



# Aladdin Enterprise CA 2.1

Служба обеспечения совместимости <[gost@basealt.ru](mailto:gost@basealt.ru)>, Константин Белаш

Version 1.0, 11.02.2025

# Оглавление

1. Дистрибутив ОС Альт	1
2. Обновление ОС до актуального состояния	2
3. Подготовка ПО для работы с токенами	3
3.1. Удаление конфликтующего ПО	3
3.2. Установка ПО	3
4. Установка AeCA	4
4.1. Установка вспомогательных пакетов	4
4.2. Настройка <code>sudo</code>	4
4.3. Установка OpenJDK	4
4.3.1. Axiom JDK	5
4.4. Установка СУБД PostgreSQL	5
4.5. Настройка СУБД PostgreSQL	6
4.5.1. ЦС	6
4.5.2. ЦВ	7
4.6. Установка JC-WebClient	7
4.7. Установка основных пакетов AeCA	8
4.7.1. ЦС (корневой и подчинённый)	8
4.7.2. ЦВ	8
5. Настройка основных компонентов AeCA	9
5.1. Развертывание AeCA	9
5.1.1. ЦС (корневой и подчинённый)	9
5.1.1.1. Настройка веб-сервера Apache	9
5.1.1.2. Настройка веб-сервера Nginx	10
5.1.1.3. Выполнение скрипта развертывания	10
5.1.2. ЦВ	11
5.2. Установка и настройка КриптоПро CSP 5	13
5.3. Настройка доступа к веб-интерфейсу AeCA	15
5.4. Добавление лицензии и активация корневого и подчинённого ЦС	17
5.5. Регистрация ЦВ и активация службы OCSP	23
6. Подключение ресурсной системы	26
7. Обеспечение возможности строгой аутентификации пользователей в домене	30
7.1. Выдача сертификата контроллера домена	30
7.2. Настройка службы Kerberos контроллера домена Samba DC	32
7.3. Выдача сертификата пользователя домена	34
8. Настройка АРМ пользователя домена	38
8.1. Установка ПО SecurLogon	38
8.2. Настройка двухфакторной аутентификации	39
8.3. Проверка двухфакторной аутентификации	42



# 1. Дистрибутив ОС АЛЬТ

В данной инструкции используются следующие ОС:

- **Альт Сервер 10.2** (репозиторий p10) — Центр сертификации (**ЦС**, корневой и подчинённый), Центр валидации (**ЦВ**), контроллер домена **Samba DC**;
- **Альт Рабочая станция 10.2** (репозиторий p10) — ПК в домене **Samba DC**.

С полным списком поддерживаемых дистрибутивов семейства ОС АЛЬТ можно ознакомиться в документации **Aladdin Enterprise CA (АЕСА)** — <https://www.aladdin-rd.ru/catalog/aladdin-eca/#documentation/>.

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

**АЕСА** 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 **Samba DC** и его пользователей.

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

## 2. Обновление ОС до актуального состояния

Процедуры установки, обновления и удаления AeSA 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 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 psc-lite-openct
```

### 3.2. Установка ПО

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

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

- `libjcpkcs11` — библиотеки `PKCS#11` вендора токенов Аладдин;
- `opensc`, `psc-lite-ccid`, `psc-lite`, `psc-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/](https://www.aladdin-rd.ru/support/downloads/jacarta_client/) — Аладдин JaCarta (`libjcpkcs11-2`) (теперь ещё и в составе «Единого Клиента JaCarta»).

## 4. Установка AeSA

Если компоненты AeSA не выделены в отдельный подпункт, то необходимо считать, что настройки для них одинаковы.

### 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 AeSA используют это приложение. Разрешите членам группы wheel использовать sudo.

```
# sed -i -E 's/# (User_Alias\s*WHEEL_USERS)/\1/' /etc/sudoers
# sed -i -E 's/# (WHEEL_USERS ALL=\(ALL:ALL\) ALL)/\1/' /etc/sudoers
```

### 4.3. Установка OpenJDK

Согласно документации, для ЦС необходимо установить OpenJDK 17. Для ЦВ необходимо установить OpenJDK 11.

Определите установленную версию Java:

```
# java -version
openjdk version "17.0.12" 2024-07-16
OpenJDK Runtime Environment (Red_Hat-17.0.12.0.7-alt1) (build 17.0.12+7)
OpenJDK 64-Bit Server VM (Red_Hat-17.0.12.0.7-alt1) (build 17.0.12+7, mixed mode, sharing)
```

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

Если версия OpenJDK отличается от нужной, то удалите её и установите необходимую (пример для ЦВ, где необходима версия 11):

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

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

Убедитесь, что установлена и будет использоваться OpenJDK версии 11:

```
# java -version
```

```
openjdk version "11.0.24.1" 2024-07-16
OpenJDK Runtime Environment 18.9 (build 11.0.24.1+1)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.24.1+1, mixed mode, sharing)
```

### 4.3.1. Axiom JDK

При использовании **Axiom JDK** необходимо дополнительно указать параметр **JAVA\_HOME** в конфигурационном файле `/etc/java/java.conf`. Содержимое параметра **JAVA\_HOME** (путь к используемой версии **JDK**) можно получить при помощи следующей команды:

```
# dirname $(dirname $(readlink -f $(which java)))
```

## 4.4. Установка СУБД PostgreSQL

Установите последнюю доступную версию **СУБД PostgreSQL** из репозитория. Список версий:

```
# apt-get install postgresql-server
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Виртуальный пакет postgresql-server предоставляется следующими пакетами:
 postgresql16-server p10+340531.100.1.1@1707759757
 postgresql15-1C-server p10+340531.600.1.1@1707761177
 postgresql15-server p10+340531.500.1.1@1707760888
 postgresql14-1C-server p10+315989.200.3.1@1678642617
 postgresql14-server p10+340531.400.1.1@1707760600
 postgresql13-server p10+340531.300.1.1@1707760316
 postgresql12-server p10+340531.200.1.1@1707760041
 postgresql11-server p10+333972.300.4.1@1699932811
 postgresql10-server p10+333972.200.4.1@1699932493
Необходимо точно указать, какой из пакетов должен быть установлен.
E: Виртуальный пакет postgresql-server предоставляется многими пакетами.
```

```
# apt-get install postgresql16-server
# apt-get install postgresql16-contrib
```

```
# /etc/init.d/postgresql initdb
...
Введите новый пароль суперпользователя:
```

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

```
# sed -i -E 's/(host.*all.*all.*12[78].*\s)\S+$/\1password/g' \
/var/lib/pgsql/data/pg_hba.conf
```



```
# sed -i -E 's/(local.*all.*all.*\s)\S+$/\1trust/g' \
/var/lib/pgsql/data/pg_hba.conf
```

Убедитесь, чтобы следующие строки файла `/var/lib/pgsql/data/pg_hba.conf` выглядели следующим образом:

```
# grep -E 'all.*all.*(password|trust)' /var/lib/pgsql/data/pg_hba.conf
local  all          all                      trust
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

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

### 4.5.1. ЦС

```
# psql -U postgres
psql (16.6)
Введите "help", чтобы получить справку.

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

Перезапустите `postgresql`:

```
# systemctl restart postgresql
```

Установите расширение `pgcrypto`:

```
# psql -U postgres -d аеса \
-c "CREATE EXTENSION IF NOT EXISTS pgcrypto WITH SCHEMA pg_catalog;"
```



Возможно появление ошибки — **ОШИБКА: функция "gen\_random\_uuid" с аргументами таких типов уже существует**. На работу БД не влияет. Говорит о том, что данная функция уже реализована в схеме `pg_catalog`. Продолжайте установку.

## 4.5.2. ЦВ

```
# psql -U postgres
psql (16.6)
Введите "help", чтобы получить справку.

postgres=# CREATE USER аеса;
CREATE ROLE
postgres=# ALTER USER аеса WITH PASSWORD 'аеса';
ALTER ROLE
postgres=# CREATE DATABASE аеatest;
CREATE DATABASE
postgres=# ALTER DATABASE аеatest OWNER TO аеса;
ALTER DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE аеatest TO аеса;
GRANT
postgres=# ALTER USER аеса SUPERUSER;
ALTER ROLE
postgres=# \q
```

Перезапустите `postgresql`:

```
# systemctl restart postgresql
```

## 4.6. Установка JC-WebClient

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



Если **Альт Сервер 10.2** установлен в минимальной конфигурации (без поддержки работы в графическом сеансе), то `JC-WebClient` необходимо установить на ПК, который будет выступать в роли администратора **АеСА**.

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.

### 4.7.1. ЦС (корневой и подчинённый)

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_*.rpm
```

### 4.7.2. ЦВ

Для установки ЦВ выполните следующую команду:

```
# cd /var/tmp
# apt-get install ./aeca-va_*.rpm
```

# 5. Настройка основных компонентов AeCA

## 5.1. Развертывание AeCA

### 5.1.1. ЦС (корневой и подчинённый)

В конфигурационном файле AeCA ЦС переопределите порт ЦВ (параметр `aeca_cdp_port`), чтобы избежать конфликта с веб-интерфейсом Альтератора (использует порт `8080`):

```
# sed -i -E "s/(aeca_cdp_port=).*\/\1'8085'/" /opt/aecaCa/scripts/config.sh
```

Измените пароль пользователя базы данных, если он не совпадает с введённым ранее:

```
# sed -i -E "s/(database_password=).*\/\1'aeca'/" /opt/aecaCa/scripts/config.sh
```

Выберите используемый веб-сервер. Возможные варианты - `nginx` или `apache`. Выбрать веб-сервер можно также интерактивно, при выполнении скрипта развёртывания (далее).

При использовании `apache` отредактируйте конфигурационный файл следующим образом:

```
# sed -i -E "s/(webservice=).*\/\1'apache'/" /opt/aecaCa/scripts/config.sh
# sed -i -E "s:(webservice_path=).*:\1'/etc/httpd2':" /opt/aecaCa/scripts/config.sh
```

При использовании `nginx` отредактируйте конфигурационный файл следующим образом:

```
# sed -i -E "s/(webservice=).*\/\1'nginx'/" /opt/aecaCa/scripts/config.sh
# sed -i -E "s:(webservice_path=).*:\1'/etc/nginx':" /opt/aecaCa/scripts/config.sh
```

#### 5.1.1.1. Настройка веб-сервера Apache

В случае выбора веб-сервера `Apache` выполните следующие действия.

Установите необходимый модуль для `Apache2`:

```
# apt-get install apache2-mod_http2 apache2-mod_ssl
```

Создайте конфигурационный файл `http2.load`:

```
# echo "LoadModule http2_module /usr/lib64/apache2/modules/mod_http2.so" > \
/etc/httpd2/conf/mods-available/http2.load
```

Создайте конфигурационный файл `http2.conf`:

```
# cat > /etc/httpd2/conf/mods-available/http2.conf <<-EOF
<IfModule !mpm_prefork>
    Protocols h2 h2c http/1.1
</IfModule>
EOF
```

Активируйте модули **Apache2**:

```
# a2enmod ssl
# a2enmod proxy
# a2enmod proxy_http
# a2enmod headers
# a2enmod cgi
# a2enmod rewrite
# a2enmod http2
```

Включите порт **https**:

```
# a2enport https
```

Перезапустите службу **httpd2**:

```
# systemctl condreload httpd2.service
```

#### 5.1.1.2. Настройка веб-сервера Nginx

В случае выбора веб-сервера **Nginx** выполните следующие действия.

```
# apt-get install nginx
```

```
# systemctl enable --now nginx.service
```

Остановите службу **httpd2**, которая занимает порт **443** (используется для публикации справочных материалов).

```
# systemctl disable --now httpd2.service
```

#### 5.1.1.3. Выполнение скрипта развертывания

Запустите следующий скрипт (на примере **nginx**):

```
# bash /opt/aecaCa/scripts/install.sh
...
[AECA - patch] Создание каталогов...
[AECA - patch] Установка прав на /opt/aecaCa/services для аеса...
[AECA - patch] Установка прав на /opt/aecaCa/static для аеса...
...
[WEBSERVER - setup] Выберите используемый веб-сервер
1) apache
2) nginx
#? 2
...
[WEBSERVER - setup] Введите расположение конфигурации: /etc/nginx
...
AECA-CA : [#####] 100.00%
[SUCCESS] AECA-CA успешно запущен.
Установка завершена.
```

### 5.1.2. ЦВ

В конфигурационном файле `/opt/aecaVa/scripts/config.sh` задайте значение параметра `aeca_https_server_hostname`.

```
# sed -i -r 's/(aeca_https_server_hostname=").*"/\1srv-va\2/' \
/opt/aecaVa/scripts/config.sh

# grep hostname /opt/aecaVa/scripts/config.sh
aeca_https_server_hostname="srv-va"
```

Отредактируйте следующие параметры, чтобы изменить порт для **CDP** (CRL distribution point) и **OCSP**, и тем самым избежать конфликта с веб-интерфейсом **Альтератора** (использует порт **8080**).

Для изменения порта подключения (на **8085**) к точке распространения **ЦВ** откорректируйте следующие файлы:

```
# sed -i -r 's/(SERVER_PORT=).*"/\1"8085"/' /opt/aecaVa/env/cdp.env

# grep SERVER_PORT /opt/aecaVa/env/cdp.env
SERVER_PORT="8085"
```

```
# sed -i -r 's_(socket-binding=http:.*)"8080_\1"8085_' \
/opt/aecaVa/scripts/auxiliary_wildfly_setup.sh

# grep 'http:add(port=' /opt/aecaVa/scripts/auxiliary_wildfly_setup.sh
wildfly_add "/socket-binding-group=standard-sockets/socket-
binding=http:add(port="\8085\", interface="\http\)"
```

```
# sed -i -r 's/#(httpserver.pubhttp=)8080/\18085/' \
/opt/aecaVa/properties/web.properties

# grep -i 'pubhttp=' /opt/aecaVa/properties/web.properties
httpserver.pubhttp=8085
```

Запустите следующий скрипт развертывания ЦВ:

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

Далее в процессе развертывания необходимо ответить на вопросы установщика.

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 **АеСА**, 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
```

## 5.2. Установка и настройка **КриптоПро CSP 5**

Установите **КриптоПро CSP 5** согласно инструкции, размещённой на следующей странице [https://www.basealt.ru/fileadmin/user\\_upload/compatibility/P10-view2.html?s=КриптоПро%20CSP](https://www.basealt.ru/fileadmin/user_upload/compatibility/P10-view2.html?s=КриптоПро%20CSP). В данном примере **КриптоПро CSP 5** устанавливается на подчинённый ЦС.

```
zsh 0 $ rsync -avz /mnt/s1000/Temp/cpro_dist/cpro_13300_r4_amd64/linux-amd64
user@10.33.11.2:/var/tmp
```

Создайте каталог `/opt/aecaCa/services/cryptoproviders`, при его отсутствии.

