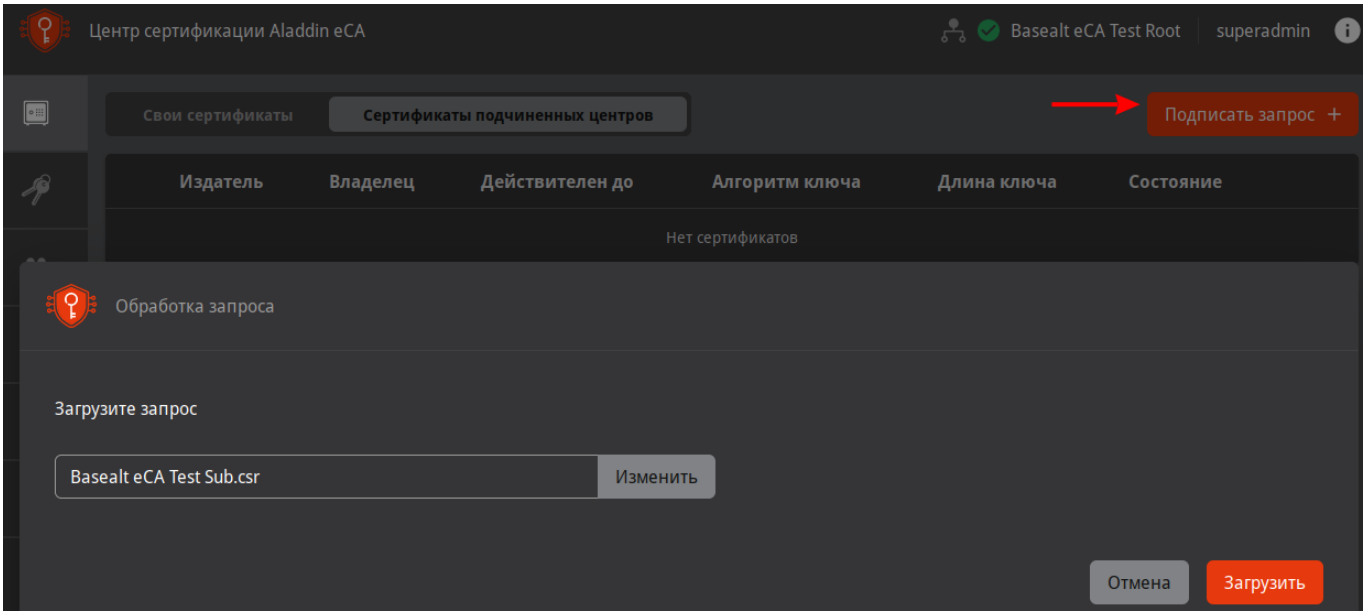
В итоге должны получить сообщение **Корневой центр сертификации Basealt eCA Test Root успешно создан и активирован.**

После добавления файла лицензии, указания суффикса различающегося имени и параметров криптографии для подчинённого **ЦС** последний перейдёт в состояние ожидания удовлетворения запроса:

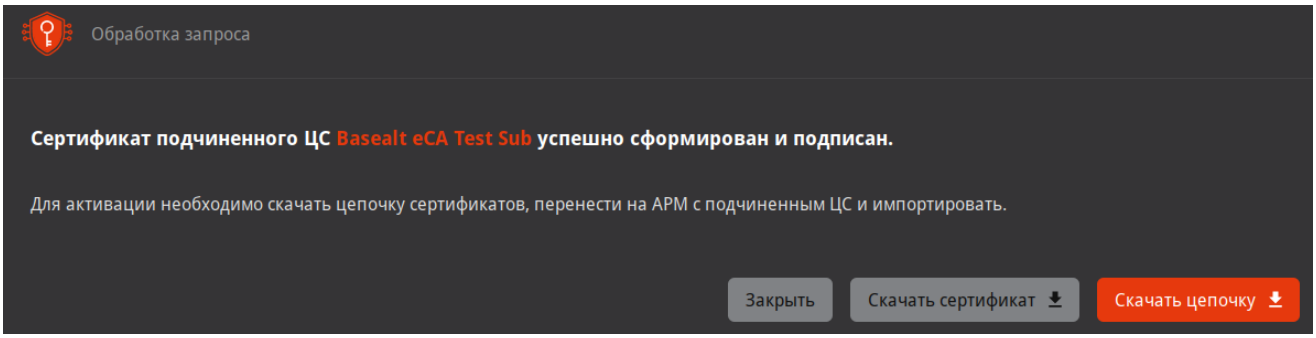


Нажмите **Скачать запрос**. Будет сформирован `.csr`, который необходимо перенести на корневой **ЦС**.

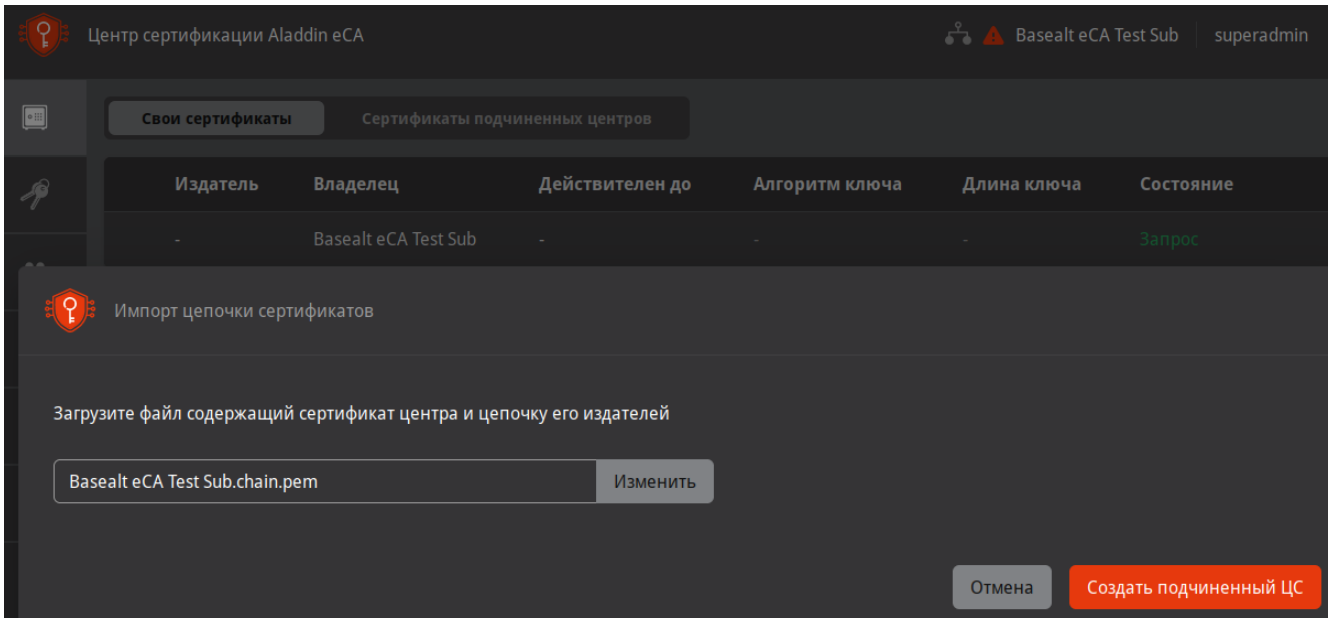
В веб-интерфейсе корневого **ЦС** перейдите в **Центр сертификации** - **Сертификаты подчинённых центров** и нажмите **Подписать запрос** +:



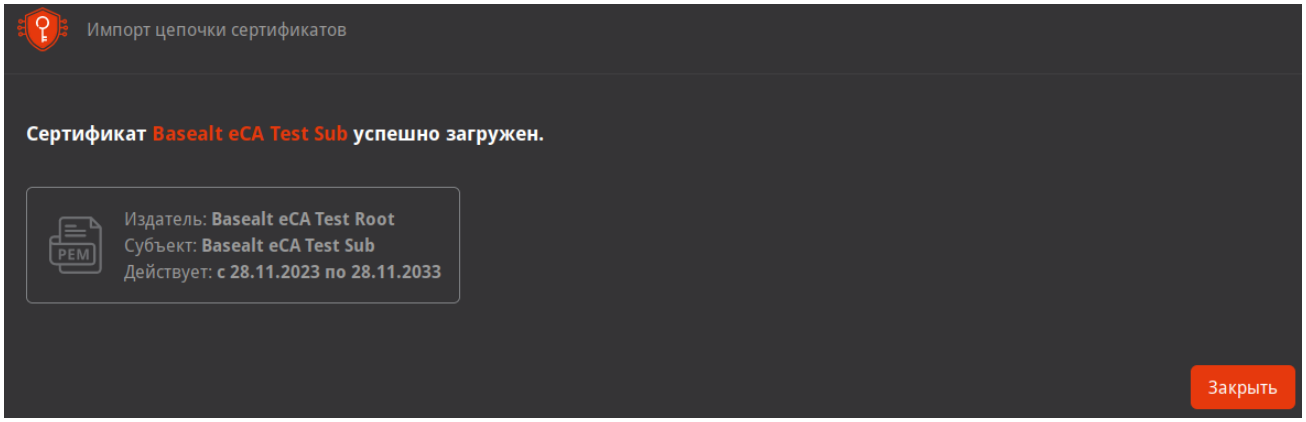
Загрузите перенесённый с подчинённого **ЦС** запрос (в примере `Basealt eCA Test Sub.csr`). После загрузки запроса должны получить сообщение **«Сертификат подчинённого ЦС Basealt eCA Test Sub успешно сформирован и подписан»**. Нажмите **Скачать цепочку**:



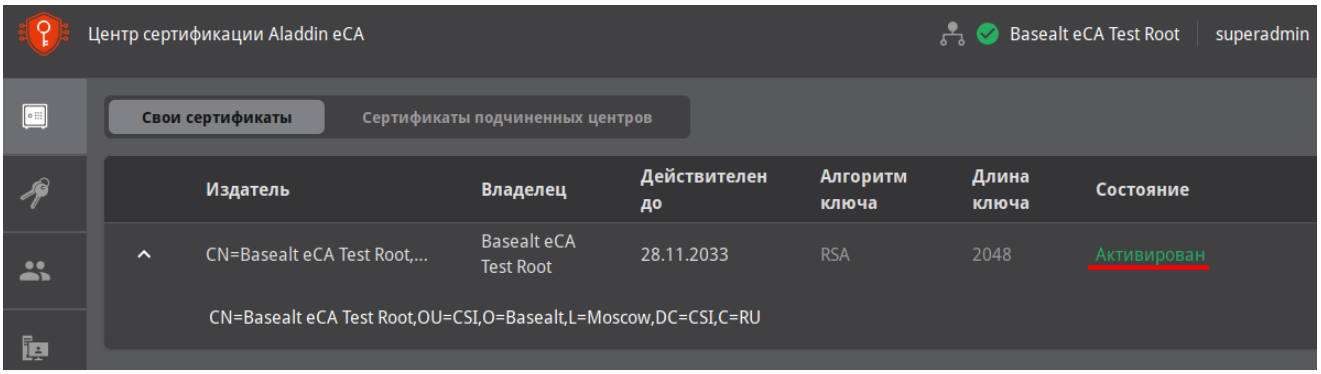
Перенесите скаченный файл цепочки сертификатов (в примере `Basealt eCA Test Sub.chain.pem`) на подчинённый **ЦС**. На подчинённом **ЦС** нажмите **Импортировать цепочку сертификатов**, затем **Создать подчинённый ЦС**:

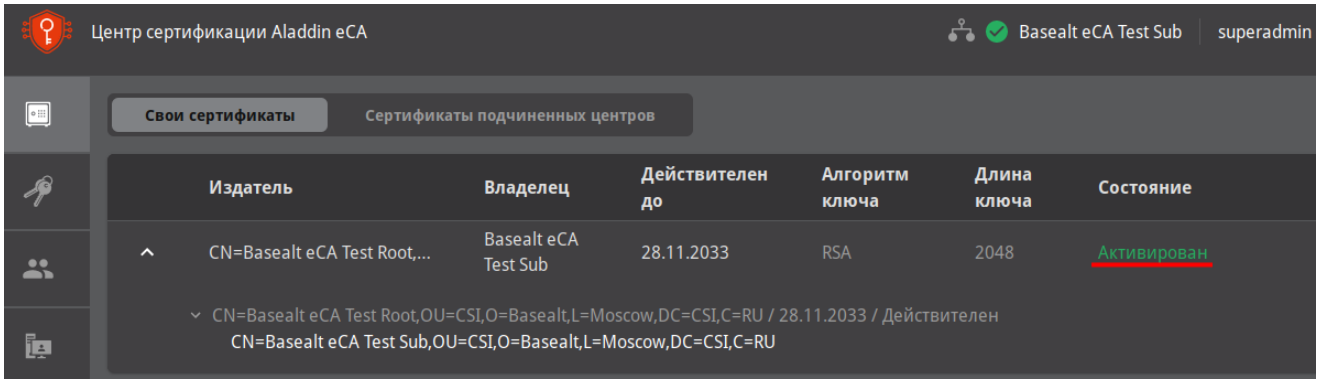


Должны получить сообщение **«Сертификат Basealt eCA Test Sub успешно загружен»**:



После вышеуказанных действий корневой и подчинённый **ЦС** должны быть в состоянии **Активирован**:

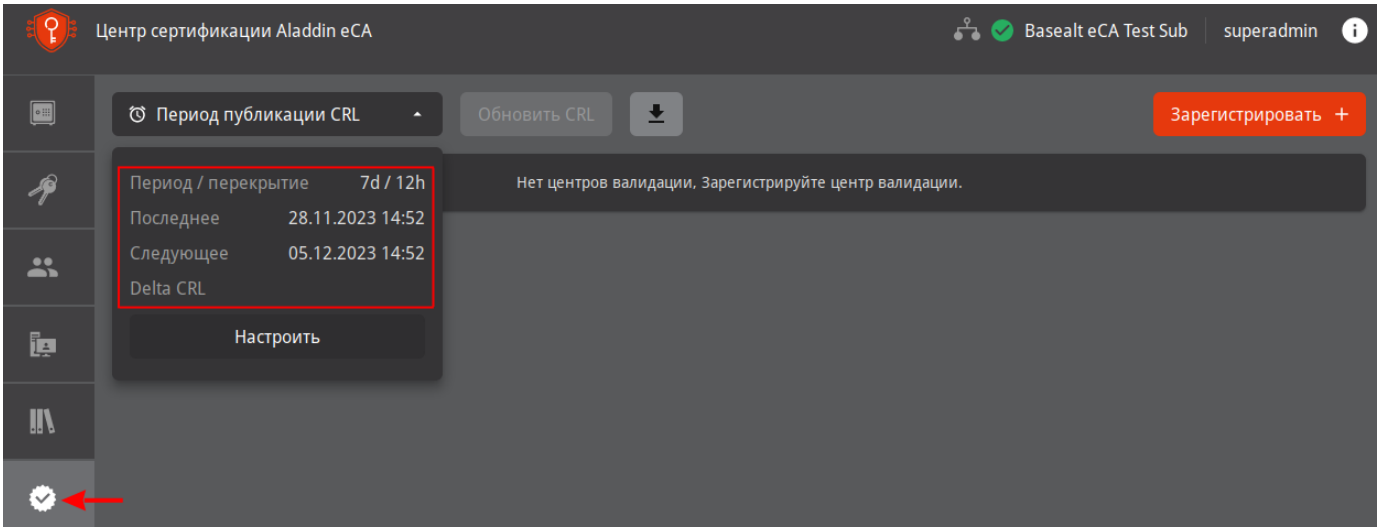




5.4. Регистрация ЦВ и активация службы OCSP

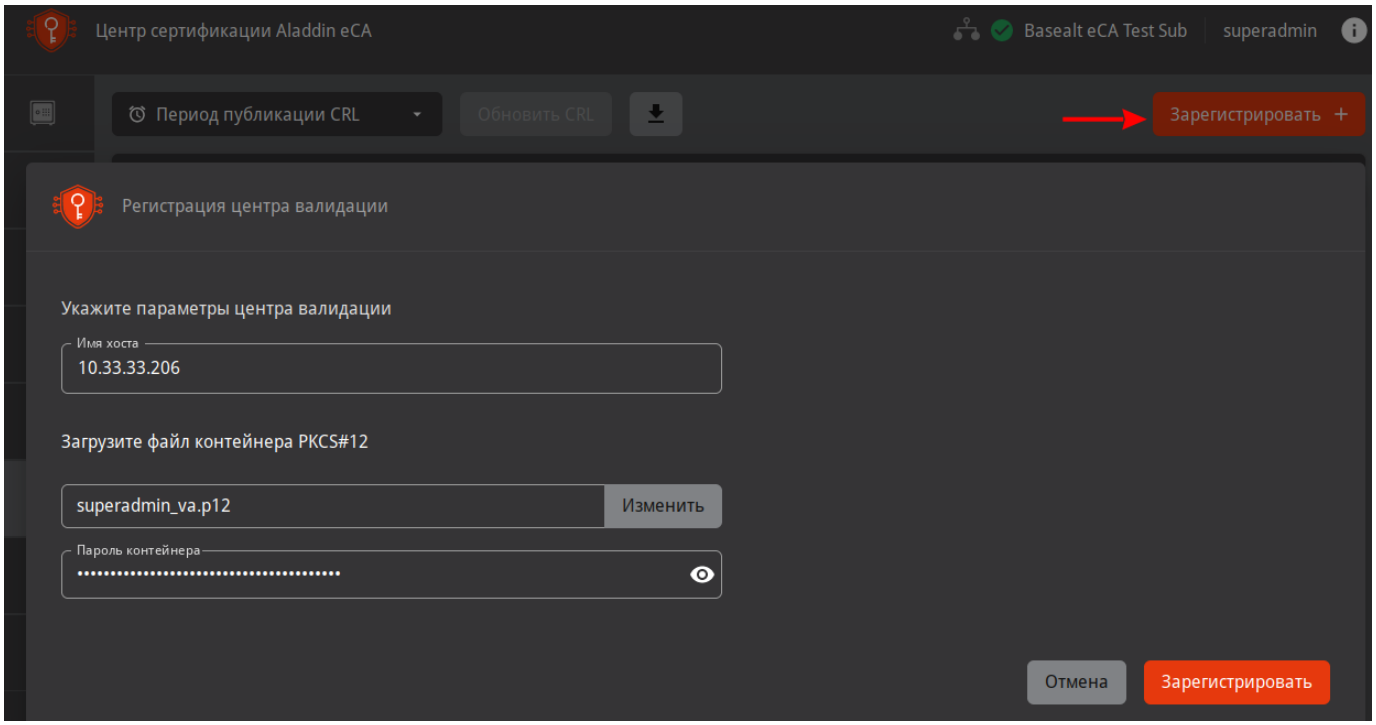
Регистрация **ЦВ** производится на подчинённом **ЦС**.

Убедитесь, что на подчинённом **ЦС** в **Центры валидации** уже настроен **Период публикации CRL**:

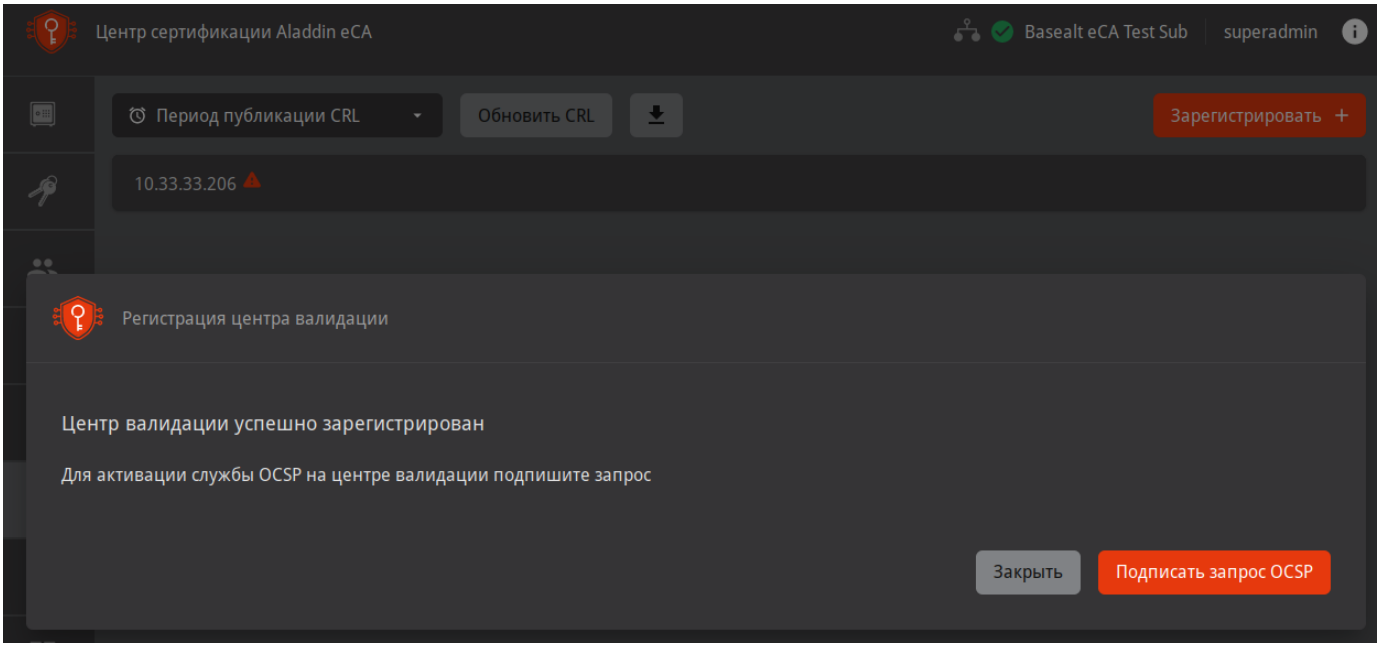


Если период публикации CRL не настроен, то это необходимо сделать.

В пункте меню **Центры валидации** нажмите **Зарегистрировать** +. В открывшемся окне укажите имя хоста, файл контейнера PKCS#12 для веб-интерфейса управления **ЦВ** и пароль от контейнера (расположение контейнера и файл с паролем указаны в Настройка доступа к веб-интерфейсу AeCA). Затем нажмите **Зарегистрировать**:

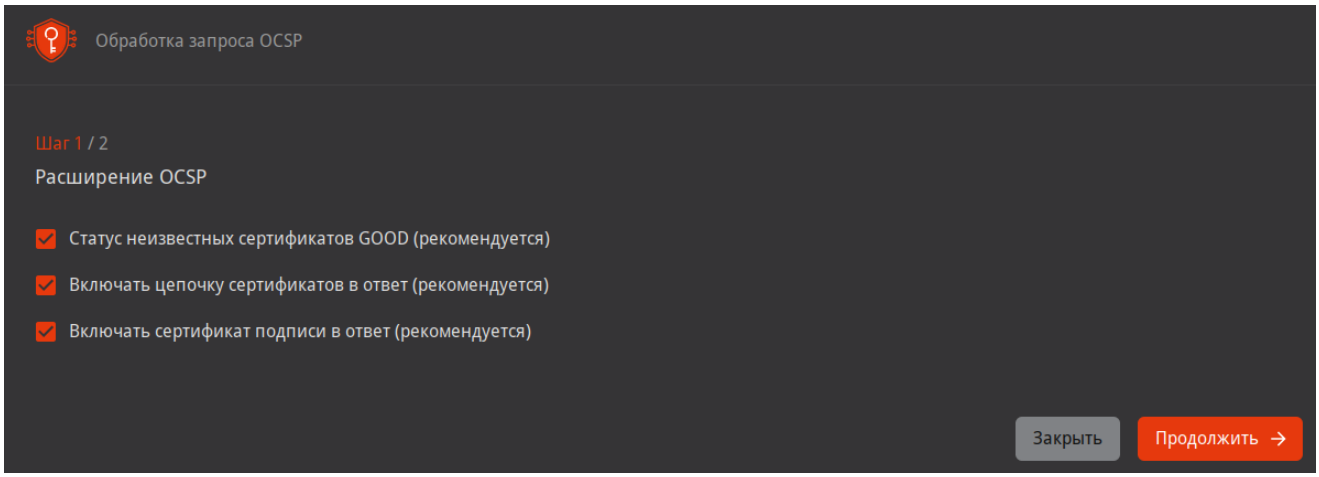


Должны получить сообщение об успешной регистрации **ЦВ**:

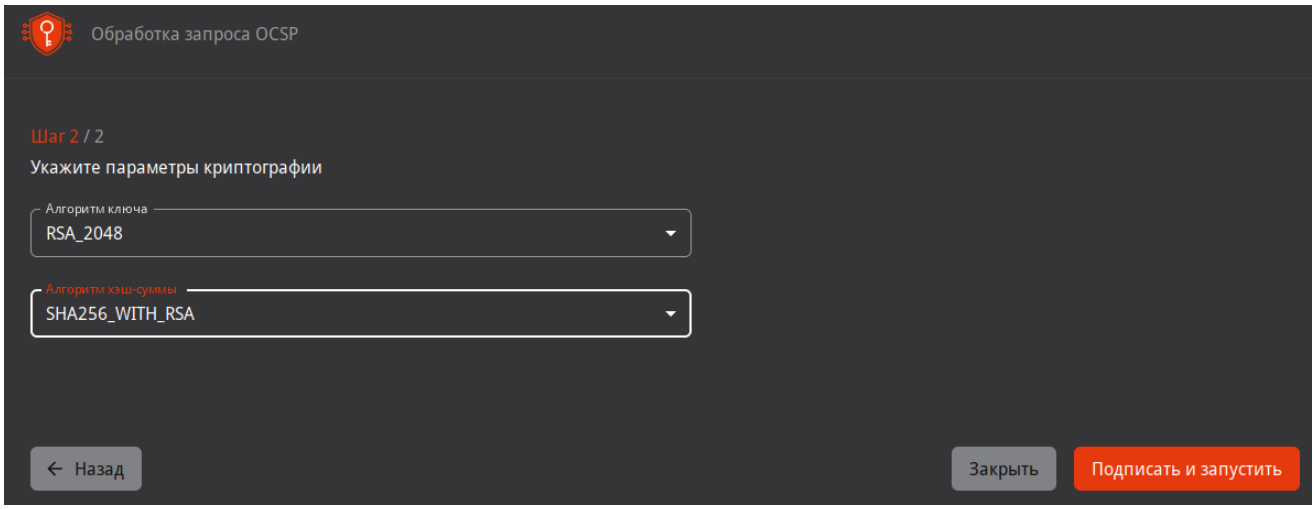


Нажмите кнопку **Подписать запрос OCSP**.

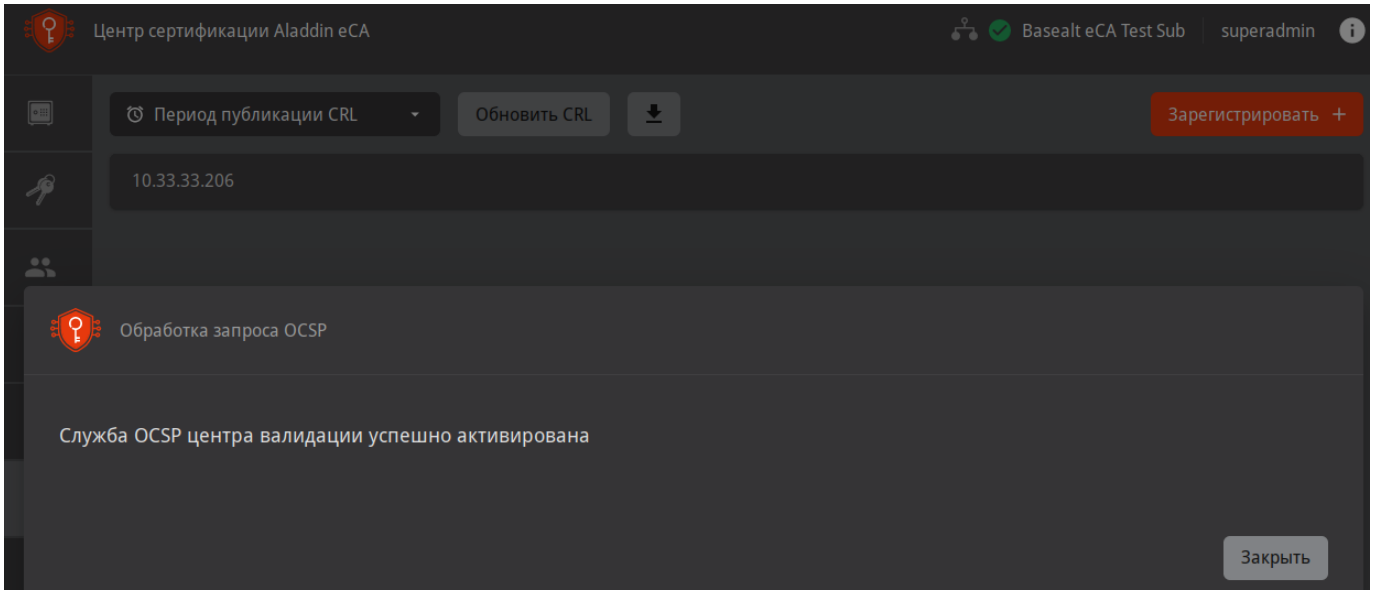
На **Шаг 1** отметьте пункты следующим образом:



На **Шаг 2** укажите параметры криптографии и нажмите **Подписать и запустить**:



Должны получить сообщение, что служба `OCSP` успешно активирована:


Центр сертификации Aladdin eCA
Basealt eCA Test Sub
superadmin
Период публикации CRL
Обновить CRL
Зарегистрировать +
10.33.33.206
Обработка запроса OCSP
Служба OCSP центра валидации успешно активирована
Закрыть

6. Подключение ресурсной системы

В качестве ресурсной системы будет использоваться доменная структура SambaAD.

При развертывании SambaAD на контроллере домена создаётся TLS сертификат — /var/lib/samba/private/tls/cert.pem. Он будет использоваться для подключения ресурсной системы к подчинённому **ЦС**. Сертификат необходимо конвертировать в формат DER и перенести на подчинённый **ЦС** (в примере 10.33.33.205):

```
# openssl x509 -outform der -in /var/lib/samba/private/tls/cert.pem \
    -out /var/tmp/smbdc.der
```

```
# scp /var/tmp/smbdc.der user@10.33.33.205:/var/tmp/
```

Прежде чем добавить сертификат smbdc.der определите на подчинённом **ЦС** расположение java:

```
$ dirname $(dirname $(readlink -f $(which javac)))
/usr/lib/jvm/java-11-openjdk-11.0.19.0.7-0.x86_64
```

Хранилище ключей и сертификатов для java располагается в поддиректории lib/security/cacerts. Следовательно, полный путь — /usr/lib/jvm/java-11-openjdk-11.0.19.0.7-0.x86_64/lib/security/cacerts.

Добавьте сертификат SambaAD на подчинённом **ЦС** (пароль к хранилищу по умолчанию — "changeit"), используя вычисленный полный путь:

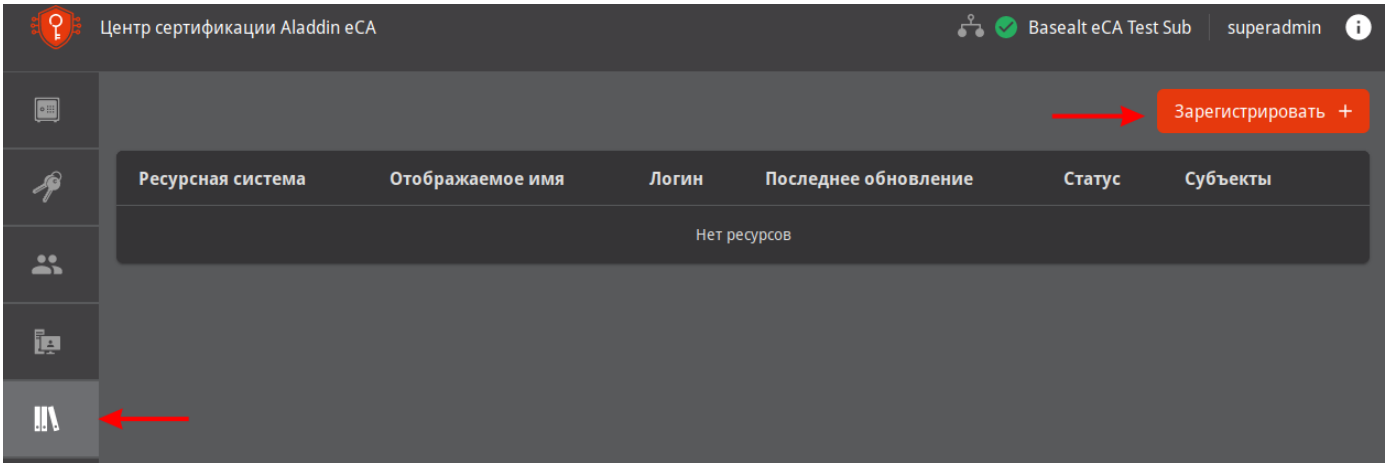
```
$ su -
# keytool -import -alias smb-cert -keystore \
    /usr/lib/jvm/java-11-openjdk-11.0.19.0.7-0.x86_64/lib/security/cacerts \
    -storepass changeit -file /var/tmp/smbdc.der
Warning: use -cacerts option to access cacerts keystore
Owner: CN=SRV101-VM203.test5.alt, OU=Samba - temporary autogenerated HOST certificate,
O=Samba Administration
Issuer: CN=SRV101-VM203.test5.alt, OU=Samba - temporary autogenerated CA certificate,
O=Samba Administration
Serial number: -75a2d99b
Valid from: Wed Oct 11 11:32:10 MSK 2023 until: Wed Sep 10 11:32:10 MSK 2025
Certificate fingerprints:
     SHA1: C8:D0:9C:58:FE:19:A0:2A:99:1F:B8:A7:6F:7D:1E:C0:A3:60:EE:67
     SHA256:
D8:7C:AF:10:E1:24:87:DF:C6:C7:E8:C4:9B:12:F4:8C:EA:37:F5:69:F8:EE:C1:CC:74:67:EE:3B:2A:29:9
5:88
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3
```

```
Extensions:

#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:false
  PathLen: undefined
]

#2: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  serverAuth
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: EB D0 2F 6A 16 BE 52 E2   F9 D6 84 0E 42 9C D7 3B  ../j..R.....B..;
0010: 55 81 EB 2A                                        U..*
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
```

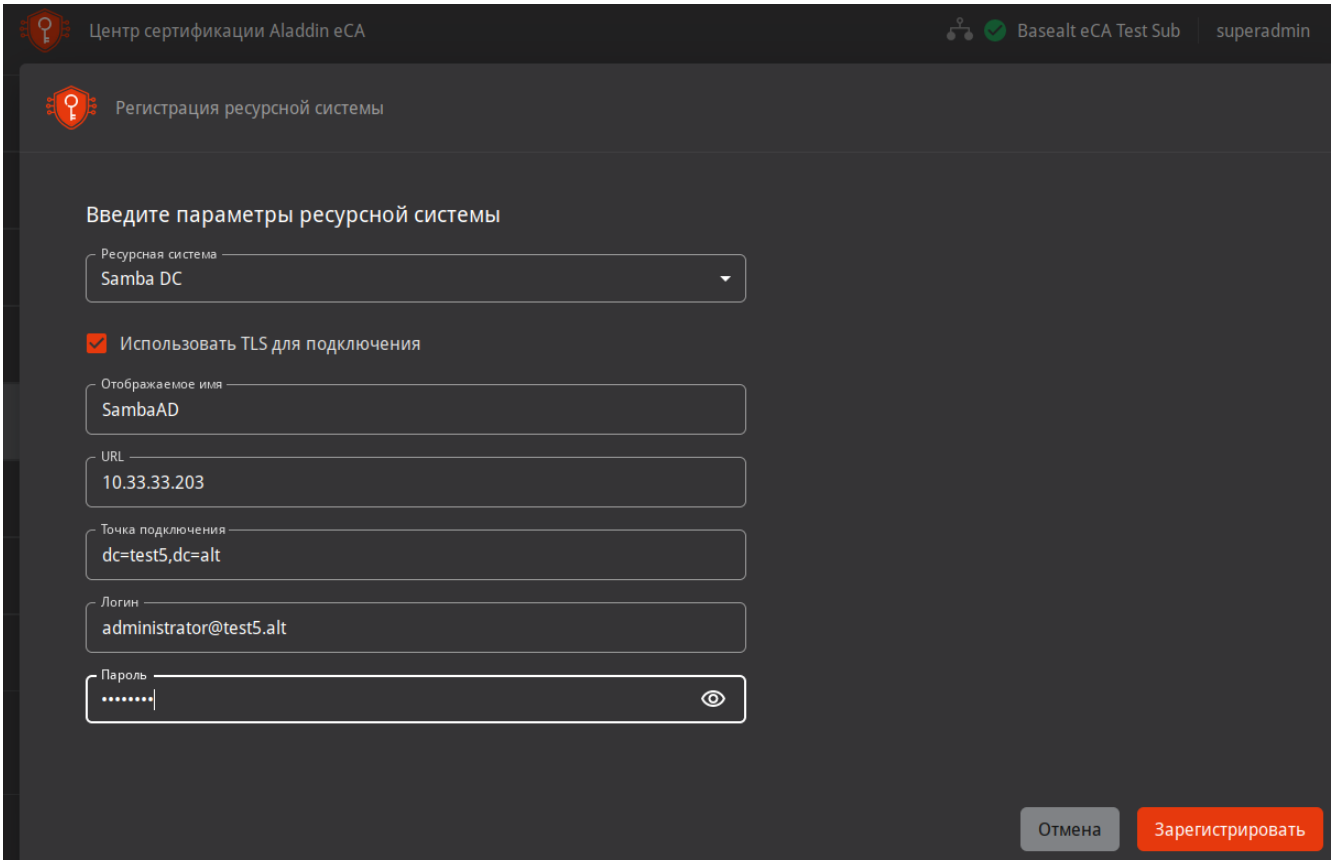
После добавления сертификата `SambaAD` перезапустите службу `aecaca.service`:

```
# systemctl restart aecaca.service
```