```
# mkdir -p /opt/aecaCa/services/cryptoproviders
```

Из архива **КриптоПро CSP для Java**, соответствующего версии установленной **КриптоПро CSP 5**, скопируйте следующие файлы:

```
# cp -v {ASN1P.jar,asn1rt.jar,JCP.jar,JCSP.jar} /opt/aecaCa/services/cryptoproviders/
```



```
'ASN1P.jar' -> '/opt/aecaCa/services/cryptoproviders/ASN1P.jar'  
'asn1rt.jar' -> '/opt/aecaCa/services/cryptoproviders/asn1rt.jar'  
'JCP.jar' -> '/opt/aecaCa/services/cryptoproviders/JCP.jar'  
'JCSP.jar' -> '/opt/aecaCa/services/cryptoproviders/JCSP.jar'
```

Назначьте владельцем данных файлов и предоставьте права пользователю `aeca`:

```
# chown aeca:aeca \  
  /opt/aecaCa/services/cryptoproviders/{ASN1P.jar,asn1rt.jar,JCP.jar,JCSP.jar}
```

```
# chmod 700 \  
  /opt/aecaCa/services/cryptoproviders/{ASN1P.jar,asn1rt.jar,JCP.jar,JCSP.jar}
```

Подготовьте внешнюю гамму (набор случайных данных), в случае её отсутствия

```
# mkdir -p ~/gamma
```

```
# /opt/cproscsp/bin/*/genkprim 10 0x12345678 ~/gamma
```

`10` — количество ключей, на которое создаётся гамма

Скопируйте гамму в AeCA

```
# cp -a ~/gamma/. /opt/aecaCa/dist/gamma
```

Назначьте владельцем данных файлов и предоставьте права пользователю `aeca`

```
# chown -R aeca:aeca /opt/aecaCa/dist/gamma  
# chmod -R 700 /opt/aecaCa/dist/gamma
```

Подключите внешнюю гамму

```
# /opt/cproscsp/sbin/*/cpconfig -hardware rndm -add cpsd \  
  -name 'cpsd rng' -level 3
```

```
# /opt/cproscsp/sbin/*/cpconfig -hardware rndm -configure cpsd \  
  -add string /db1/kis_1 /opt/aecaCa/dist/gamma/db1/kis_1
```

```
# /opt/cproscsp/sbin/*/cpconfig -hardware rndm -configure cpsd \  
  -add string /db2/kis_1 /opt/aecaCa/dist/gamma/db2/kis_1
```

Перезапустить службу `aeca-ca.service`

```
# systemctl restart aeca-ca.service
```

## 5.3. Настройка доступа к веб-интерфейсу AeCA

В процессе развёртывания на каждом сервере ЦС и ЦВ создаётся контейнер `p12` с сертификатом для доступа к веб-интерфейсу управления AeCA.

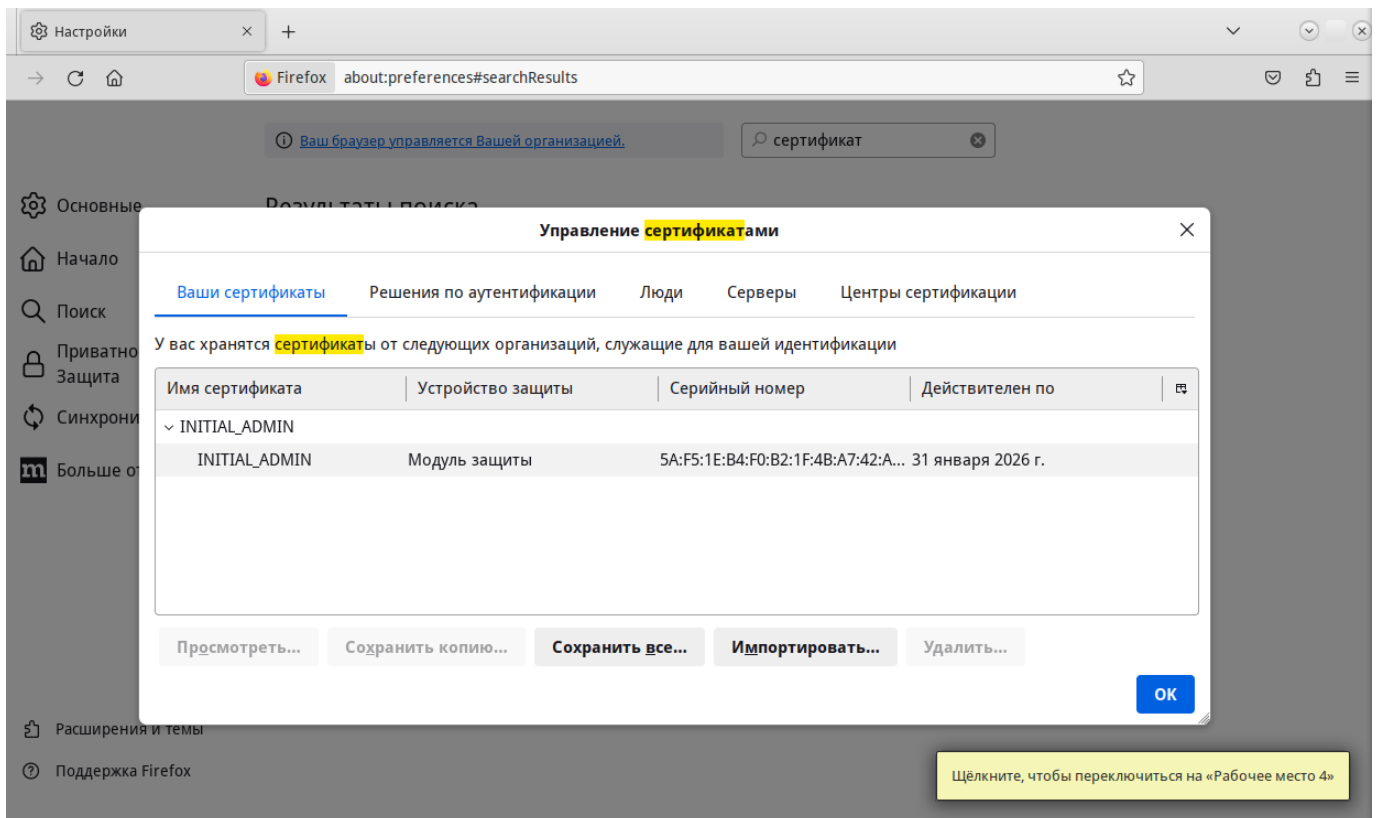
Для ЦС (корневого и подчинённого) — `/opt/aecaCa/dist/certificates/account/INITIAL_ADMIN.p12`. Пароль доступа к контейнеру по умолчанию — `INITIAL` (указан в конфигурационном файле `/opt/aecaCa/scripts/config.sh`).

Для ЦВ — `/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**

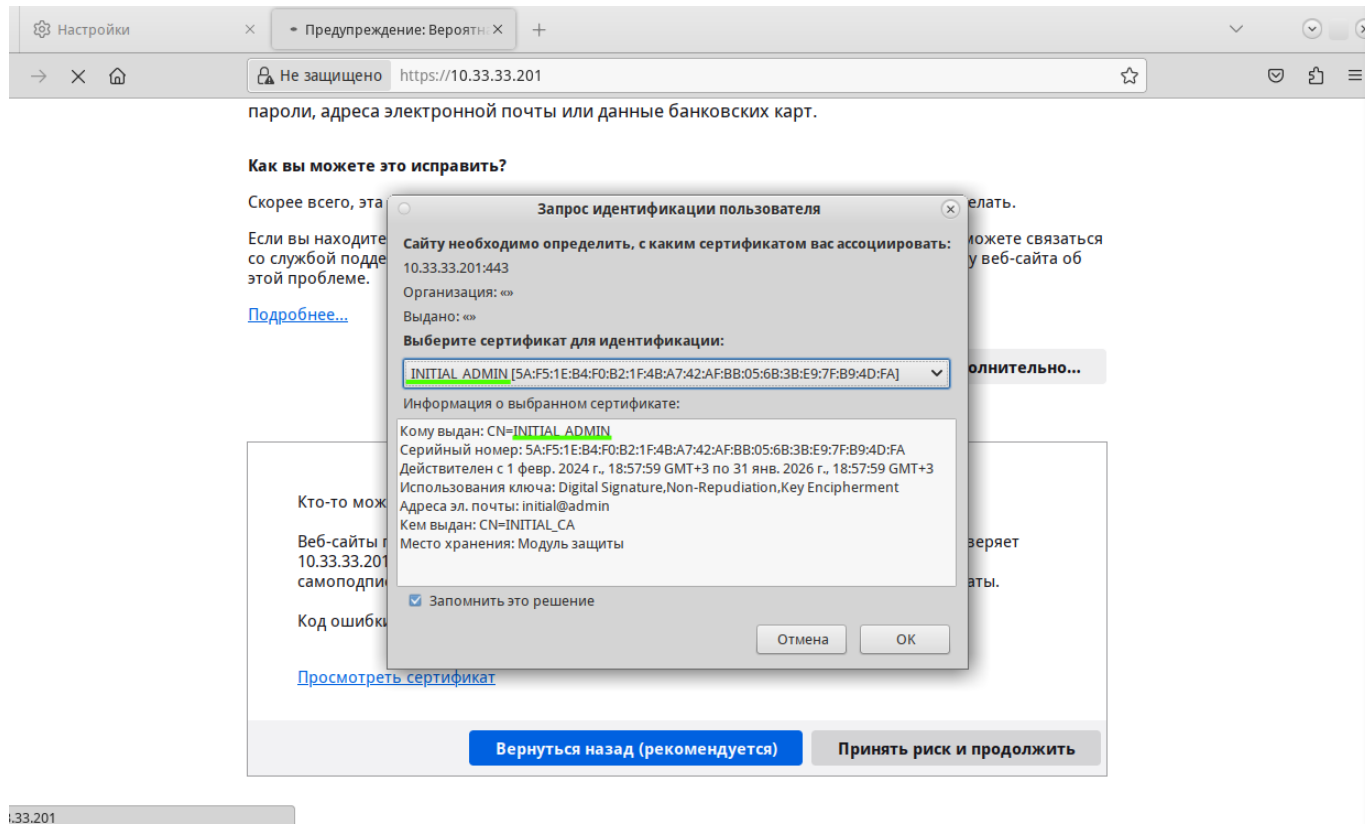
Выберите файл сертификата `p12`, созданный на этапе установки AeCA.



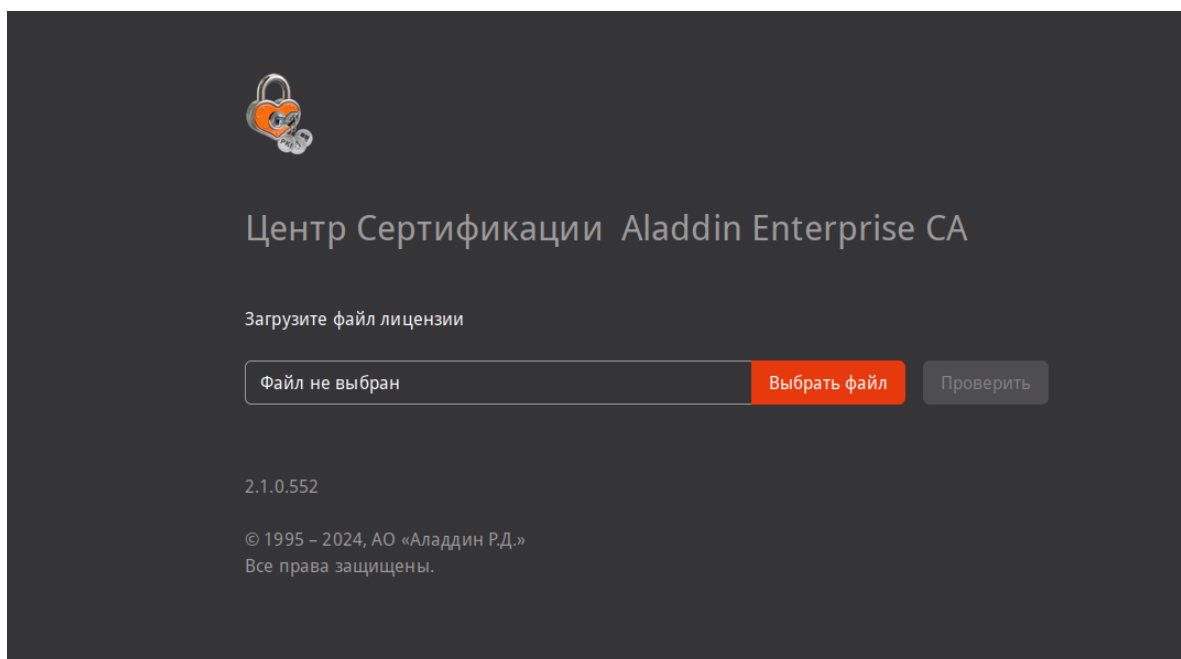
Для ЦС: перейдите по адресу [https://ip\\_address](https://ip_address)

Для ЦВ: перейдите по адресу [https://ip\\_address:8888/aecaVa/](https://ip_address:8888/aecaVa/)

При появлении окна **Предупреждение: Вероятная угроза безопасности** выберите **Дополнительно**, затем **Принять риск и продолжить**. Затем выберите сертификат **INITIAL\_ADMIN** для ЦС и **InitialAdmin** для ЦВ.



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



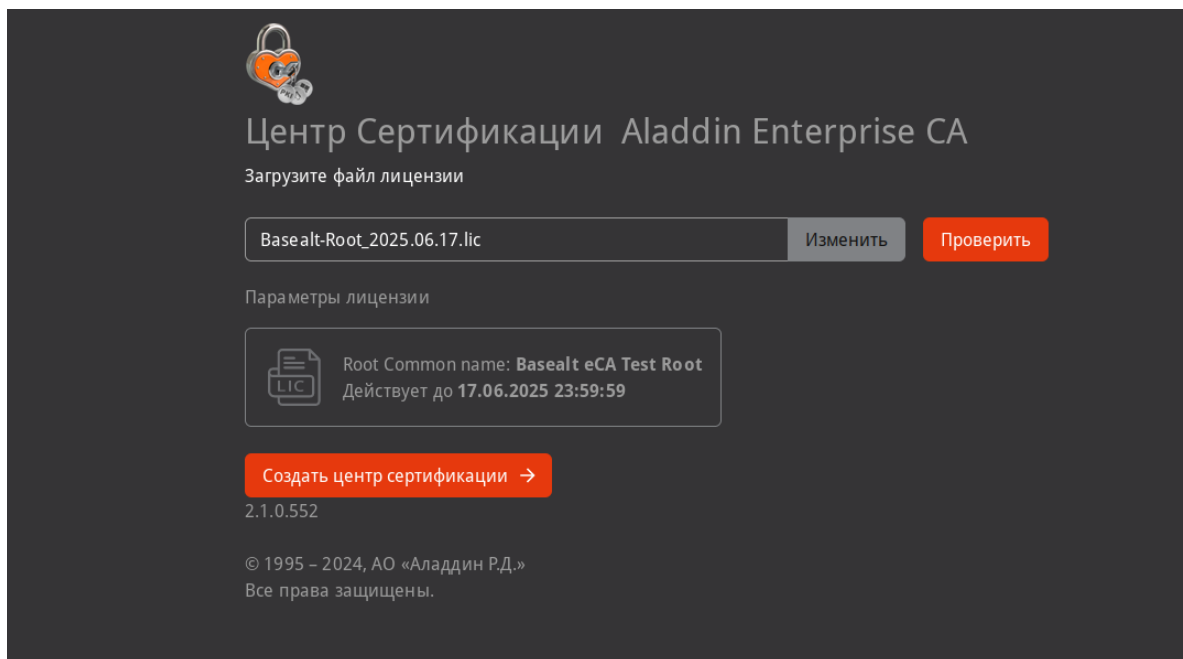
Если на сервере ЦС или ЦВ отсутствует графический интерфейс, то получить доступ можно с любого АРМ с графическим интерфейсом, предварительно добавив в браузер сертификаты из

контейнеров **p12** соответствующих серверов и установив **JC-WebClient** ([Установка JC-WebClient](#)).

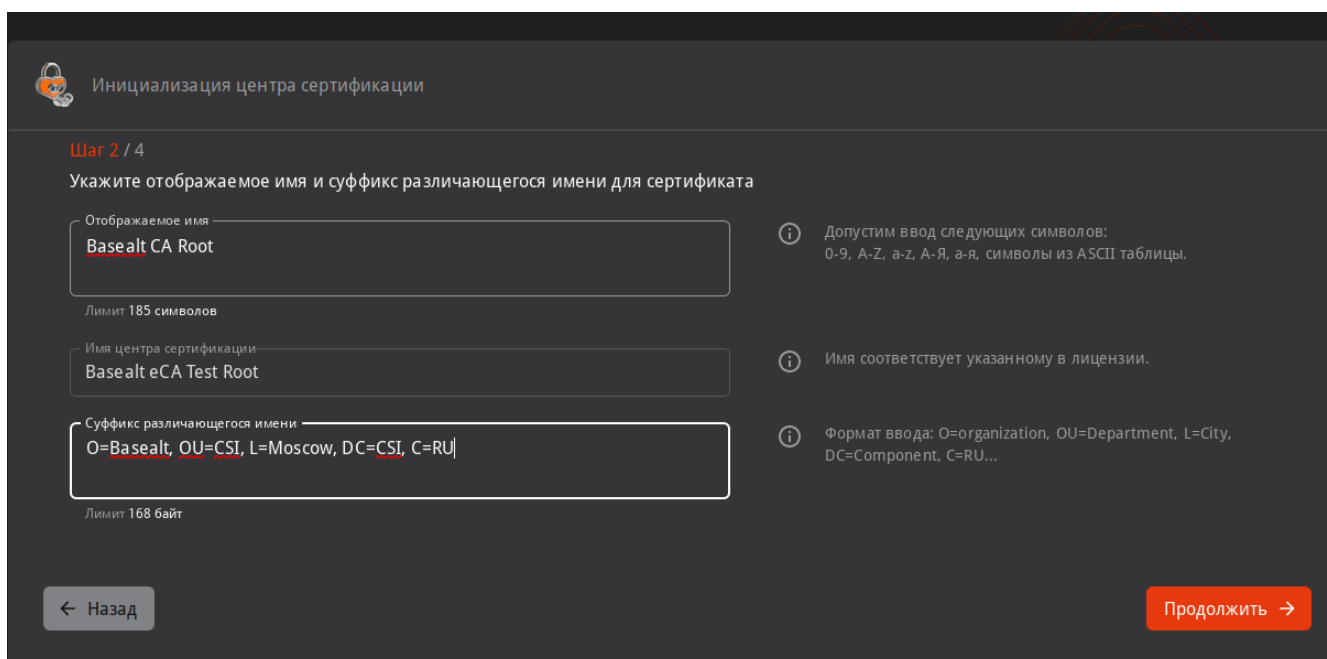
## 5.4. Добавление лицензии и активация корневого и подчинённого ЦС

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

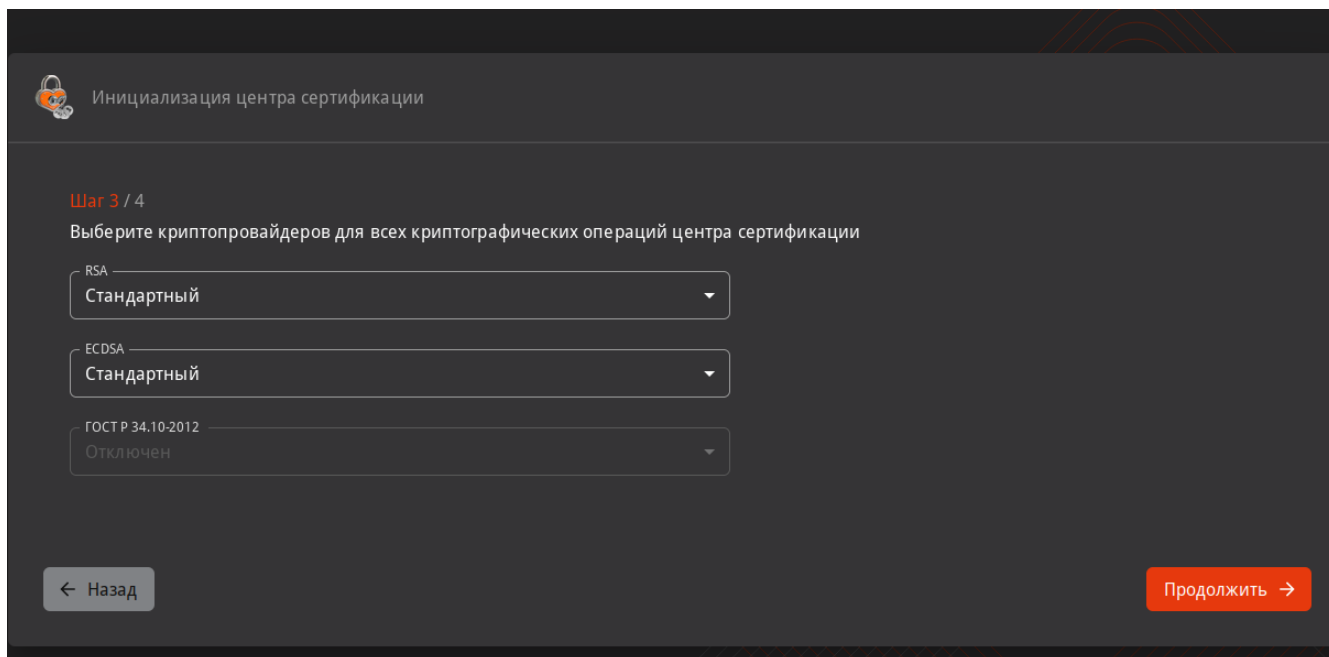
Нажмите **Выбрать файл** и добавьте файл лицензии. Затем нажмите **Проверить**. В зависимости от загруженной лицензии ЦС становится либо корневым, либо подчинённым.



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



На **Шаг 3** выберите криптопровайдеров для соответствующих криптографических операций:



Инициализация центра сертификации

Шаг 3 / 4

Выберите криптопровайдеров для всех криптографических операций центра сертификации

RSA  
Стандартный

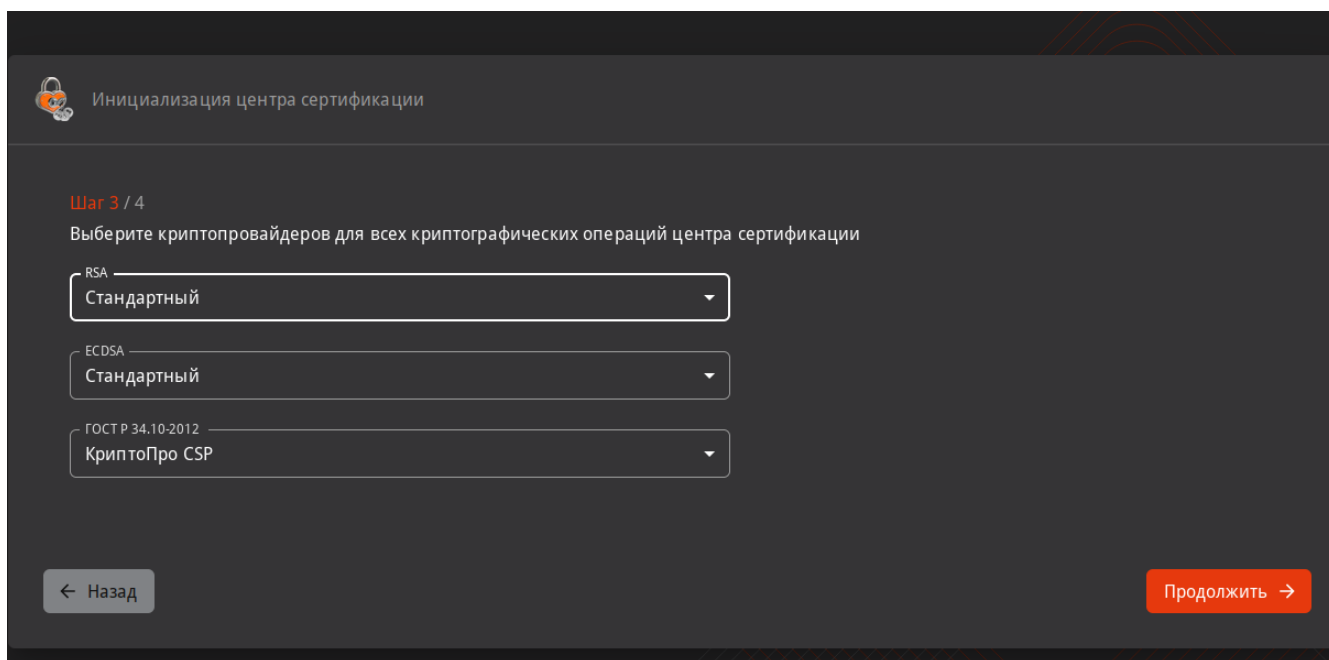
ECDSA  
Стандартный

ГОСТ Р 34.10-2012  
Отключен

← Назад

Продолжить →

Для подчинённого ЦС, где установлен **КриптоПро CSP 5**, на **Шаг 3** выберите следующие криптопровайдеры:



Инициализация центра сертификации

Шаг 3 / 4

Выберите криптопровайдеров для всех криптографических операций центра сертификации

RSA  
Стандартный

ECDSA  
Стандартный

ГОСТ Р 34.10-2012  
КриптоПро CSP

← Назад

Продолжить →

На **Шаг 4** укажите срок действия сертификата ЦС, и заполните **Параметры криптографии**. Затем нажмите **Создать ЦС**:

Инициализация центра сертификации

Укажите срок действия ЦС и параметры криптографии

Срок действия ЦС  
13.01.2035

Минимальный срок действия ЦС — 1 год,  
максимальный — 25 лет