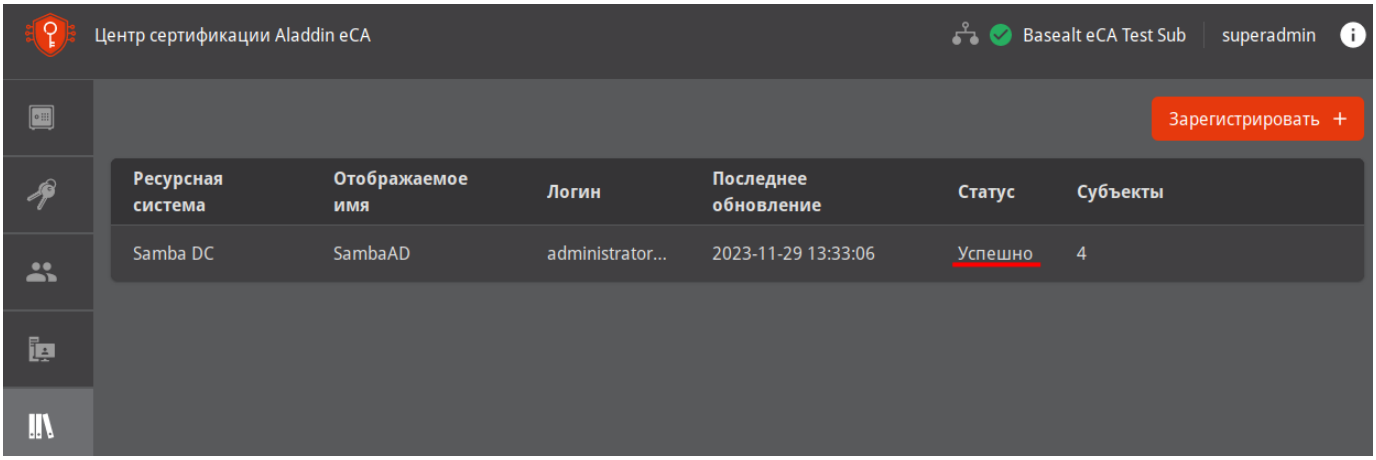
В веб-интерфейсе подчинённого **ЦС** перейдите в **Ресурсные системы** и нажмите **Зарегистрировать** +:



Заполните параметры ресурсной системы и нажмите **Зарегистрировать**:



Убеждаемся, что ресурсная система успешно подключена:

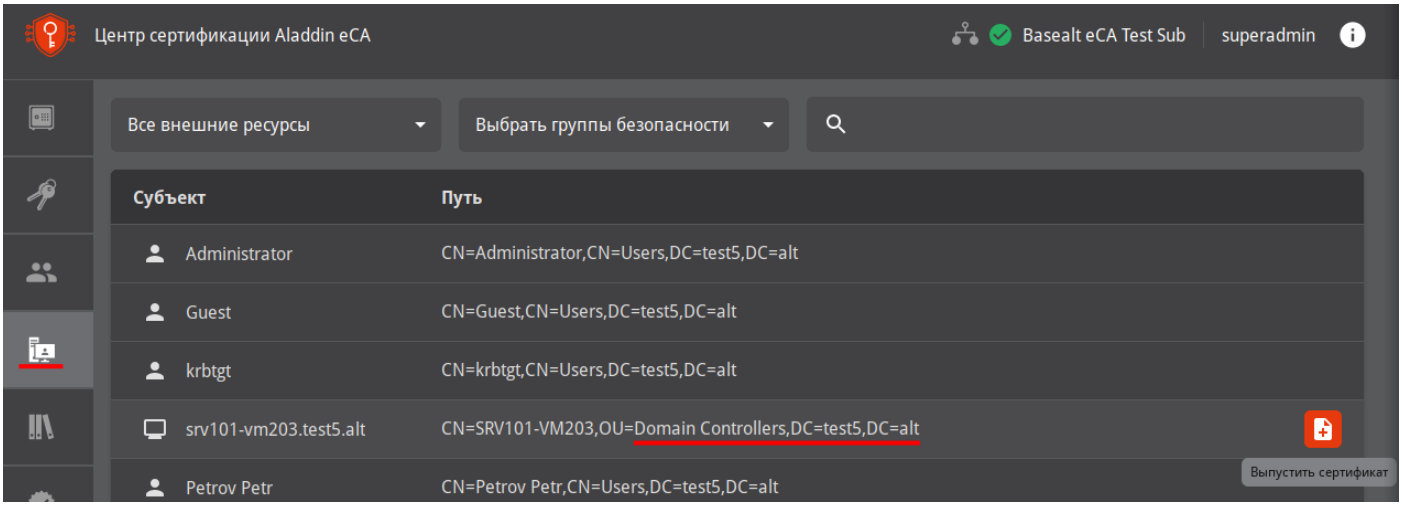


7. Обеспечение возможности строгой аутентификации пользователей в домене

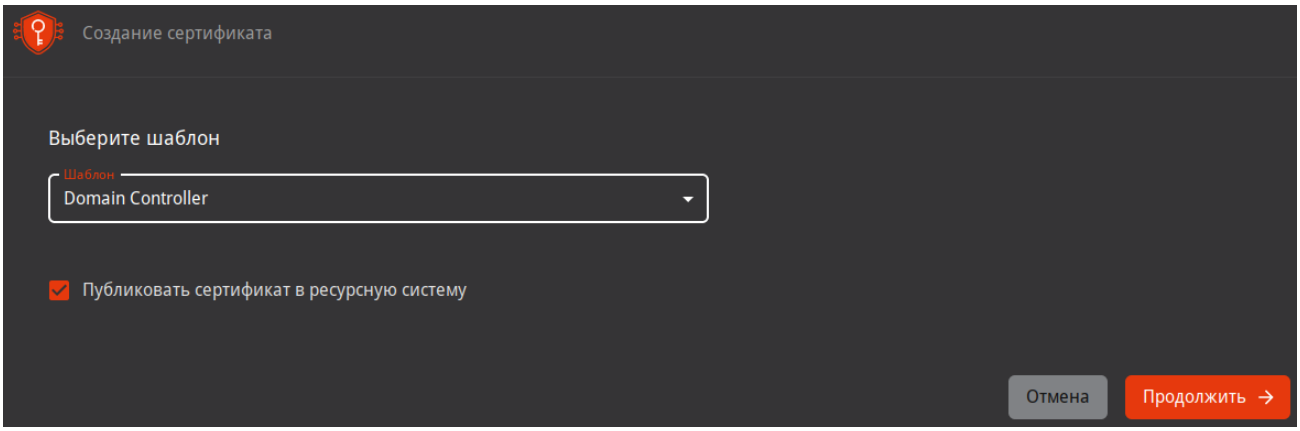
Под строгой аутентификацией подразумевается аутентификация при помощи сертификата, выданного в **АеСА** для доменного пользователя и размещённого на токене.

7.1. Выдача сертификата контроллера домена

В веб-интерфейсе подчинённого **ЦС** перейдите в **Субъекты**, выберите контроллер домена и нажмите **Выпустить сертификат. С закрытым ключом (PKCS#12)**:

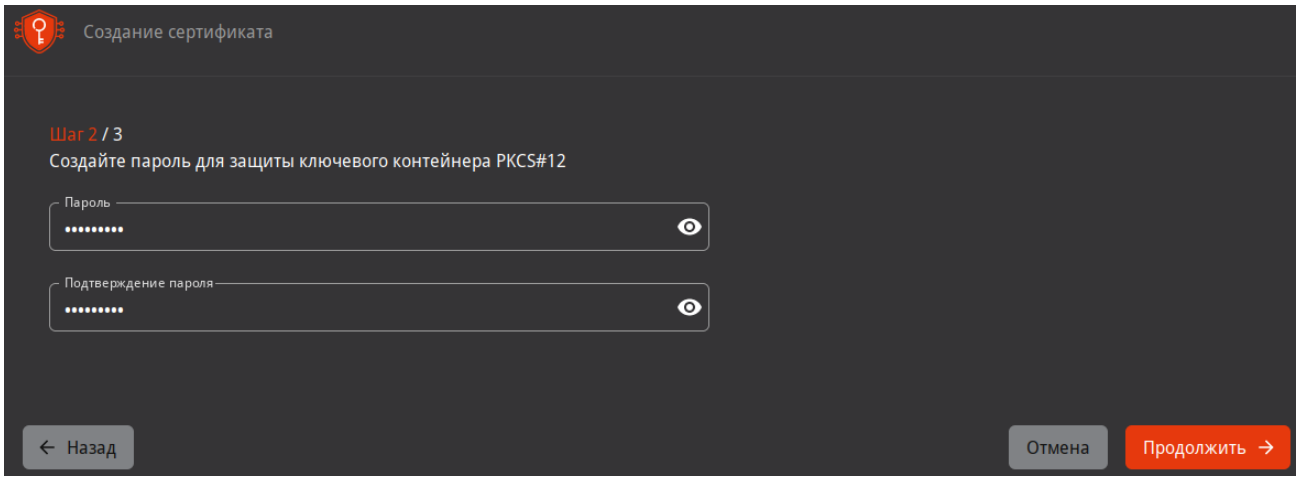


В качестве шаблона укажите **Domain Controller** и нажмите **Продолжить**:

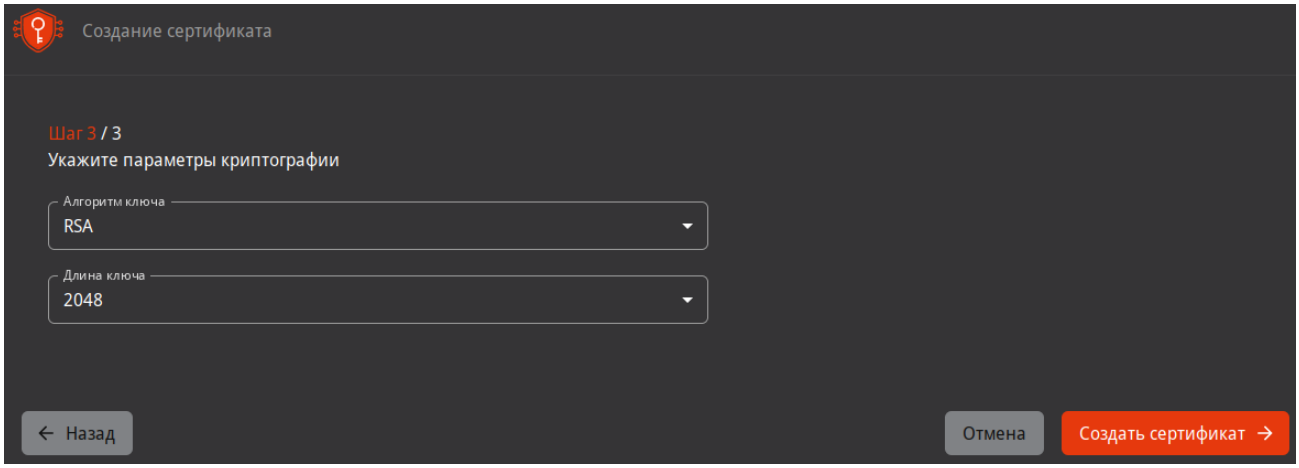


На **Шаг 1** данные контроллера домена должны заполниться автоматически:

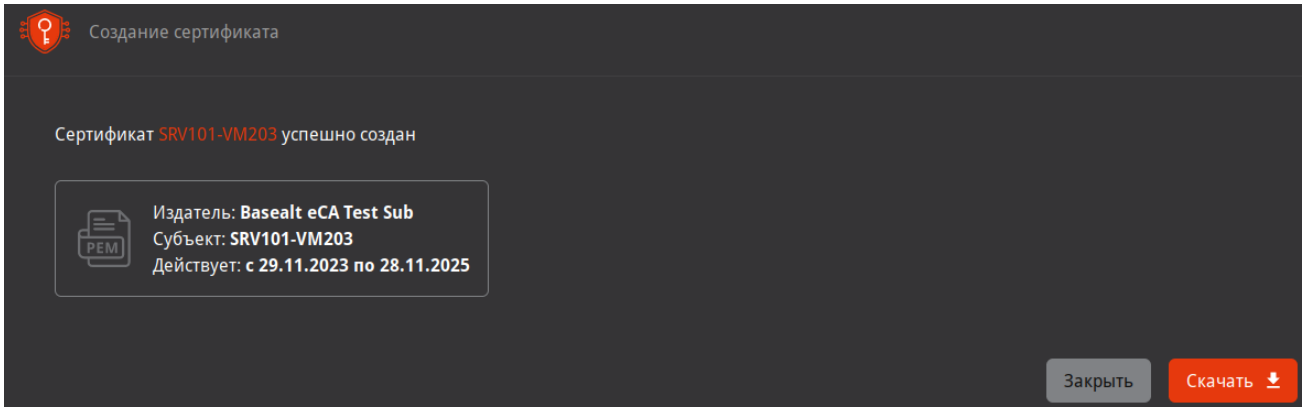
На **Шаг 2** укажите пароль для ключевого контейнера:



На **Шаг 3** укажите параметры криптографии и нажмите **Создать сертификат**:



Скачайте сертификат после его создания:



7.2. Настройка службы Kerberos контроллера домена SambaAD

Перенесите полученный сертификат (`SRV101-VM203.p12`) и цепочку сертификатов (`Basealt eCA Test Sub.chain.pem`) на контроллер домена. Для корректной работы Kerberos переименуйте файл цепочки сертификатов (`Basealt eCA Test Sub.chain.pem`) так, чтобы не было пробелов (в примере `chain.pem`).

Создайте каталог для закрытого ключа контроллера домена:

```
$ su -
# mkdir -v /var/lib/samba/private/tls/secure
mkdir: создан каталог '/var/lib/samba/private/tls/secure'
```

Извлеките сертификат и ключи из контейнера `SRV101-VM203.p12` (пароль на контейнер указывался ранее в пункте Выдача сертификата контроллера домена) в специальные каталоги для ключей и сертификатов контроллера домена:

```
# openssl pkcs12 -in /var/tmp/SRV101-VM203.p12 \
    -out /var/lib/samba/private/tls/DC.crt.pem -clcerts -nokeys
Enter Import Password:
```

```
# openssl pkcs12 -in /var/tmp/SRV101-VM203.p12 \
    -out /var/lib/samba/private/tls/secure/DC.key.pem -nocerts -nodes
Enter Import Password:
```

В каталоге для сертификатов разместите файл цепочки сертификатов **ЦС**:

```
# cp -v /var/tmp/chain.pem /var/lib/samba/private/tls/
'/var/tmp/chain.pem' -> '/var/lib/samba/private/tls/chain.pem'
```

Приведите конфигурационный файл службы Kerberos (`/etc/krb5.conf`) к следующему виду:

```
[libdefaults]
default_realm = TEST5.ALT
dns_lookup_realm = false
dns_lookup_kdc = true
ticket_lifetime = 24h
forwardable = yes
pkinit_anchors = FILE:/var/lib/samba/private/tls/chain.pem

[appdefaults]
pkinit_anchors = FILE:/var/lib/samba/private/tls/chain.pem

[realms]
TEST5.ALT = {
pkinit_require_eku = true
}

[kdc]
enable-pkinit = yes
pkinit_identity =
FILE:/var/lib/samba/private/tls/DC.crt.pem,/var/lib/samba/private/tls/secure/DC.key.pem
pkinit_anchors = FILE:/var/lib/samba/private/tls/chain.pem
pkinit_principal_in_certificate = yes
pkinit_win2k = no
pkinit_win2k_require_binding = yes
```

Сертификат `DC.crt.pem` и ключ `DC.key.pem` в параметре `pkinit_identity` необходимо указывать в одной строке.

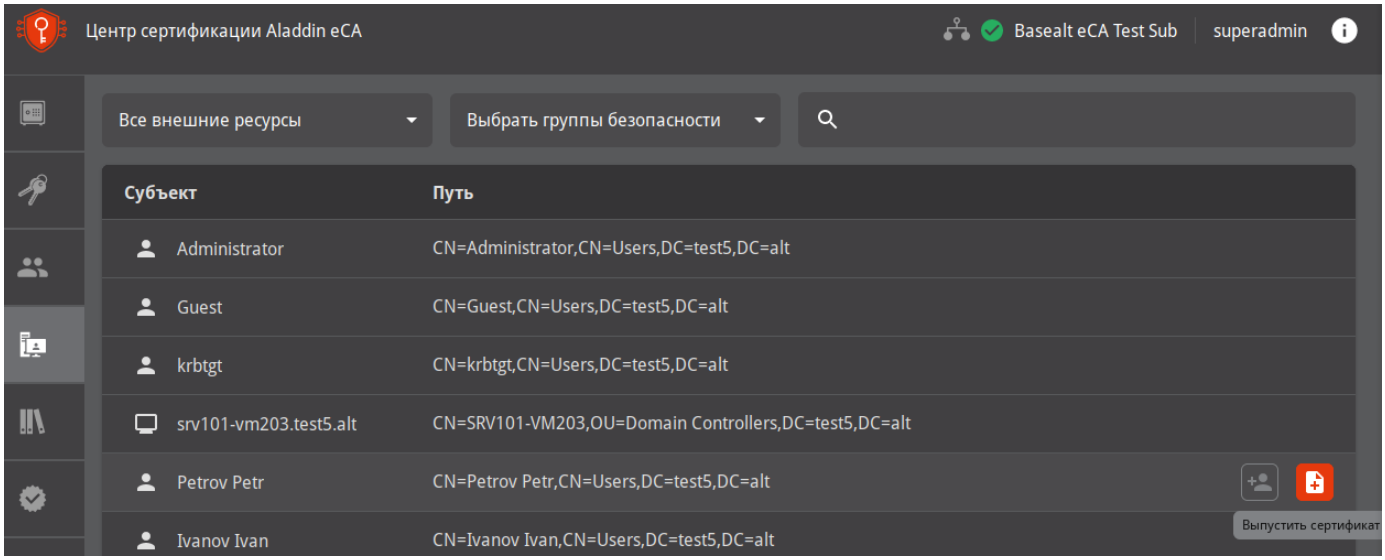
Перезапустите службу `samba`:

```
# systemctl restart samba.service
```

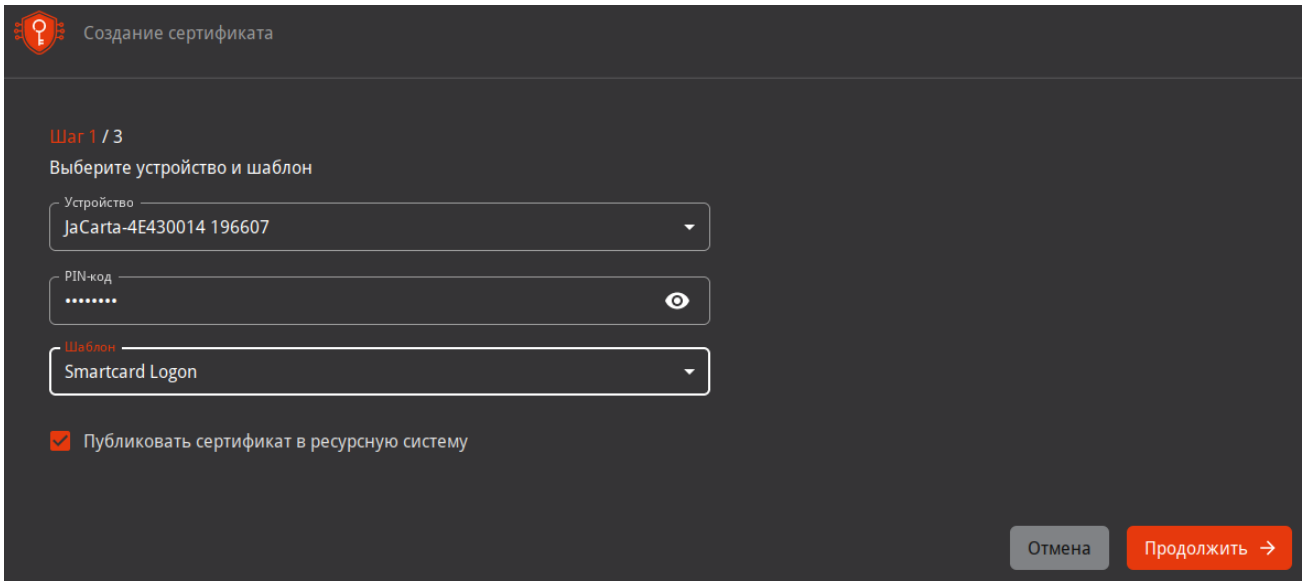
7.3. Выдача сертификата пользователя домена

Подключите токен к подчинённому **ЦС**. Если управление осуществляется на отдельном АРМ, то на нём должен быть установлен `JC-WebClient` (Установка JC-WebClient).

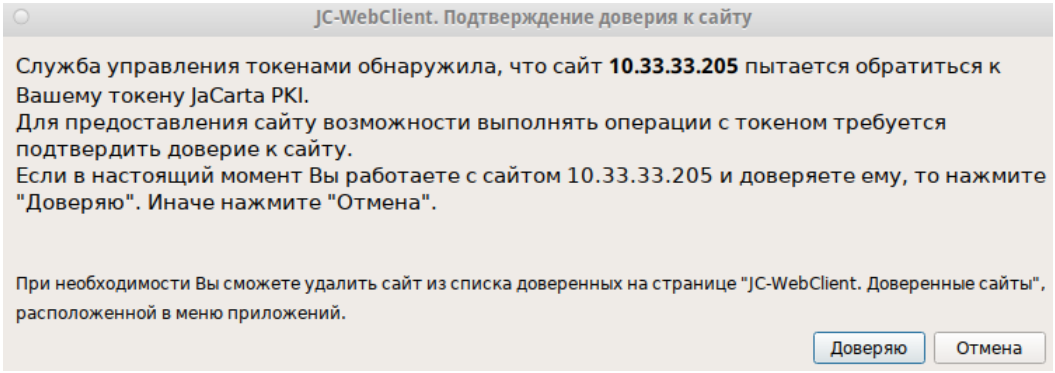
В веб-интерфейсе подчинённого **ЦС** перейдите в **Субъекты**, выберите пользователя домена и нажмите **Выпустить сертификат. На ключевом носителе**:



На **Шаг 1/3** выберите токен (слот токена) поддерживающий `RSA` или `ECDSA`, введите PIN-код и укажите шаблон сертификата (`Smartcard Logon`):