Параметры криптографии

Алгоритм ключа  
RSA

Длина ключа  
2048

Алгоритм хэш-суммы  
SHA256

← Назад

Создать ЦС →

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

Инициализация центра сертификации

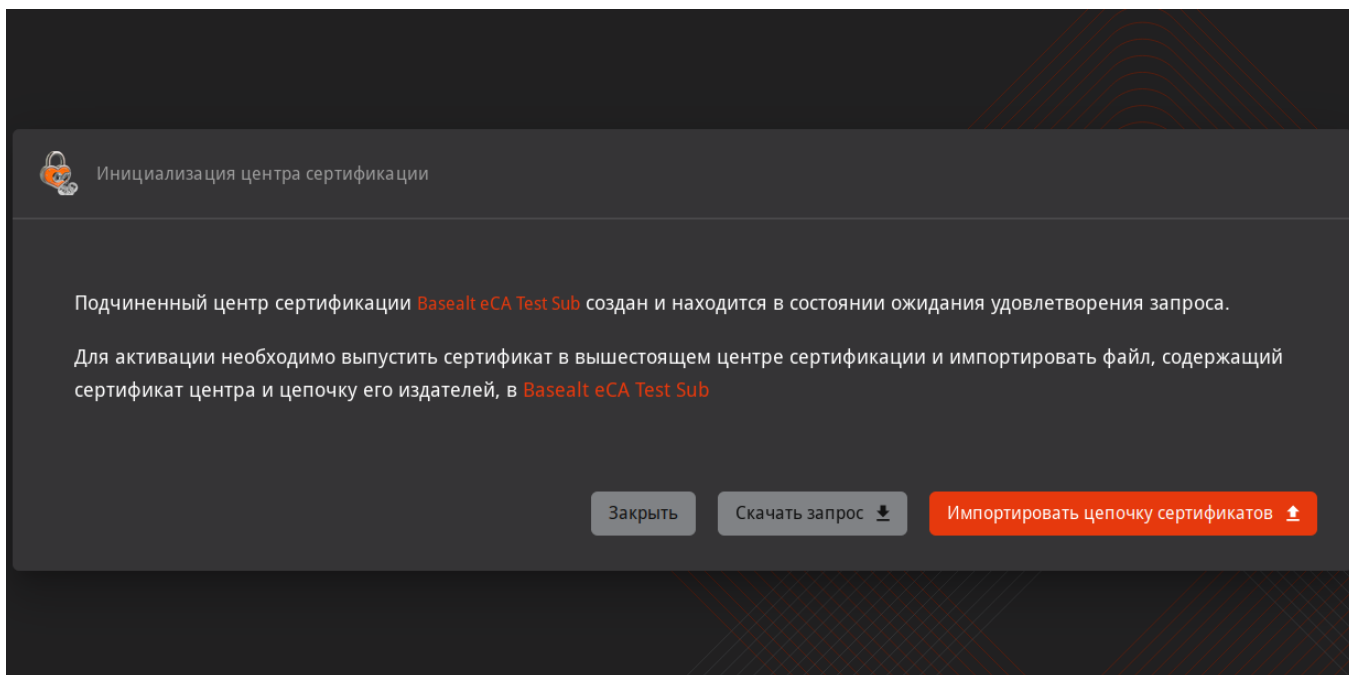
Корневой центр сертификации **Basealt eCA Test Root** успешно создан.

Скачать сертификат

Скачать цепочку

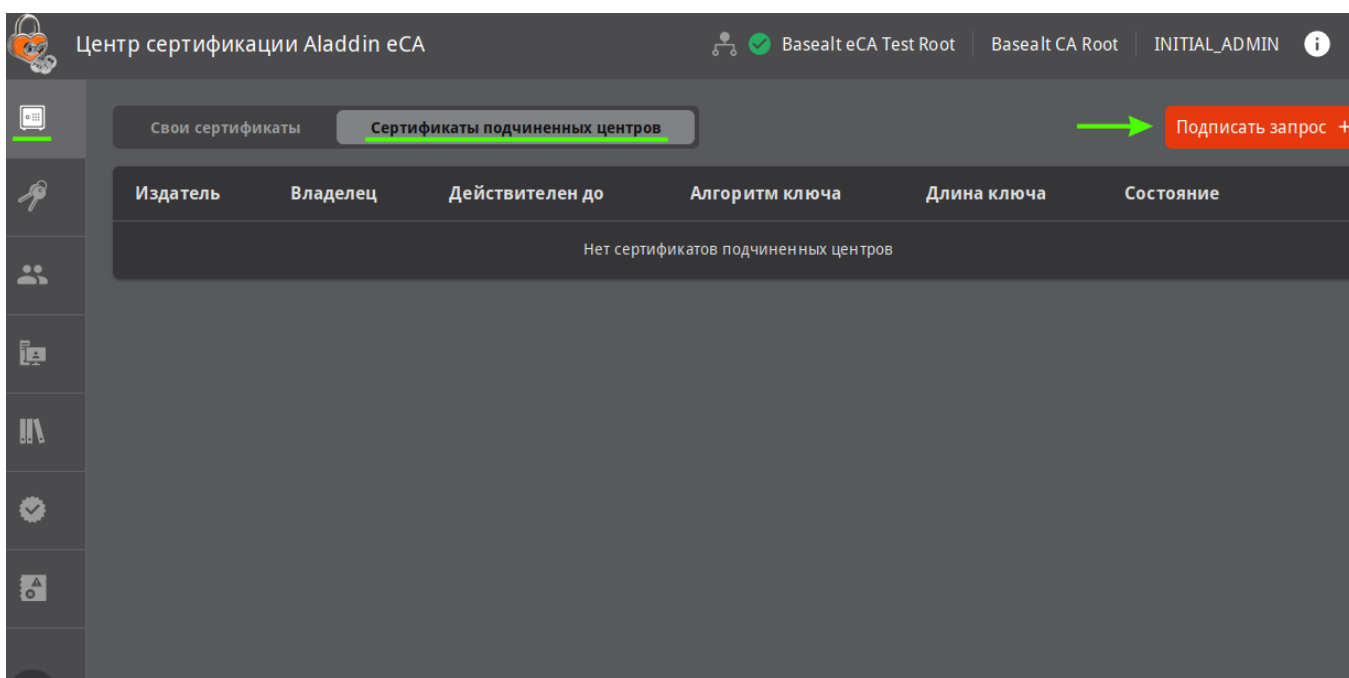
Открыть центр сертификации

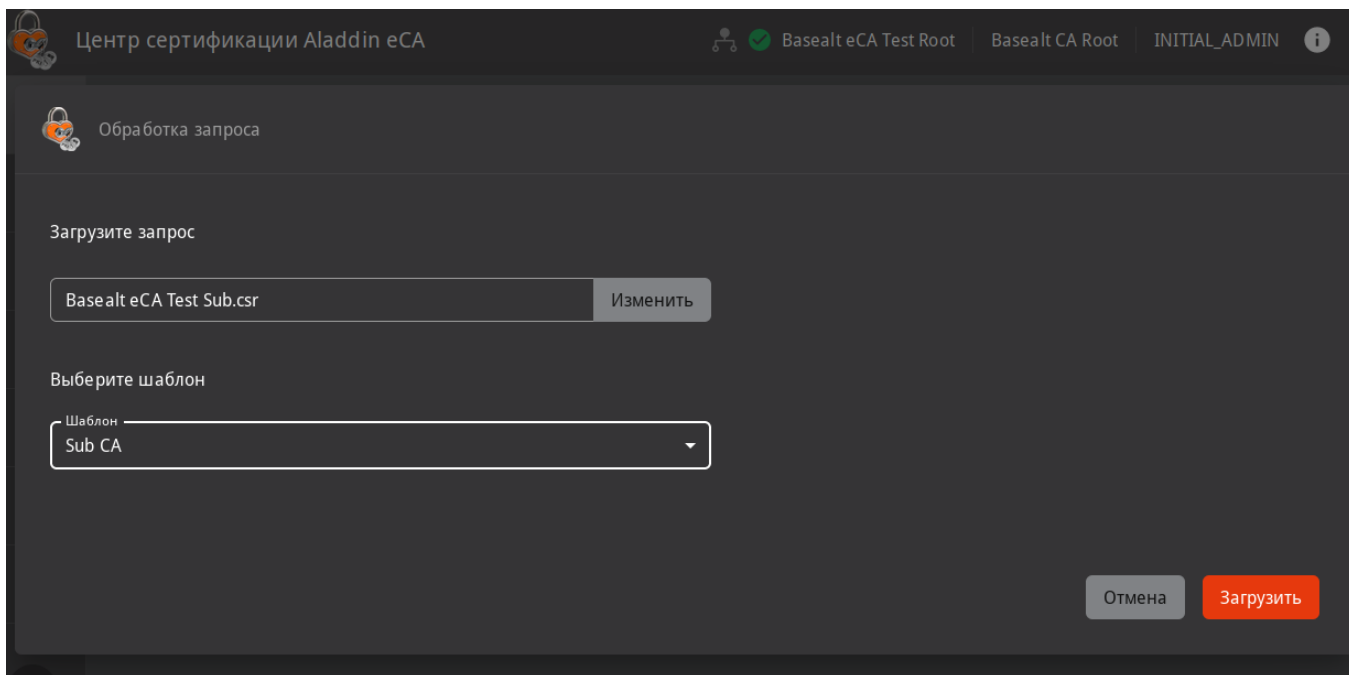
После выполнения аналогичных шагов для подчинённого ЦС — добавления файла лицензии, указания суффикса различающегося имени и параметров криптографии, последний перейдёт в состояние ожидания удовлетворения запроса:



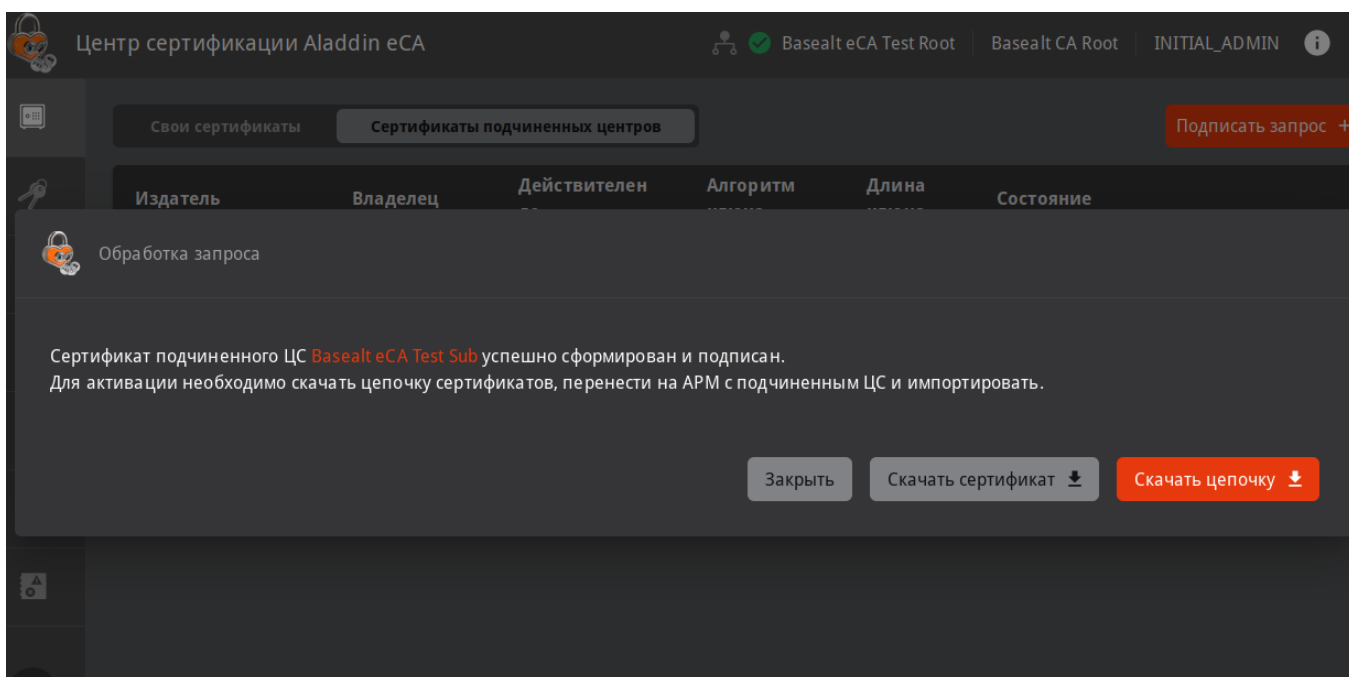
Нажмите **Скачать запрос**. Будет сформирован `.csr`, который необходимо перенести на ПК, с которого осуществляется доступ к веб-интерфейсу корневого ЦС.

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



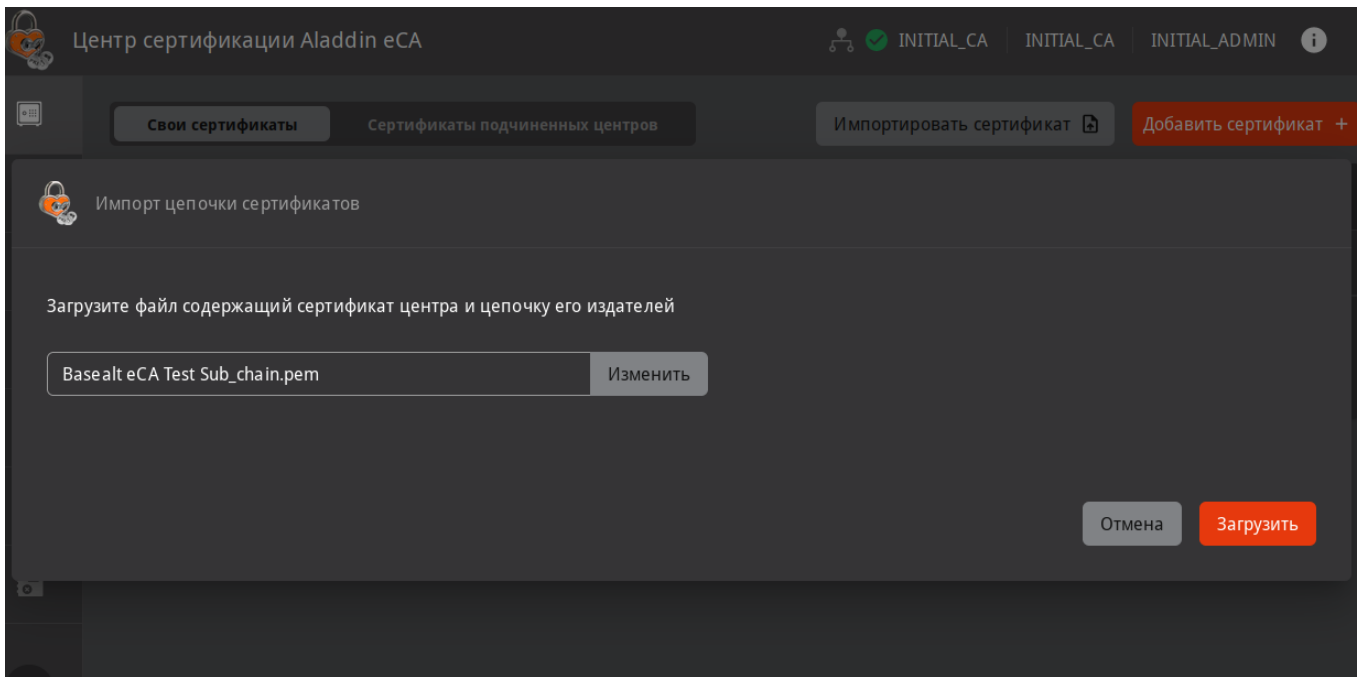


Загрузите перенесённый с подчинённого ЦС запрос (в примере `Basealt eCA Test Sub.csr`). После загрузки запроса должны получить сообщение «Сертификат подчинённого ЦС Basealt eCA Test Sub успешно сформирован и подписан». Нажмите **Скачать цепочку**:

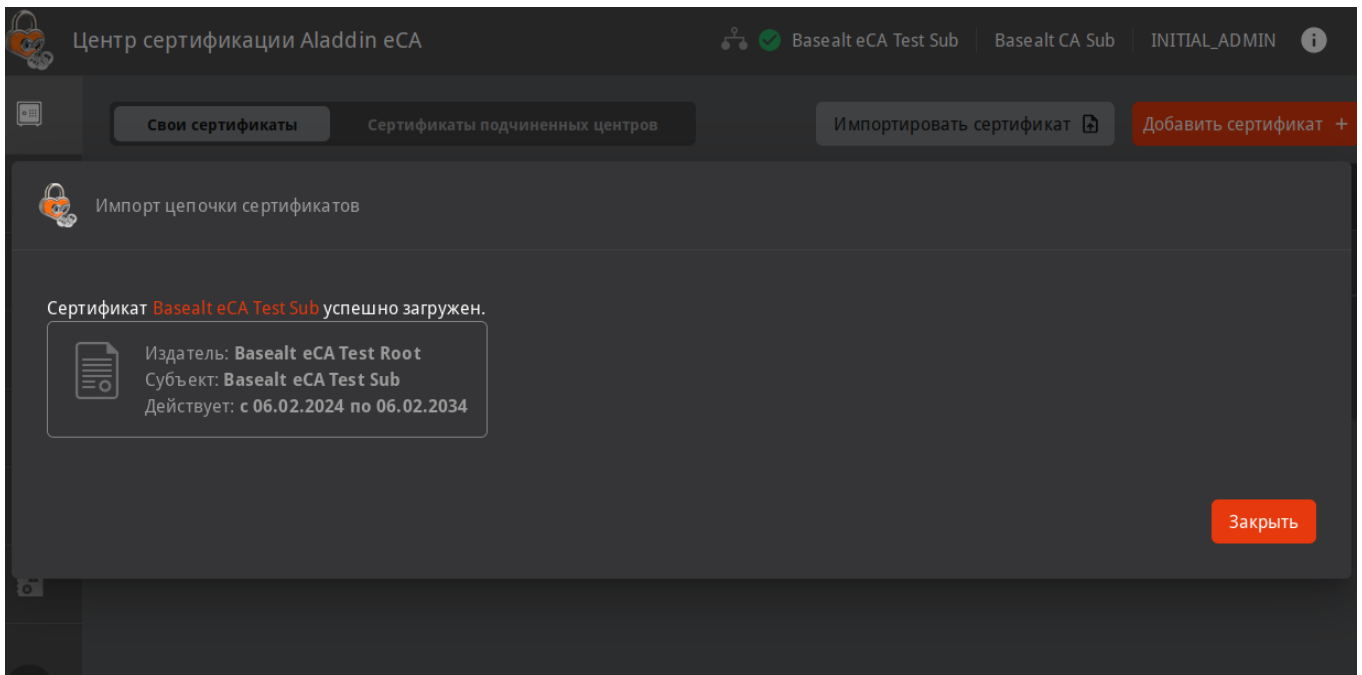


Перенесите загруженный файл цепочки сертификатов (в примере `Basealt eCA Test Sub_chain.pem`) на подчинённый ЦС. На подчинённом ЦС нажмите **Импортировать цепочку сертификатов**, затем **Загрузить**:





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



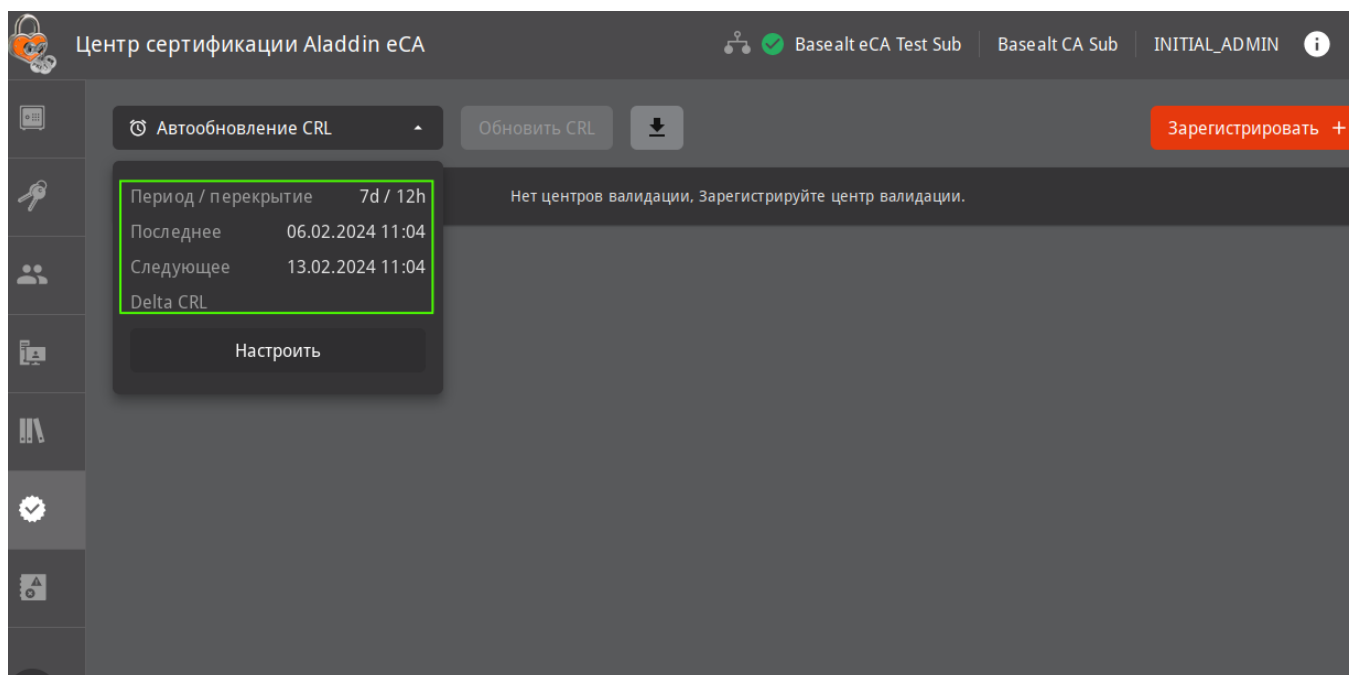
После вышеуказанных действий корневой и подчинённый ЦС должны быть в состоянии **Активирован**:

Отображаемое имя	Владелец	Действителен до	Алгоритм ключа	Длина ключа	Состояние	Кол-во выпущенных
Basealt CA Sub	Basealt eCA Test Sub	06.02.2034 10:30:48	RSA	2048	Активирован	-

## 5.5. Регистрация ЦВ и активация службы OCSP

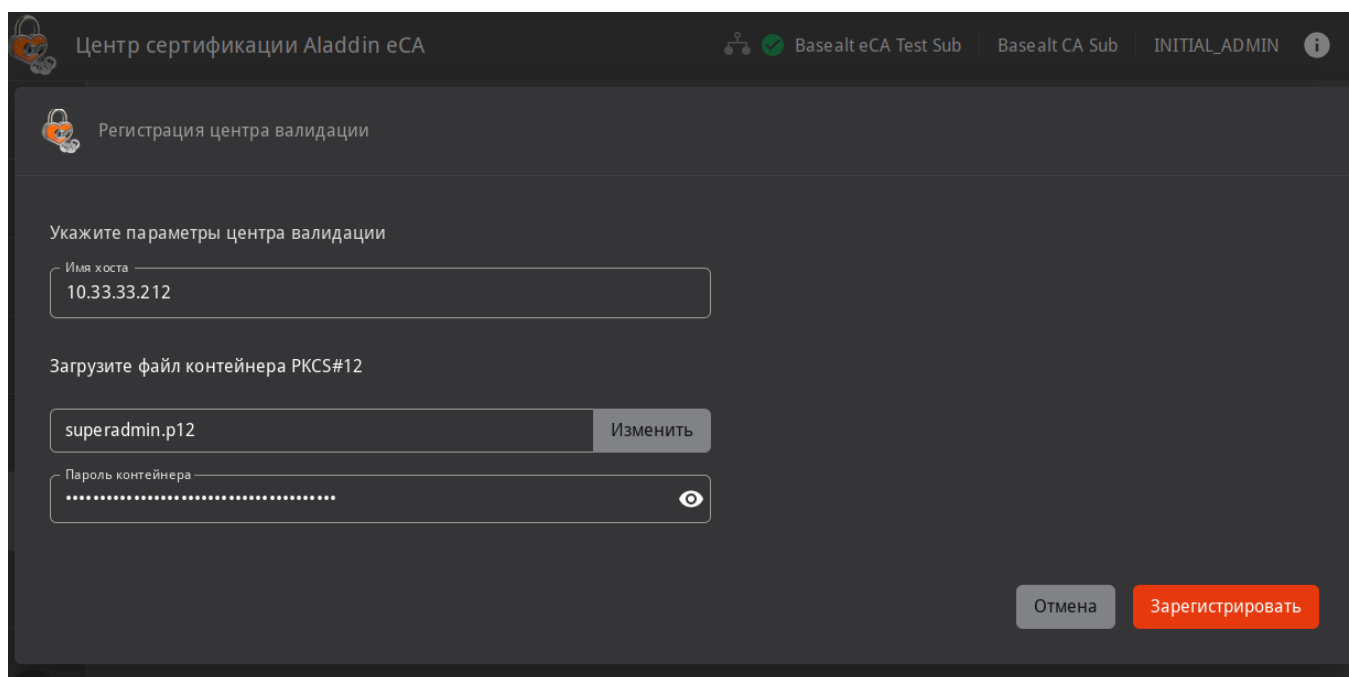
Регистрация ЦВ производится на подчинённом ЦС.

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

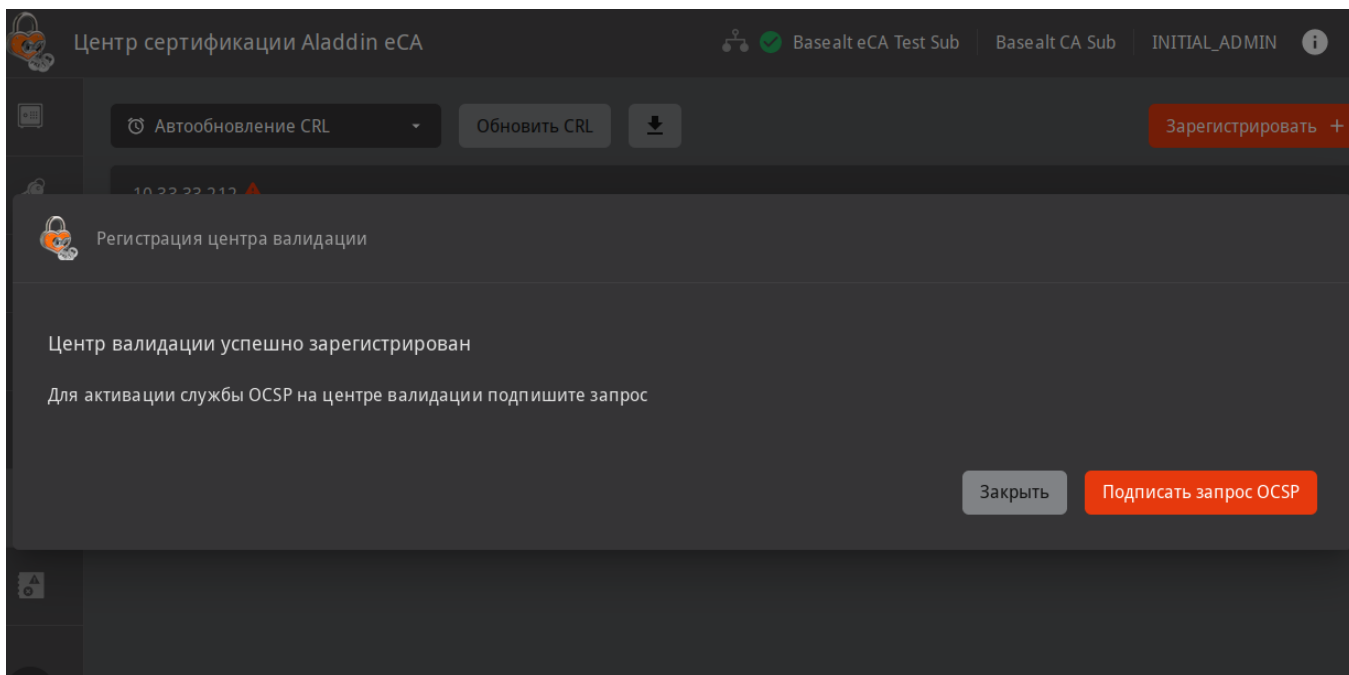


Если период публикации CRL не настроен, то это необходимо сделать.

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**:

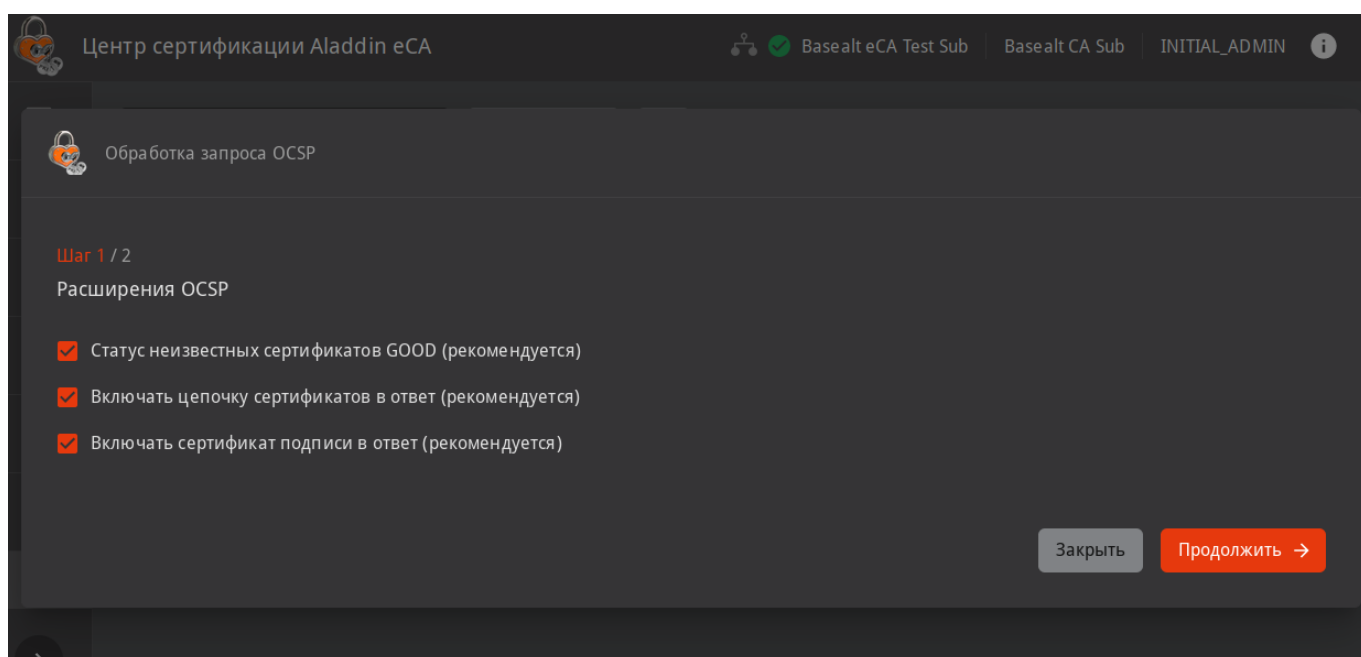


Должны получить сообщение об успешной регистрации ЦВ:

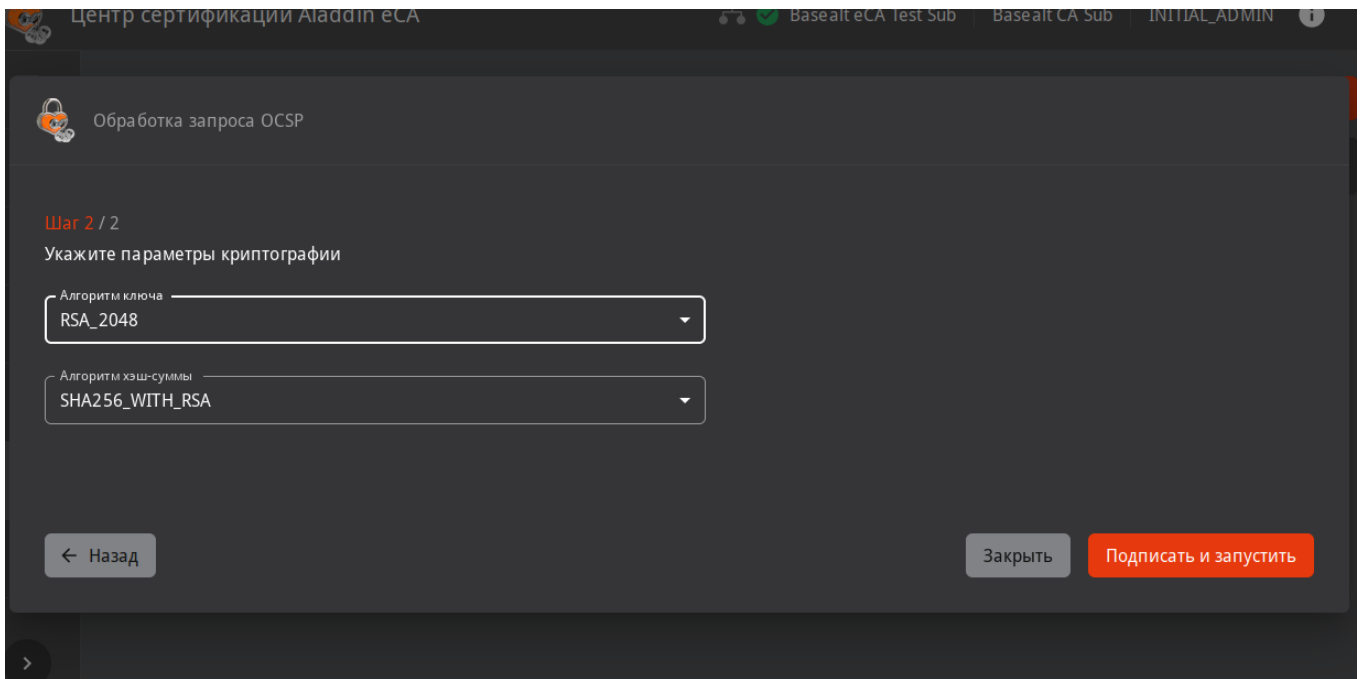


Нажмите кнопку **Подписать запрос OCSP**.

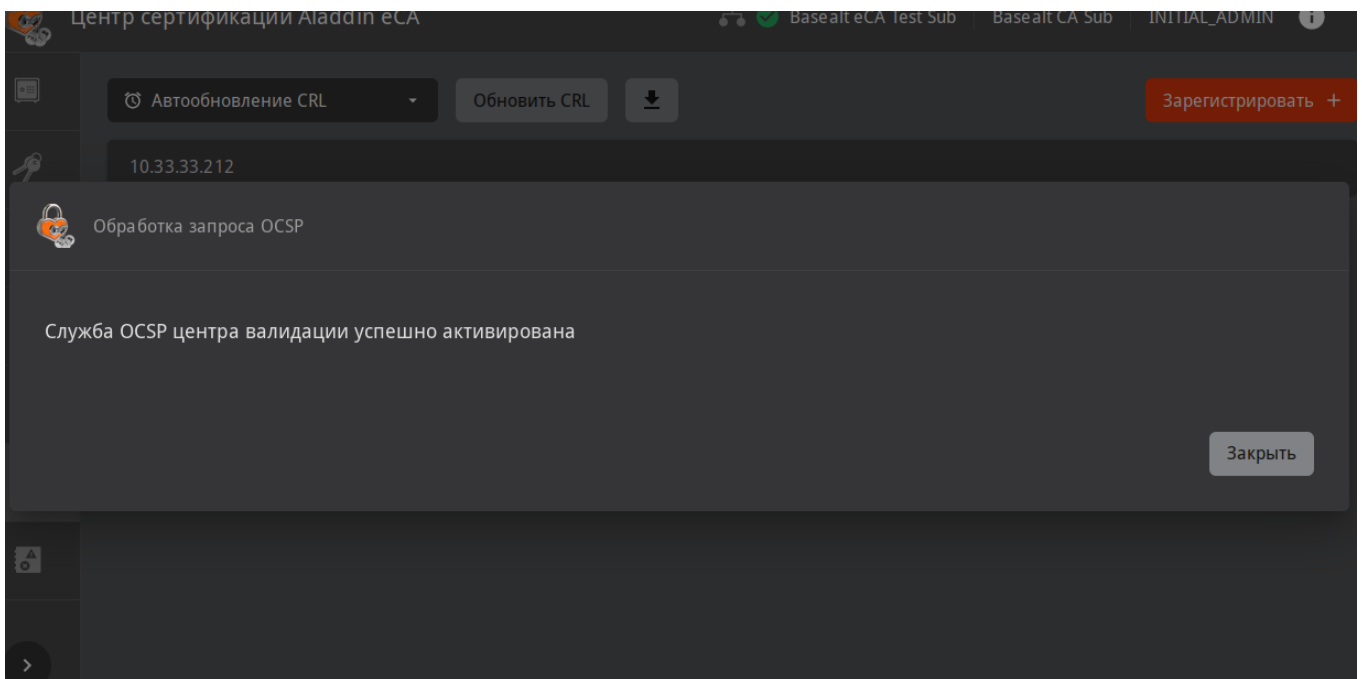
На **Шаг 1** отметьте пункты следующим образом:



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



Должны получить сообщение, что служба OSCP успешно активирована:



## 6. Подключение ресурсной системы

В качестве ресурсной системы будет использоваться доменная структура **Samba DC**.

При разворачивании **Samba DC** 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 10.33.33.211):

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

```
# scp /var/tmp/smbdc.der user@10.33.33.211:/var/tmp/
```

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

```
$ dirname $(dirname $(readlink -f $(which java)))  
/usr/lib/jvm/java-17-openjdk-17.0.10.0.7-alt1.x86_64
```

Хранилище ключей и сертификатов для **java** располагается в поддиректории `lib/security/cacerts`. Следовательно, полный путь — `/usr/lib/jvm/java-17-openjdk-17.0.9.0.9-alt1.x86_64/lib/security/cacerts`.

Добавьте сертификат **Samba DC** на подчинённом **ЦС** (пароль к хранилищу по умолчанию — «changeit»), используя вычисленный полный путь:

```
$ su -  
# keytool -import -alias smb-cert -keystore \  
/usr/lib/jvm/java-17-openjdk-17.0.13.0.11-alt1.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
```

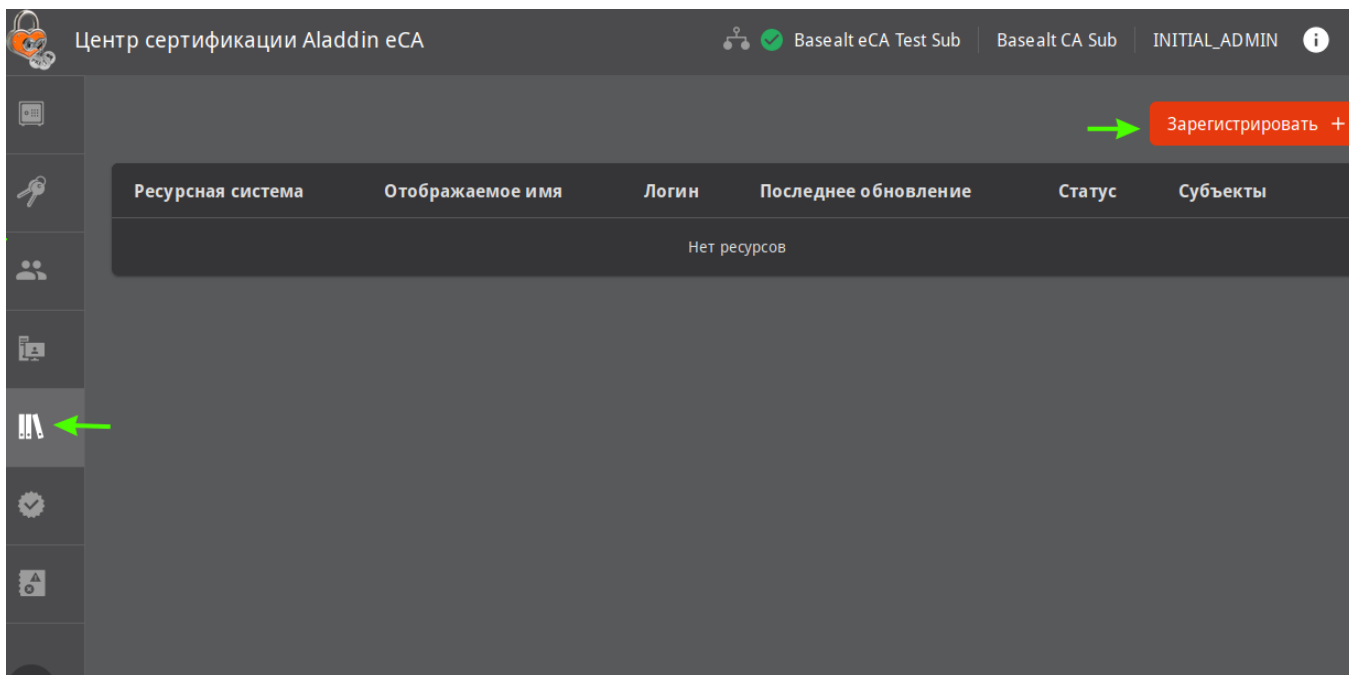
```
Certificate was added to keystore
```

После добавления сертификата **Samba DC** перезапустите службу **aeca-ca.service**:

```
# systemctl restart aeca-ca.service
```

```
# journalctl --follow -u aeca-ca.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:

Центр сертификации Aladdin eCA

Basealt eCA Test Sub | Basealt CA Sub | INITIAL\_ADMIN

Зарегистрировать +

Ресурсная система	Отображаемое имя	Логин	Последнее обновление	Статус	Субъекты
Samba DC	Samba DC	administrato...	10:49:20 09.02.2024	<u>Успешно</u>	4