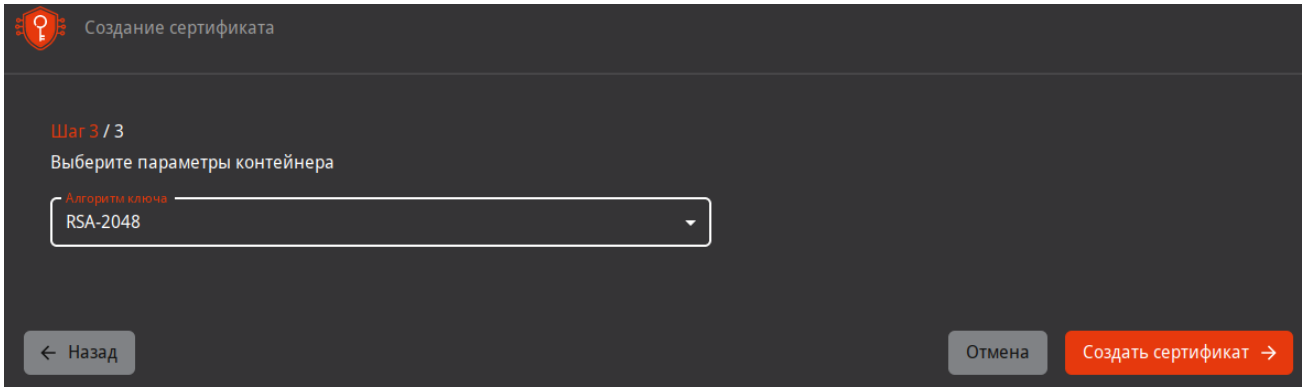


Подтвердите доверие для работы с токеном:

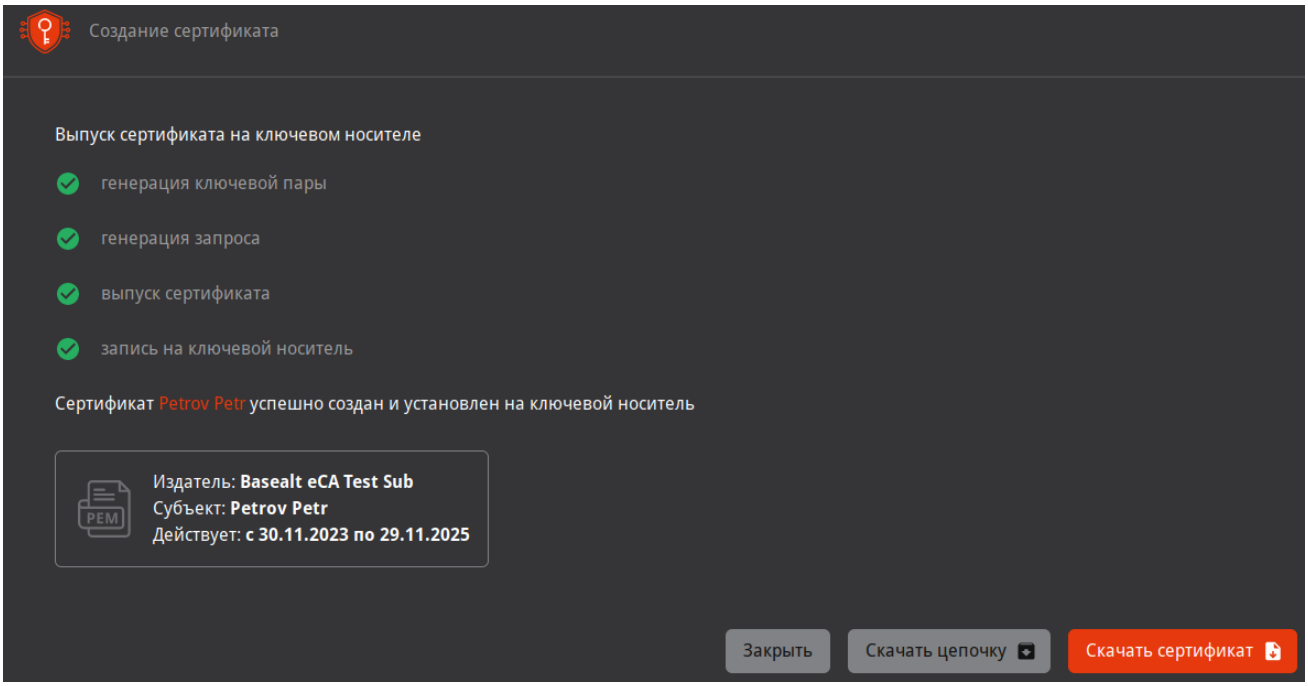


На **Шаг 2/3** укажите данные для шаблона сертификата `Smartcard Logon`:

На **Шаг 3/3** выберите параметры контейнера для ключей и сертификата пользователя и нажмите **Создать сертификат**:



При успешном выпуске сертификата должны получить следующее сообщение:



Теперь на токене находится ключевая пара и сертификат доменного пользователя.

8. Настройка АРМ пользователя домена

На данном этапе предполагается, что АРМ пользователя успешно введён в домен и осуществляется успешная аутентификация доменным пользователем с получением билета Kerberos. Проверить это можно руководствуясь следующим документом - https://docs.altlinux.org/ru-RU/alt-workstation/index.html (пункт **Ввод рабочей станции в домен Active Directory**).

8.1. Установка ПО SecurLogon

ПО SecurLogon отвечает за настройку компонентов АРМ пользователя (pam модули, greeter, библиотеки PKCS11, служба аутентификации sss, служба Kerberos и т.д.) для строгой двухфакторной аутентификации.

Для работы SecurLogon требуется предварительная установка Единый Клиент JaCarta (https://www.aladdin-rd.ru/support/downloads/jacarta_client/). Перейдите в каталог с вышеназванным распакованным ПО (в примере /var/tmp) и установите его.

Установка Единый Клиент JaCarta:

```
$ su -
# cd /var/tmp
# chmod +x jacartauc_3.0.0.3341_alt_x64/install.sh
# jacartauc_3.0.0.3341_alt_x64/install.sh
```

Если при установке вы сталкиваетесь с ошибкой "**файл /usr/lib64/libASEP11.so из устанавливаемого пакета jcpkcs11-2-⋯ конфликтует с файлом из пакета libjcpkcs11-⋯**", то удалите пакет libjcpkcs11 (apt-get remove libjcpkcs11) и произведите установку Единый Клиент JaCarta заново.

Установка SecurLogon:

```
# chmod +x SecurLogon_2.0.1.227_alt9_x64/install.sh
# SecurLogon_2.0.1.227_alt9_x64/install.sh
Проверка установки пакета jcPKCS11-2:
[Успех]
В процессе установки в системе должен присутствовать необходимый диск с дистрибутивом ОС в
CD-ROM или доступ к официальному репозиторию
Продолжить установку? [Д/н] Д
...
Обновление / установка...
1: jc_lightdm_greeter-2.0.1.227-1
########################################################################## [100%]
Завершено.
Для правильной работы SecurLogon, требуется перезагрузка
Выполнить перезагрузку сейчас? [Д/н] Д
Установка успешно завершена
```

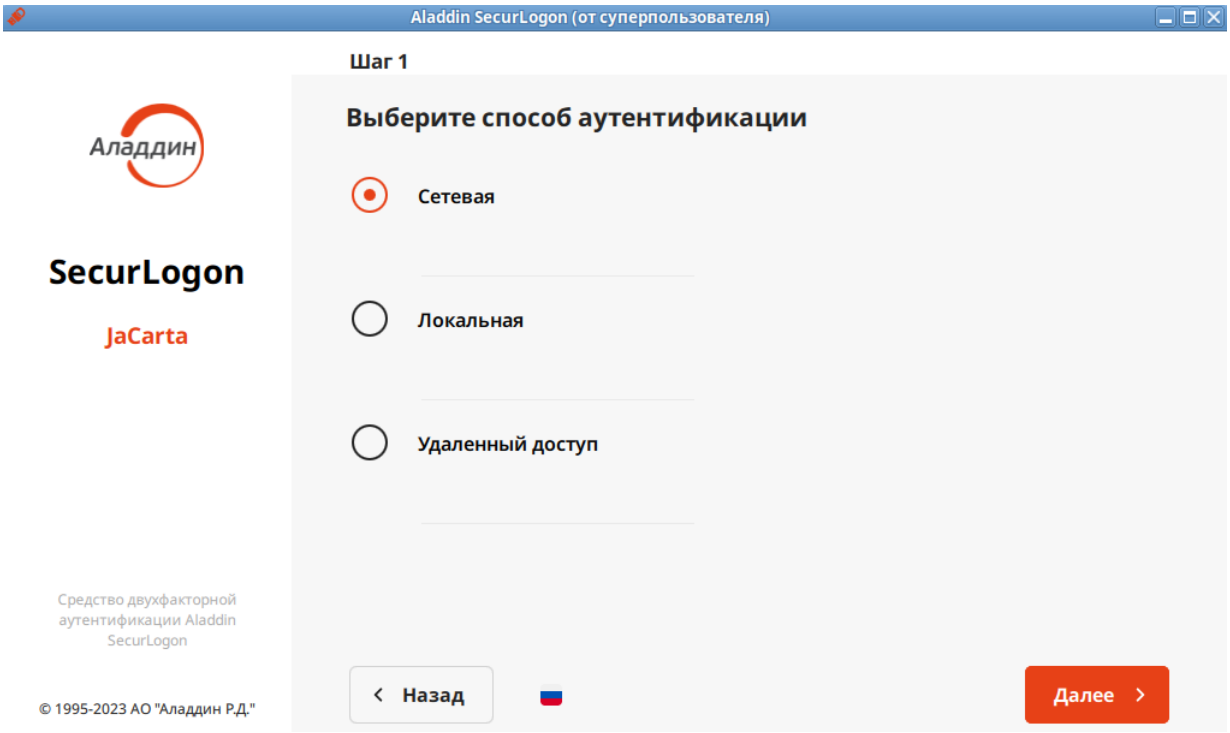
После установки `SecurLogon` необходимо перезагрузить ПК.

8.2. Настройка двухфакторной аутентификации

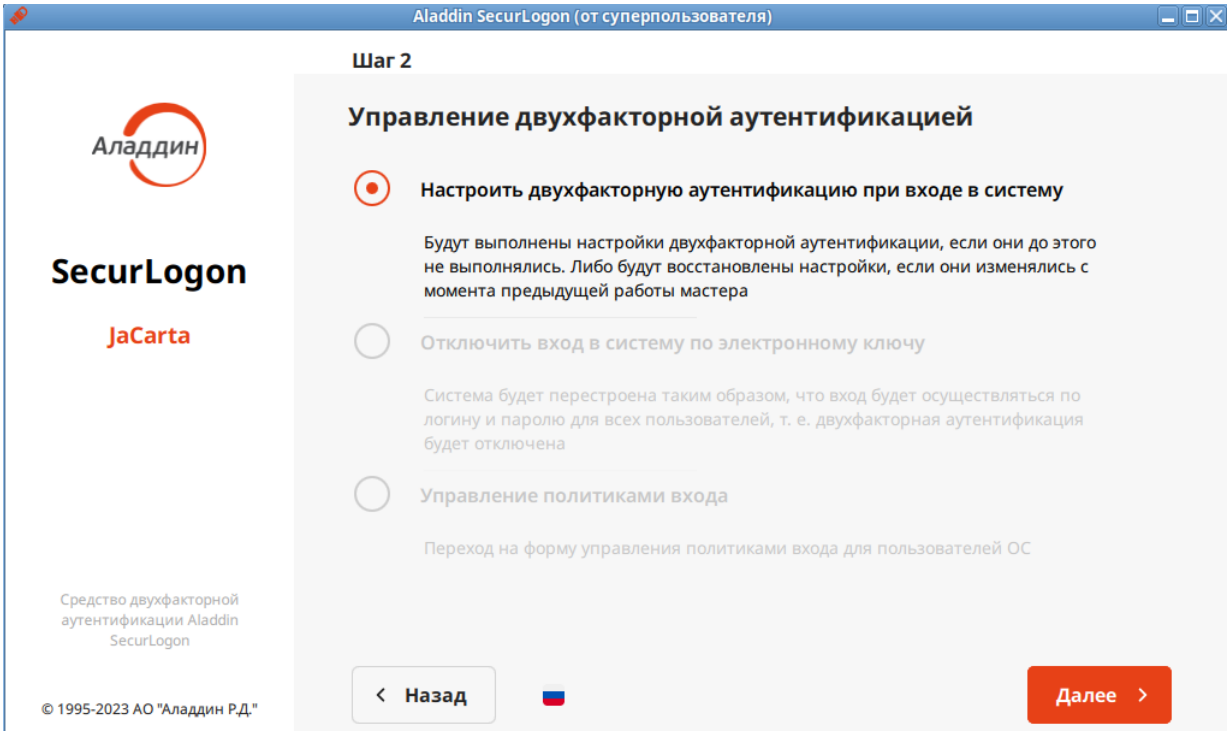
Подключите токен с сертификатом пользователя. Перенесите цепочку сертификатов **ЦС** (`chain.pem`) на АРМ пользователя.