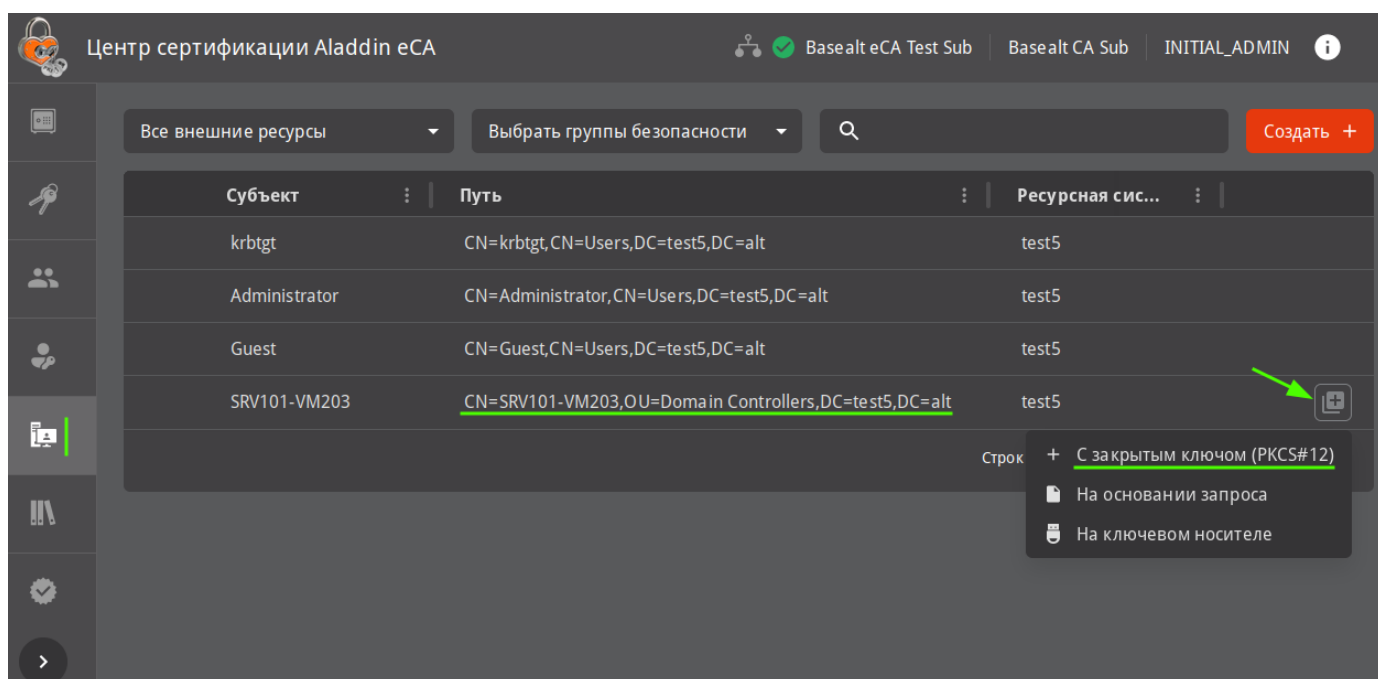


# 7. Обеспечение возможности строгой аутентификации пользователей в домене

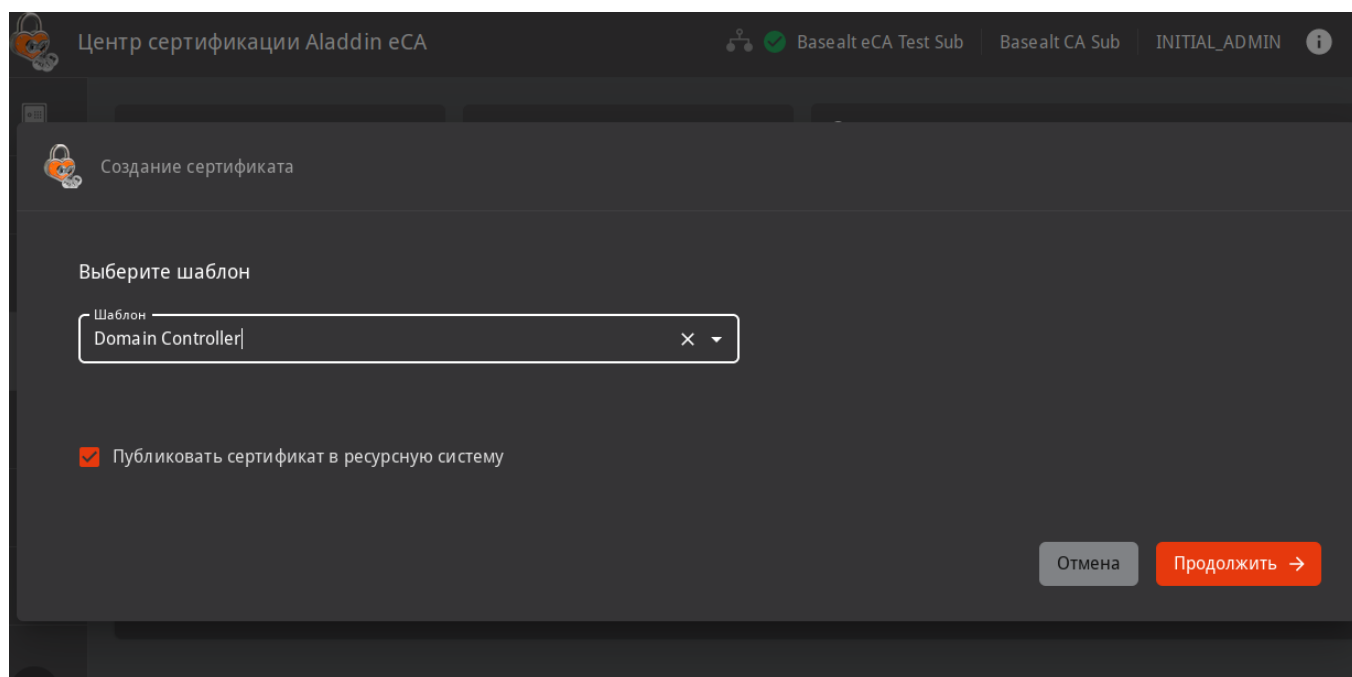
Под строгой аутентификацией подразумевается аутентификация при помощи сертификата, выданного в **AcSA** для доменного пользователя и размещённого на токене.

## 7.1. Выдача сертификата контроллера домена

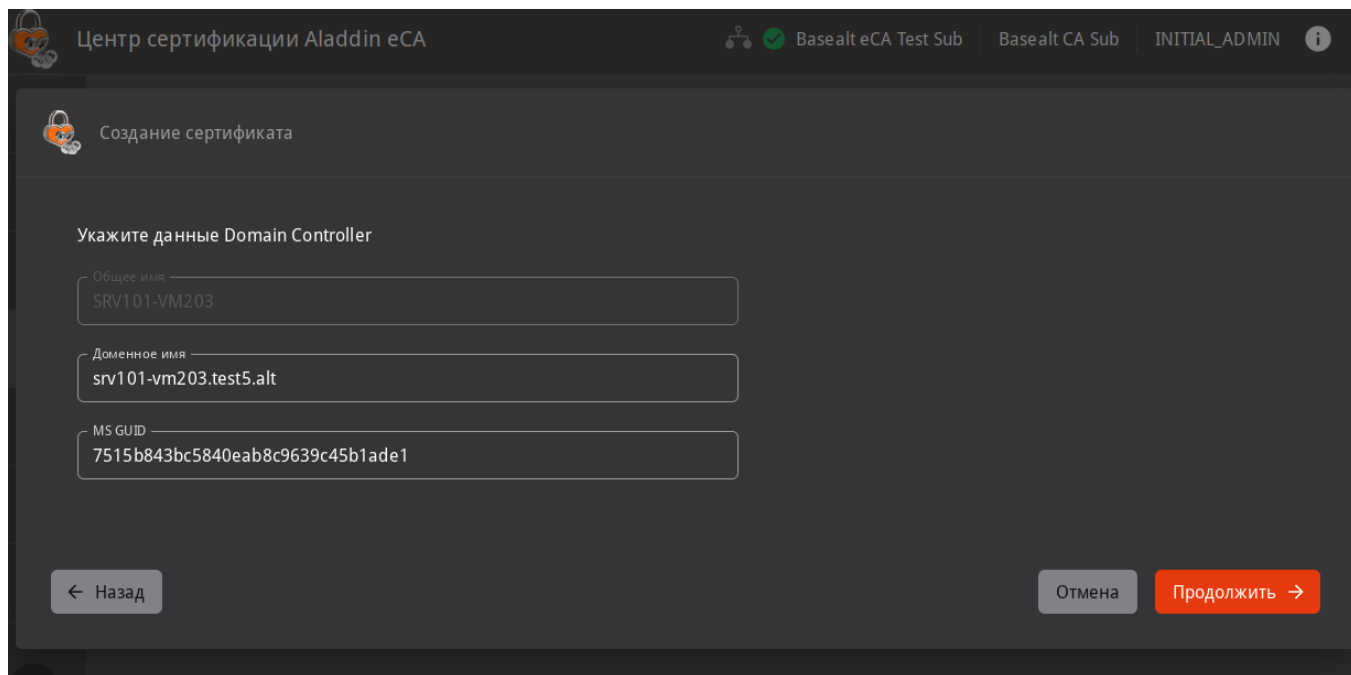
В веб-интерфейсе подчинённого ЦС перейдите в **Субъекты**, выберите контроллер домена и нажмите **Создать сертификат. С закрытым ключом (PKCS#12)**:



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



На следующем шаге **Укажите данные Domain Controller** данные контроллера домена должны заполниться автоматически:



Центр сертификации Aladdin eCA

Basealt eCA Test Sub | Basealt CA Sub | INITIAL\_ADMIN

Создание сертификата

Укажите данные Domain Controller

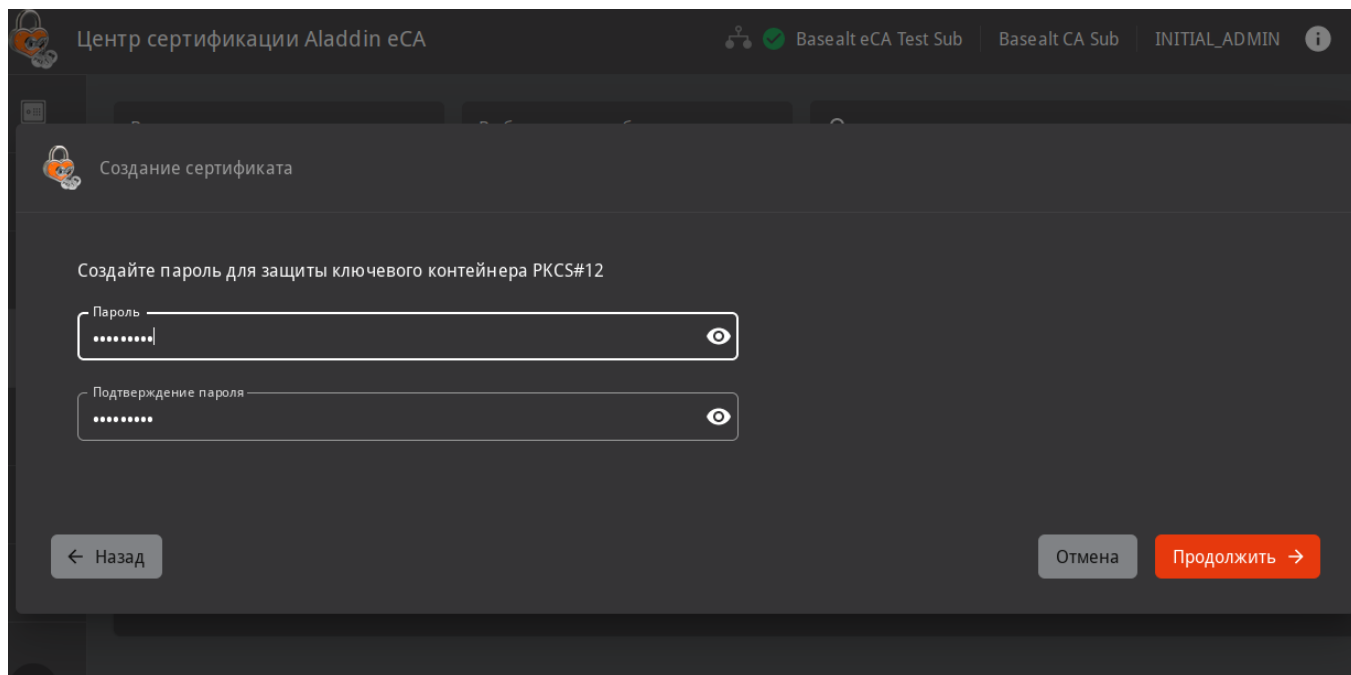
Общее имя  
SRV101-VM203

Доменное имя  
srv101-vm203.test5.alt

MS GUID  
7515b843bc5840eab8c9639c45b1ade1

← Назад | Отмена | Продолжить →

Далее укажите пароль для ключевого контейнера:



Центр сертификации Aladdin eCA

Basealt eCA Test Sub | Basealt CA Sub | INITIAL\_ADMIN

Создание сертификата

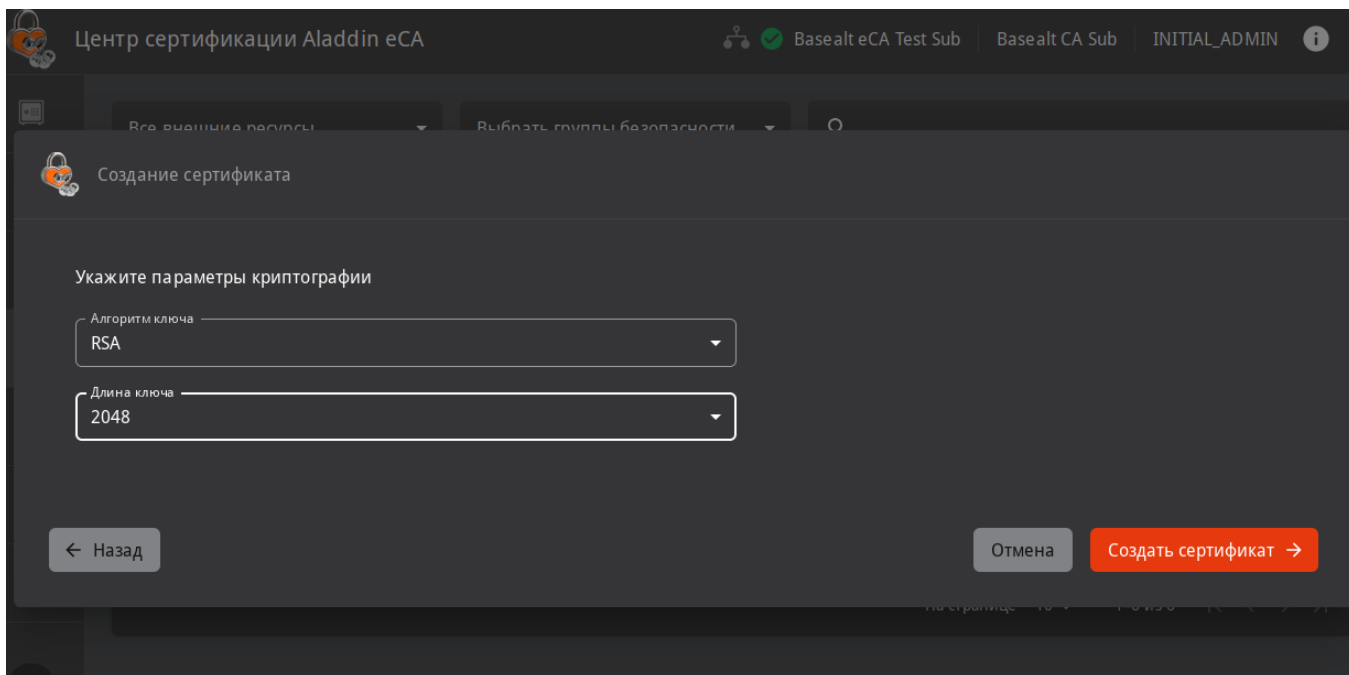
Создайте пароль для защиты ключевого контейнера PKCS#12

Пароль  
.....

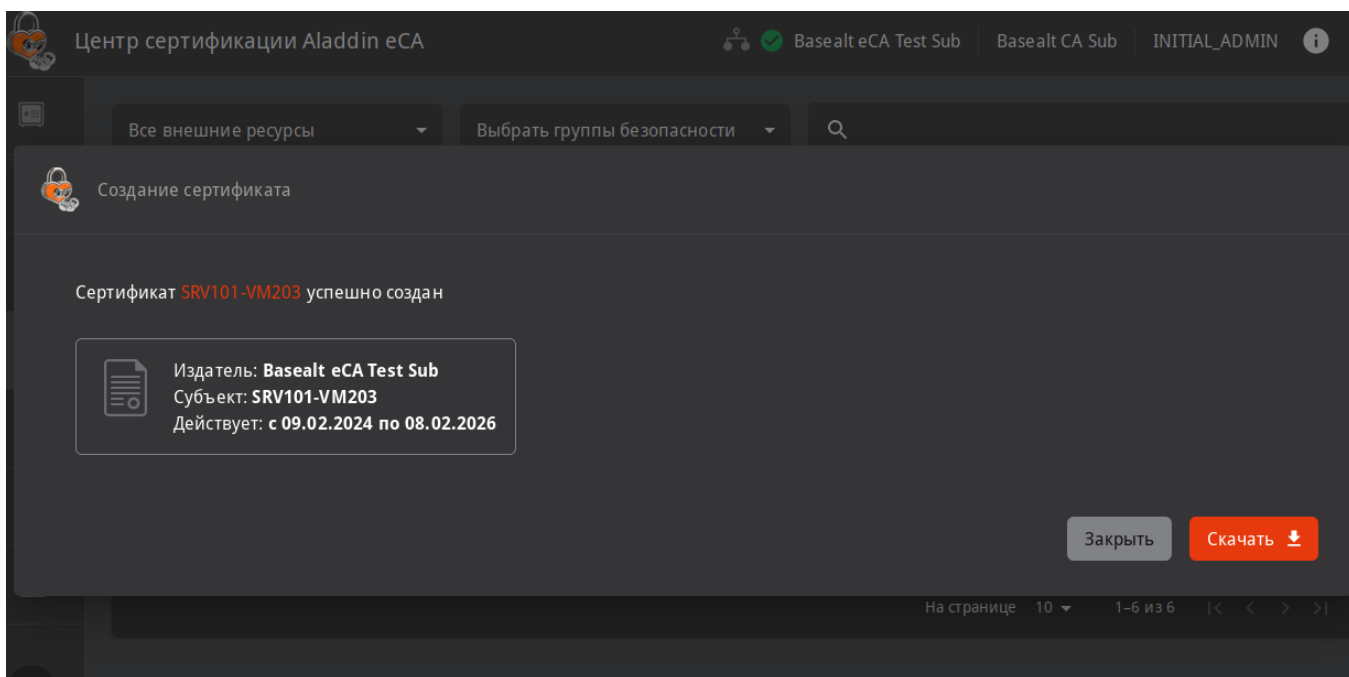
Подтверждение пароля  
.....

← Назад | Отмена | Продолжить →

Затем укажите параметры криптографии и нажмите **Создать сертификат**:



Скачайте сертификат после его создания:



## 7.2. Настройка службы Kerberos контроллера домена Samba DC

Перенесите полученный контейнер `p12 Samba DC` (в примере `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
```



Параметр `pkinit_identity`, сертификат `DC.crt.pem` и ключ `DC.key.pem` должны быть на одной строке!

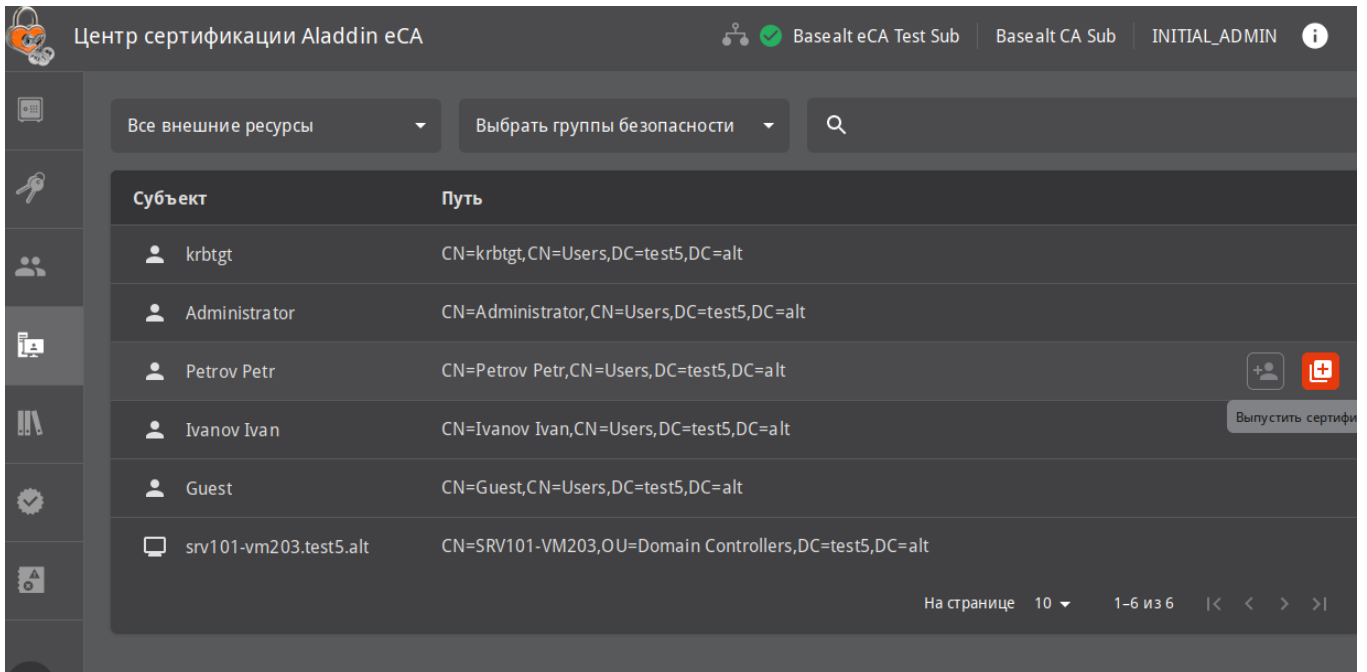
Перезапустите службу `samba`:

```
# 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`):

Центр сертификации Aladdin eCA Basealt eCA Test Sub Basealt CA Sub INITIAL\_ADMIN

Создание сертификата

Шаг 1 / 3  
Выберите устройство и шаблон

Устройство  
JaCarta-4E46000625123957 131071

PIN-код  
.....

Шаблон  
Smartcard Logon

Публиковать сертификат в ресурсную систему

Отмена Продолжить →

Подтвердите доверие для работы с токеном:

Центр сертификации Aladdin eCA Basealt eCA Test Sub Basealt CA Sub INITIAL\_ADMIN

Создание сертификата

Шаг 1 / 3  
Выберите устройство

Устройство  
JaCarta-4E46000625

PIN-код  
.....

Шаблон  
Smartcard Logon

Публиковать сертификат в ресурсную систему

Отмена Продолжить →

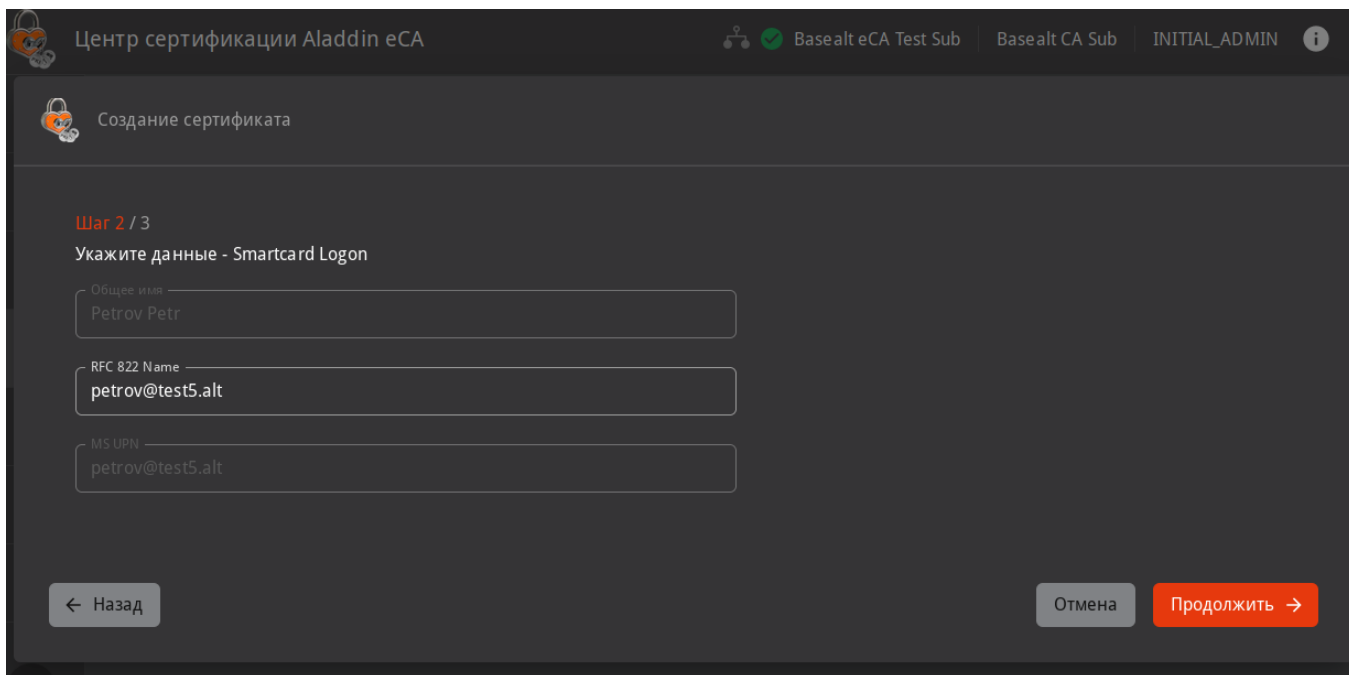
**JC-WebClient. Подтверждение доверия к сайту**

Служба управления токенами обнаружила, что сайт **10.33.33.211** пытается обратиться к Вашему токenu JaCarta PKI.  
Для предоставления сайту возможности выполнять операции с токеном требуется подтвердить доверие к сайту.  
Если в настоящий момент Вы работаете с сайтом 10.33.33.211 и доверяете ему, то нажмите "Доверяю". Иначе нажмите "Отмена".

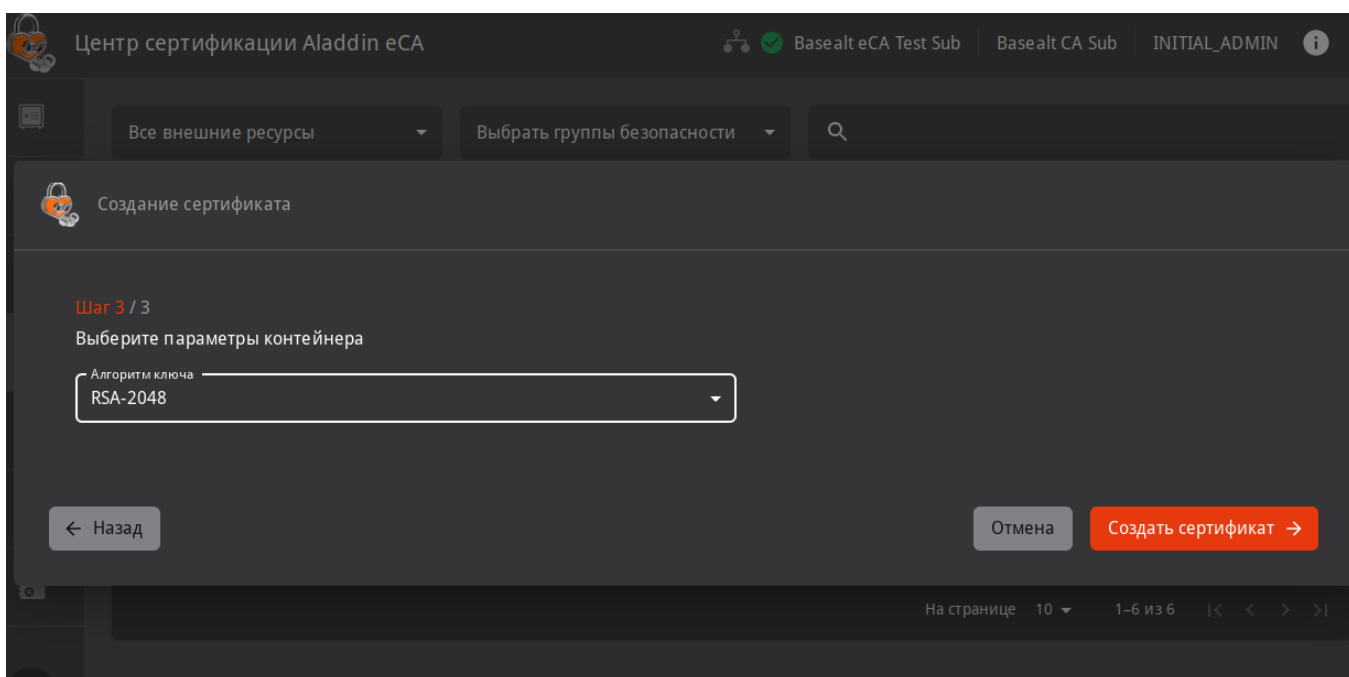
При необходимости Вы сможете удалить сайт из списка доверенных на странице "JC-WebClient. Доверенные сайты", расположенной в меню приложений.

Доверяю Отмена

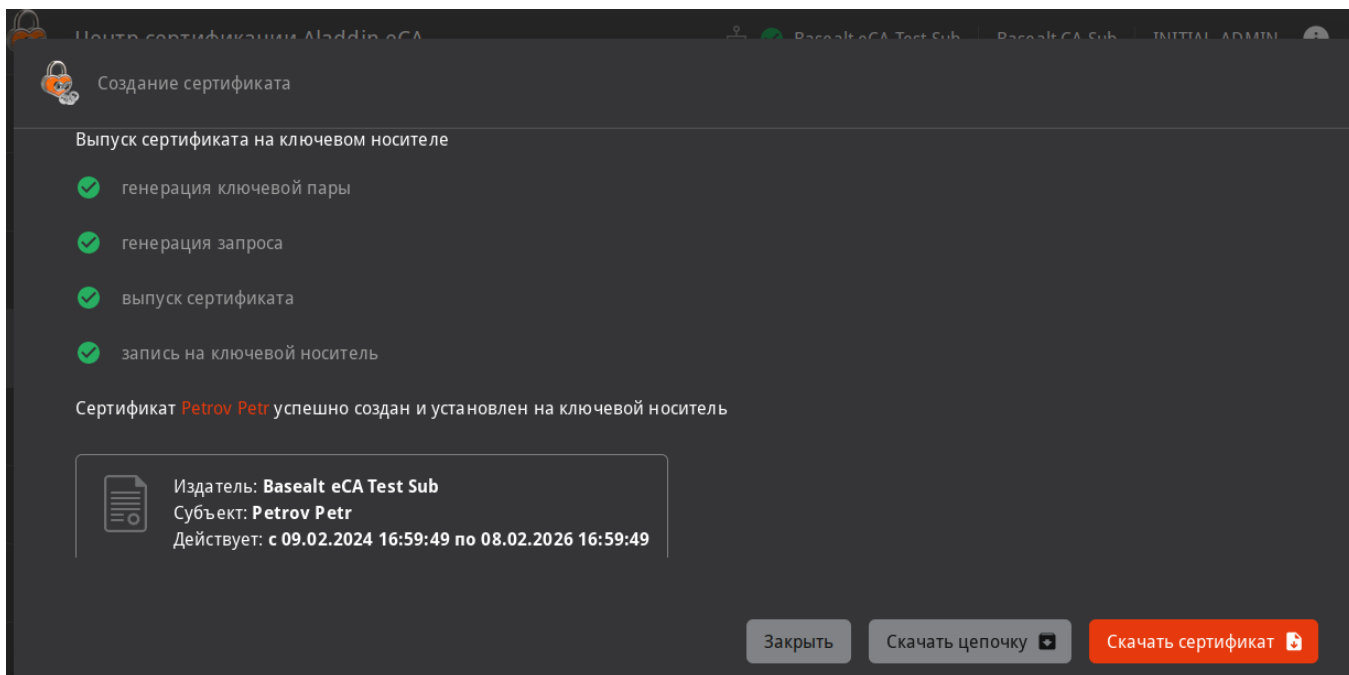
На **Шаг 2/3** укажите данные для шаблона сертификата **Smartcard Logon**:



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  **PKCS#11** , 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/](https://www.aladdin-rd.ru/support/downloads/jacarta_client/)). Перейдите в каталог с вышеуказанным распакованным ПО (в примере  **/var/tmp** ) и установите его.

Установка  **Единый Клиент JaCarta** :

```
$ su -
# cd /var/tmp
# chmod +x jacartauc_3.1.2.3478_rpm_x64/install.sh
# jacartauc_3.1.2.3478_rpm_x64/install.sh
```