Запустите ПО `SecurLogon` (**Меню — Стандартные — SecurLogon**).

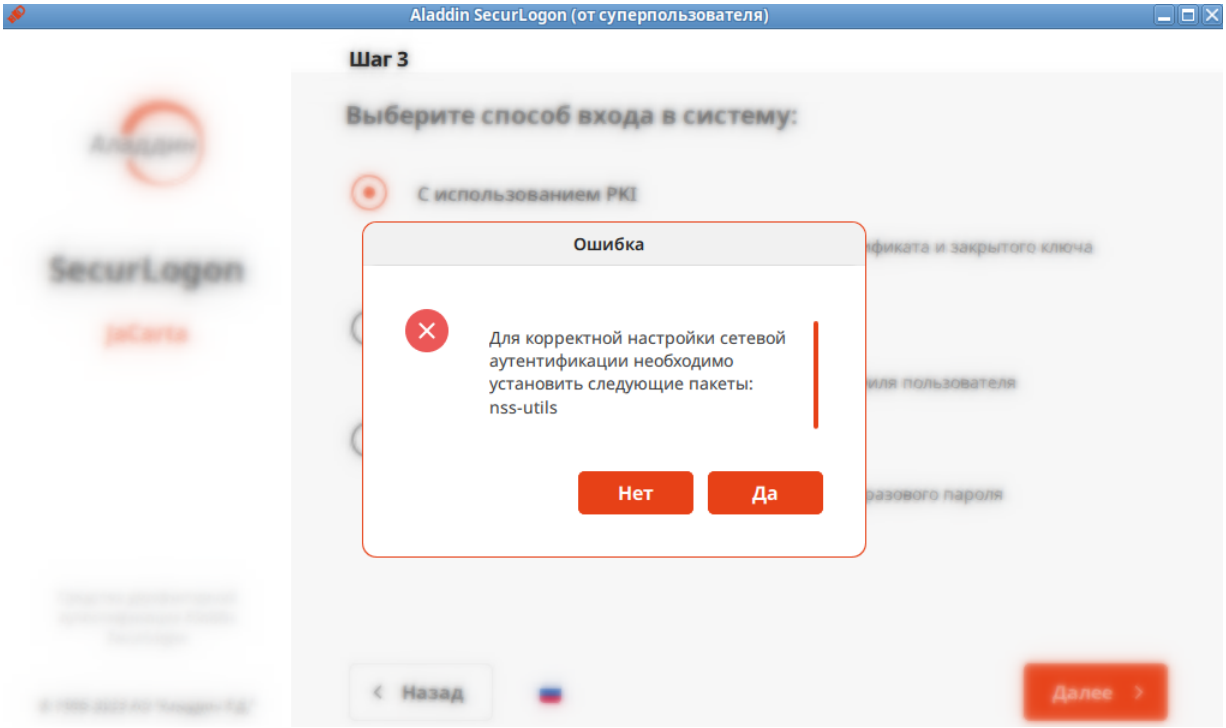
После ввода лицензии на **Шаг 1** выберите способ аутентификации **Сетевая**:



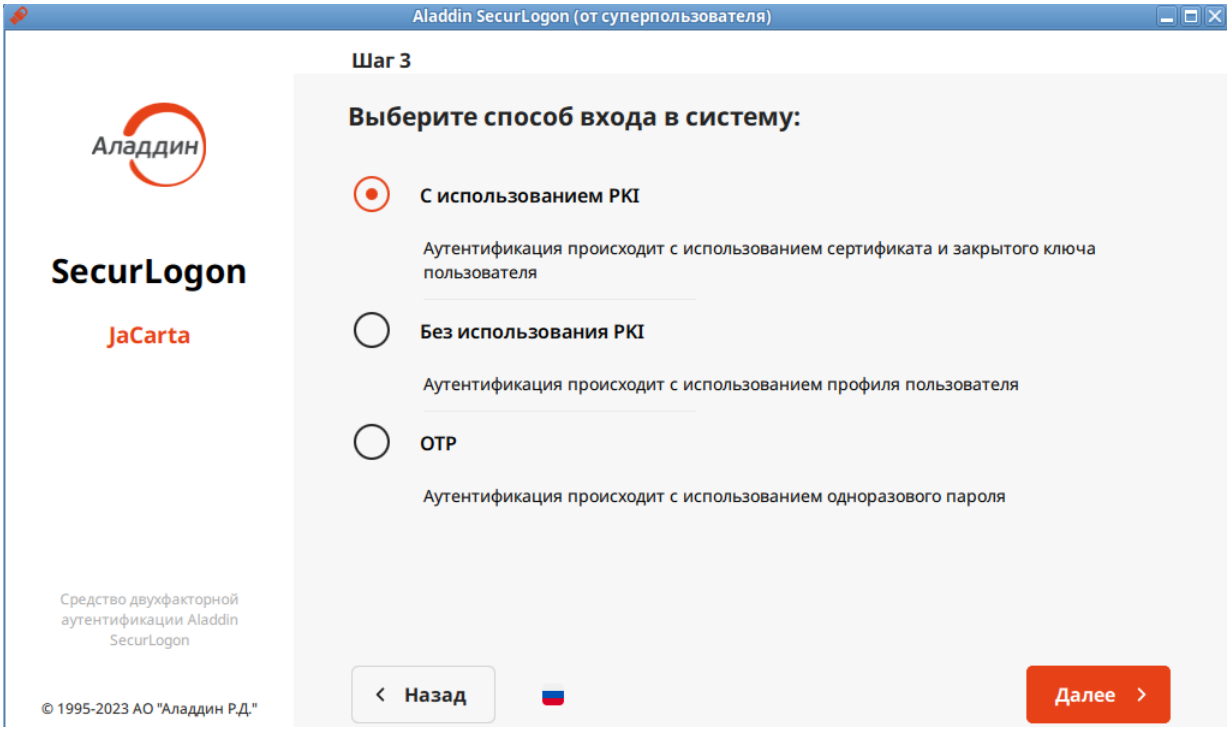
На **Шаг 2** выберите **Настроить двухфакторную аутентификацию при входе в систему**:



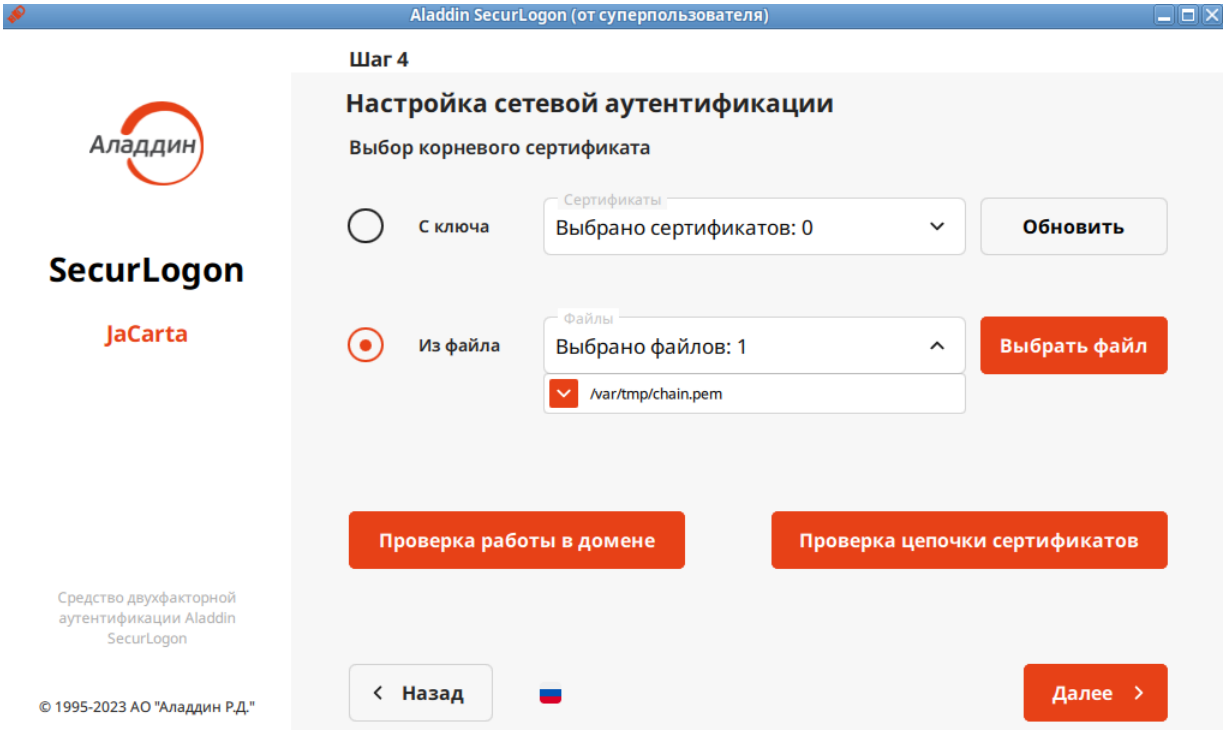
Установите дополнительные пакеты, если будет предложено:



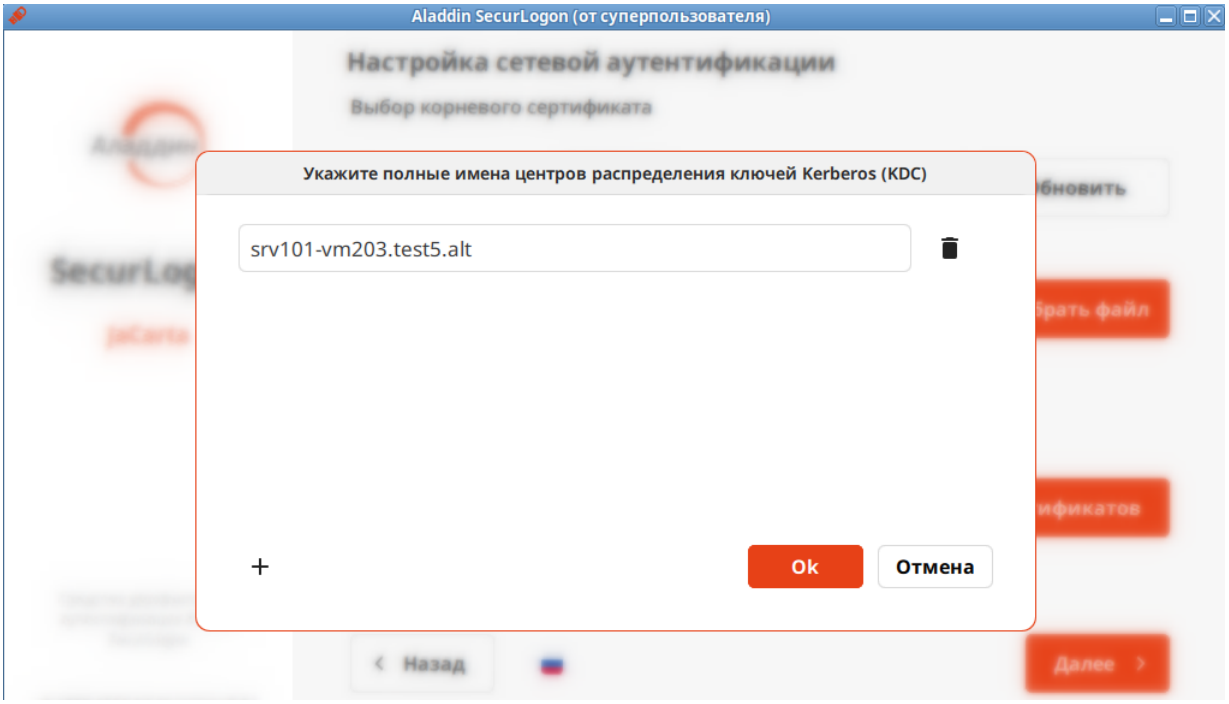
На **Шаг 3** выберите **С использованием PKI**:



На **Шаг 4** выберите файл цепочки сертификатов корневых центров `chain.pem`:

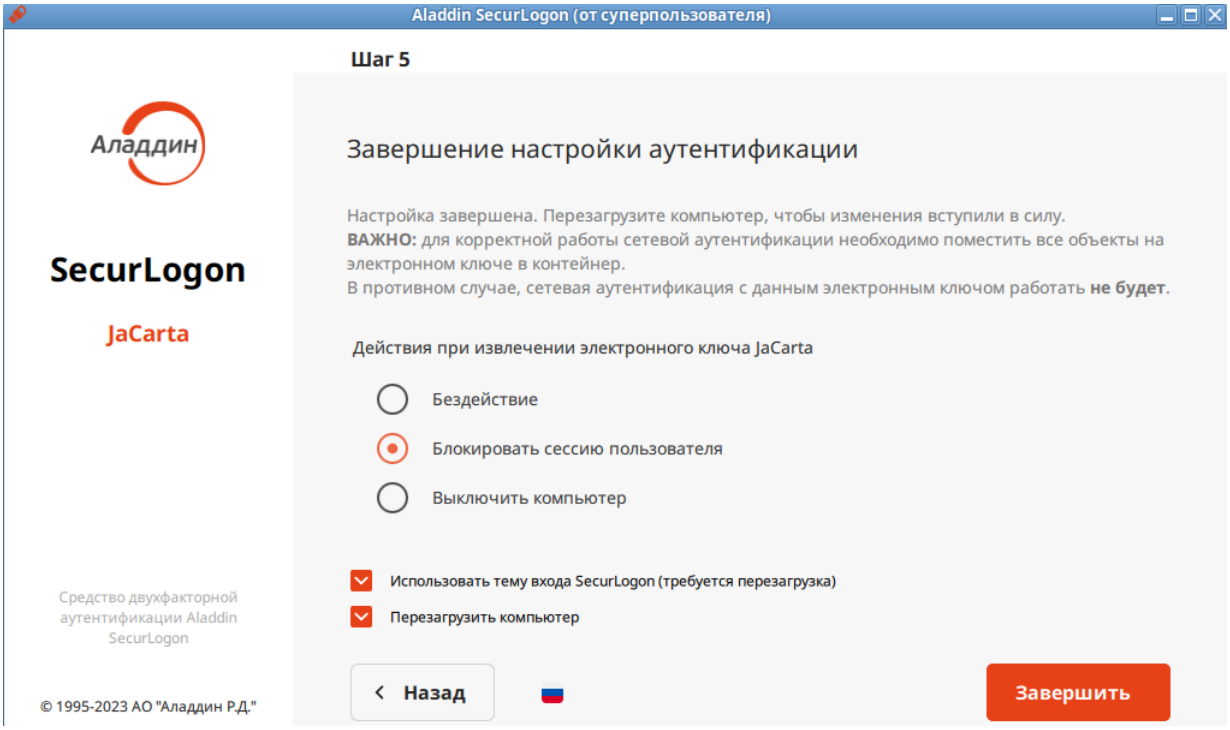


Далее укажите FQDN центра распределения ключей Kerberos (KDC):



На вопрос **«Установить дополнительно OTP аутентификацию?»** ответьте **Нет**.

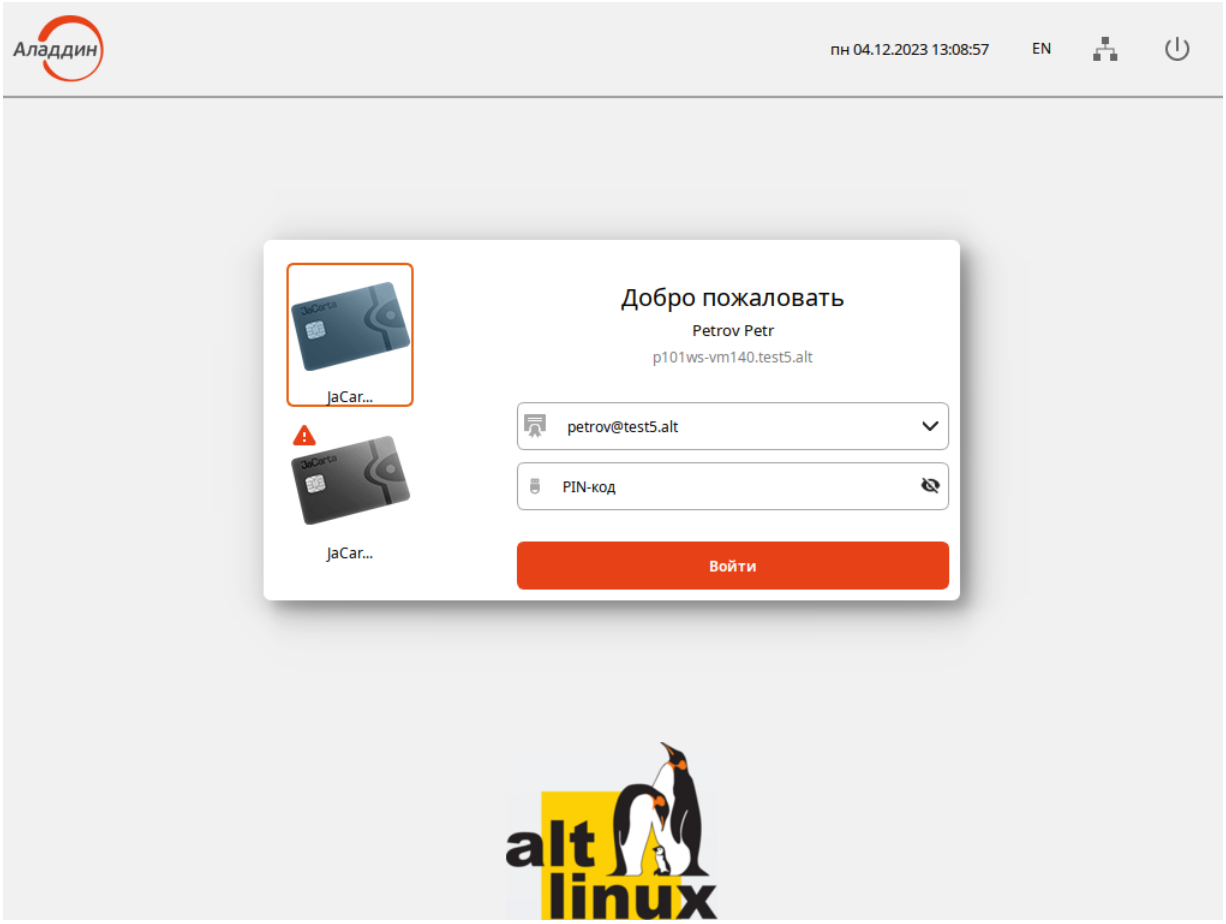
На **Шаг 5** выберите необходимые действия при извлечении электронного ключа:



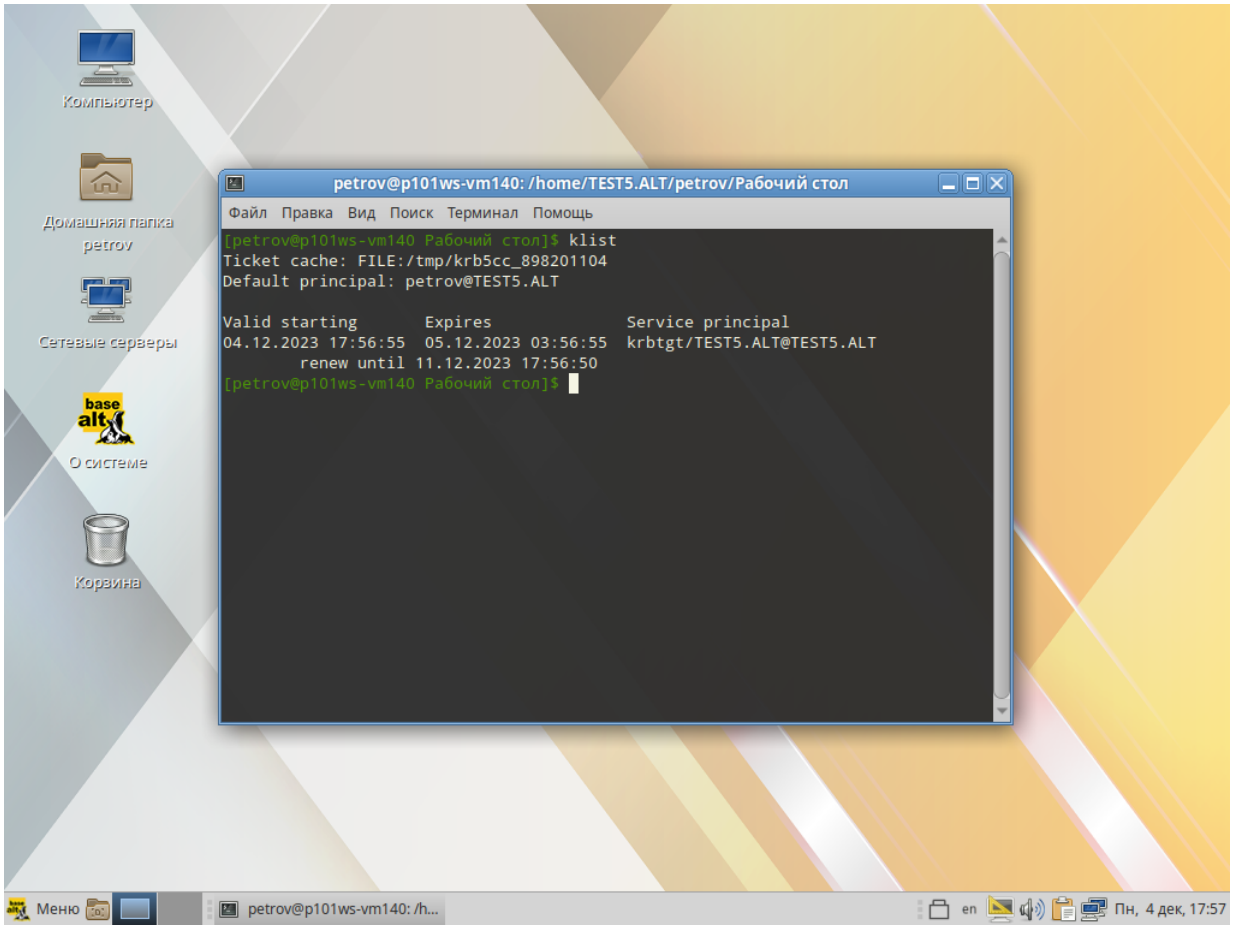
Перезагрузите компьютер.

8.3. Проверка двухфакторной аутентификации

После настройки ПО `SecurLogon` и перезагрузки ПК, на этапе логина в ОС необходимо выбрать слот токена (смарт-карты), на котором находится сертификат, и ввести PIN-код. Нужный слот будет без восклицательного знака и при этом будет автоматически заполнена учётная запись пользователя:



После успешного входа при помощи команды `klist` убедитесь, что автоматически получен билет Kerberos:



9. Удаление AeCA

Процедура удаления для **ЦС** и **ЦВ** идентична.

Для **ЦВ**:

```
$ su -
# bash /opt/aecaVa/scripts/uninstall.sh
================================================
LAST CHANCE TO STOP THIS
You want to fully remove AECA and it's dependencies
Are you sure you want to continue?
1) Yes
2) No
#? 1
```

Для **ЦС**:

```
$ su -
# bash /opt/aecaCa/scripts/uninstall.sh
================================================
LAST CHANCE TO STOP THIS
You want to fully remove AECA and it's dependencies
Are you sure you want to continue?
1) Yes
2) No
#? 1
```

Затем удаляем пакет `aeca` (для **ЦС** и **ЦВ**):

```
# apt-get remove aeca
```

Контакты службы обеспечения совместимости

Электронная почта:

gost@basealt.ru

Телефоны для оперативной связи:

+7 (495) 123-47-99, доб. 558
+7 (812) 66-789-33

Служба всегда стремится дать ответ в течение 48 часов. Если на третий рабочий день ответ не будет получен, повторите свой запрос.