Если при установке вы сталкиваетесь с ошибкой "файл  **/usr/lib64/libASEP11.so**  из устанавливаемого пакета  **jcrkcs11-2-...** конфликтует с файлом из пакета  **libjcrkcs11-...**", то удалите пакет  **libjcrkcs11**  (`apt-get remove libjcrkcs11`) 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 jcrkcs11-2:
[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
```

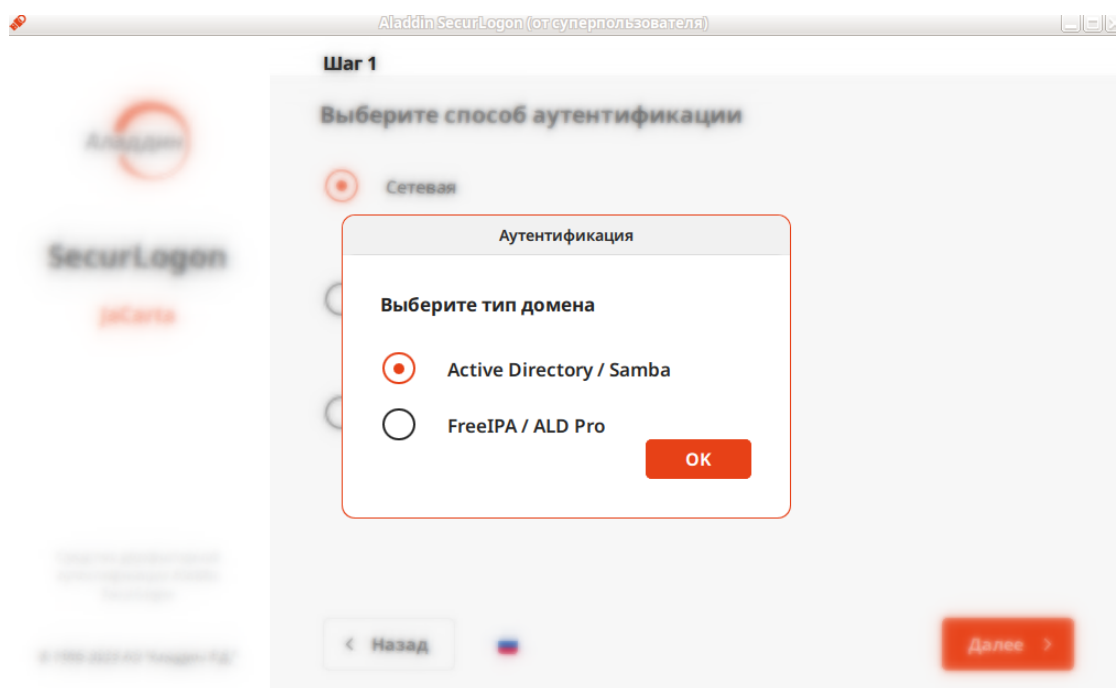
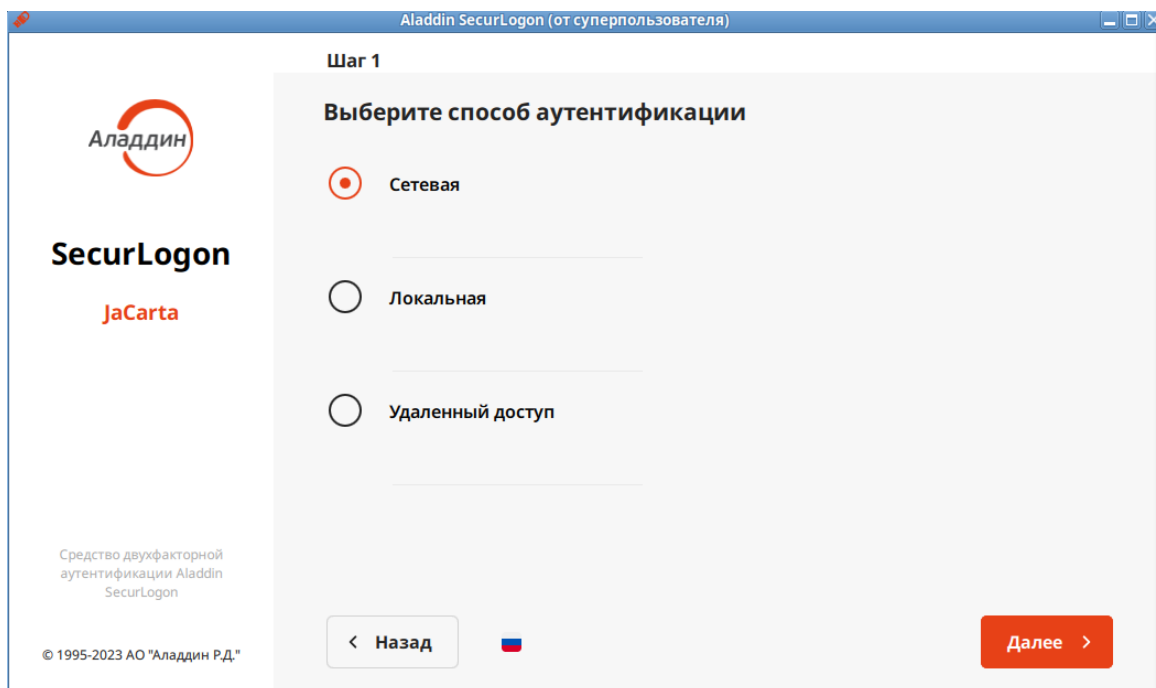
После установки SecurLogon необходимо перезагрузить ПК.

## 8.2. Настройка двухфакторной аутентификации

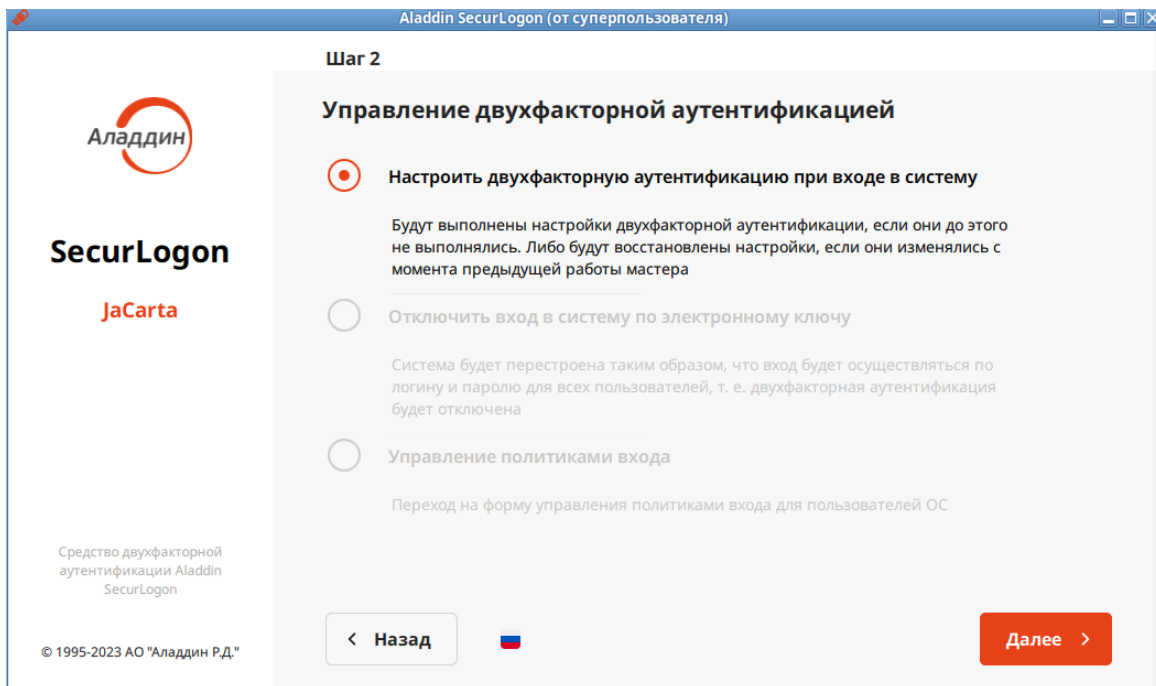
Подключите токен с сертификатом пользователя. Перенесите цепочку сертификатов ЦС (chain.pem) на АРМ пользователя.

Запустите ПО SecurLogon (Меню — Стандартные — SecurLogon).

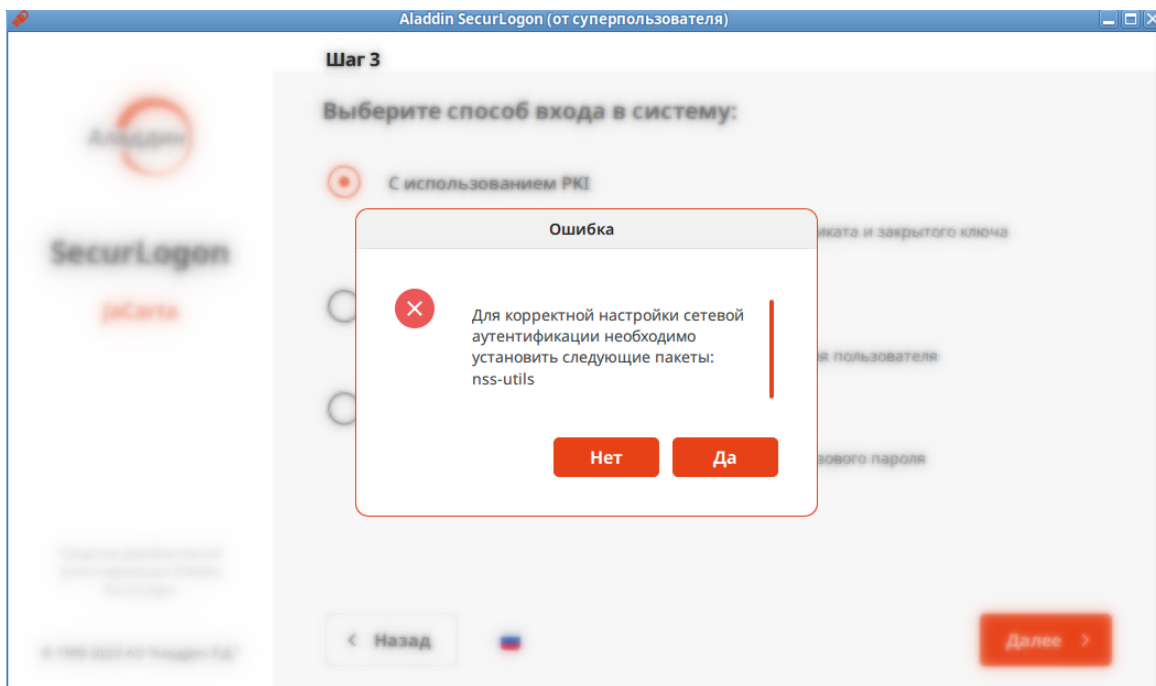
После ввода лицензии на Шаг 1 выберите способ аутентификации Сетевая и тип домена Active Directory/Samba:



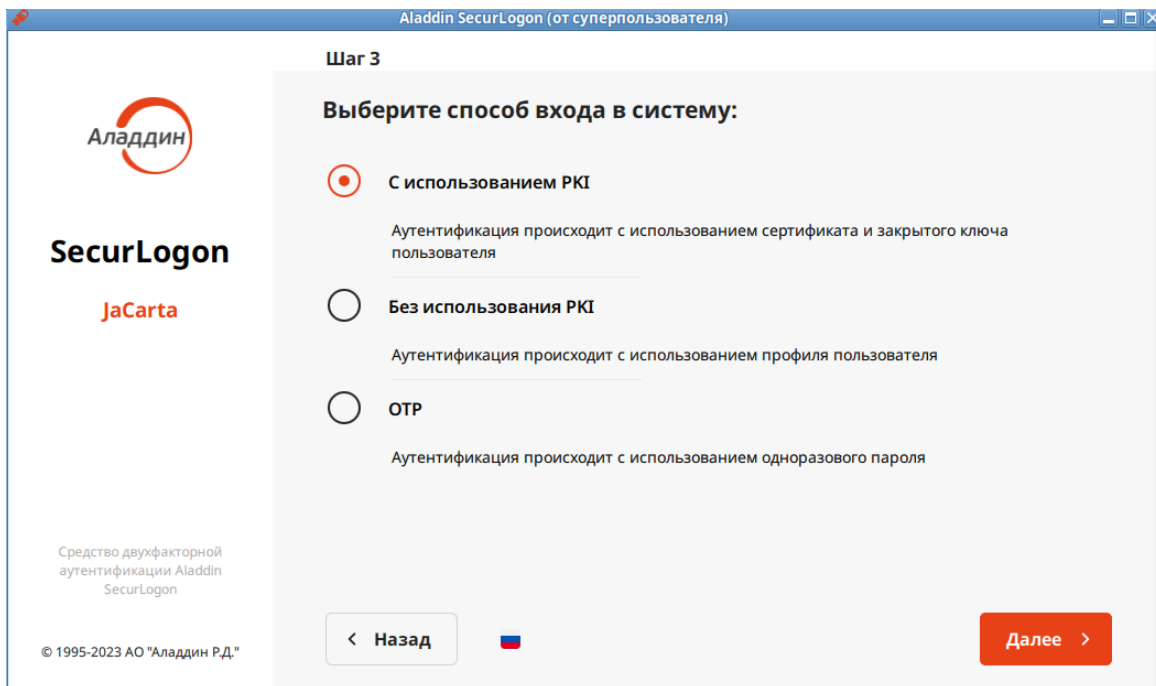
На Шаг 2 выберите Настроить двухфакторную аутентификацию при входе в систему:



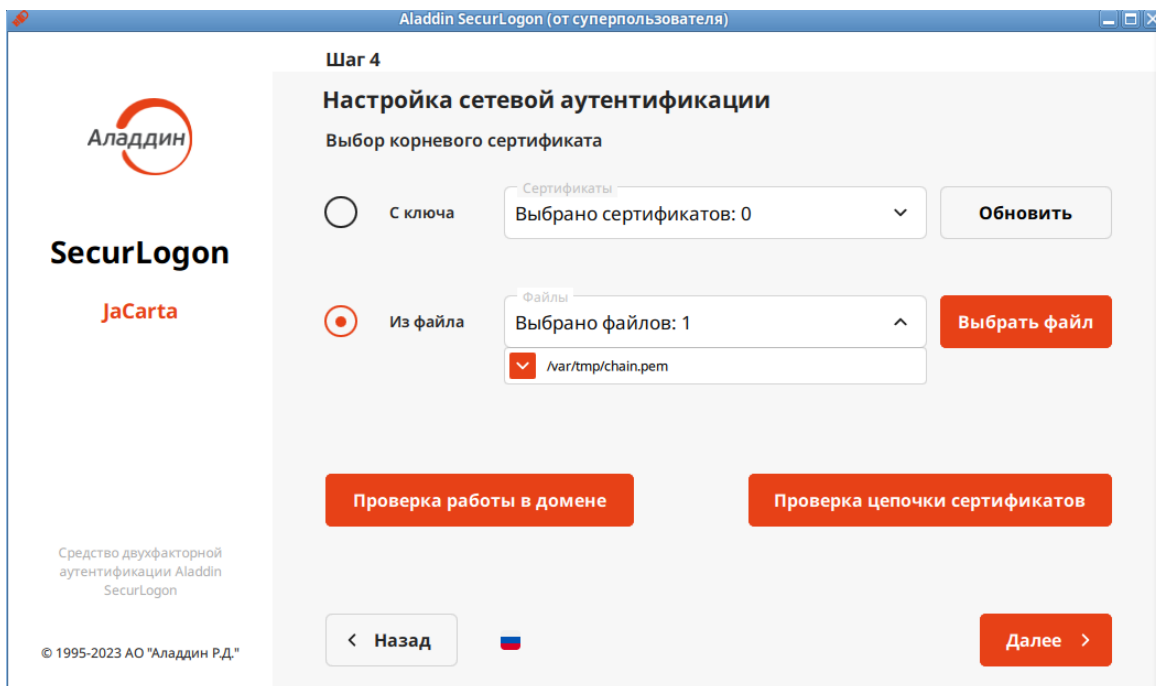
Установите дополнительные пакеты, если будет предложено:



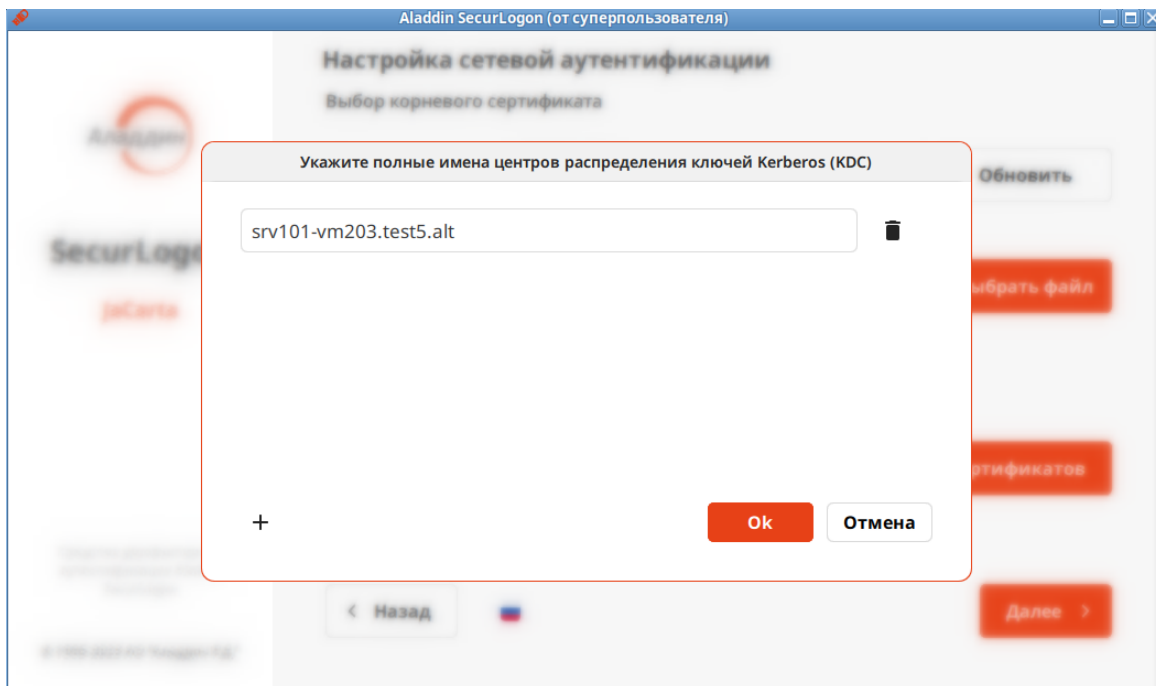
На Шаг 3 выберите **С использованием PKI**:



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

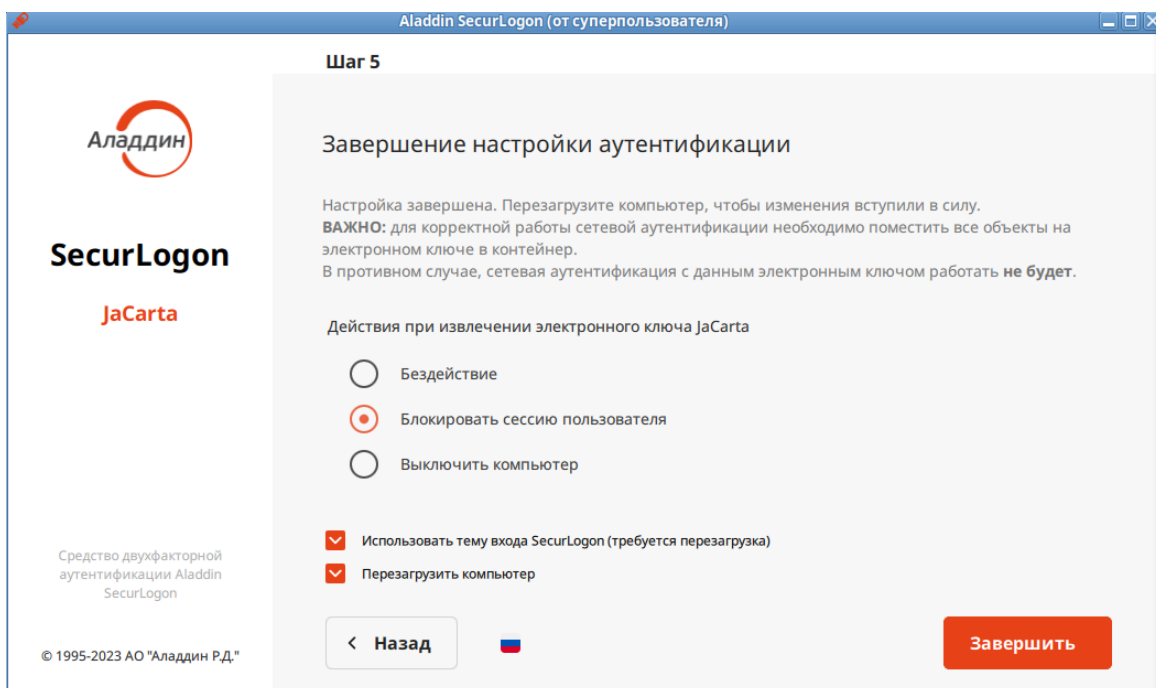


Далее укажите FQDN центра распределения ключей Kerberos (KDC):



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 (если токен мультислотовый), на котором находится сертификат, и ввести PIN-код. Нужный слот будет без восклицательного знака и при этом будет автоматически заполнена учётная запись пользователя:

Добро пожаловать

Petrov Petr  
p101ws-vm141.test5.alt

 JaCar...

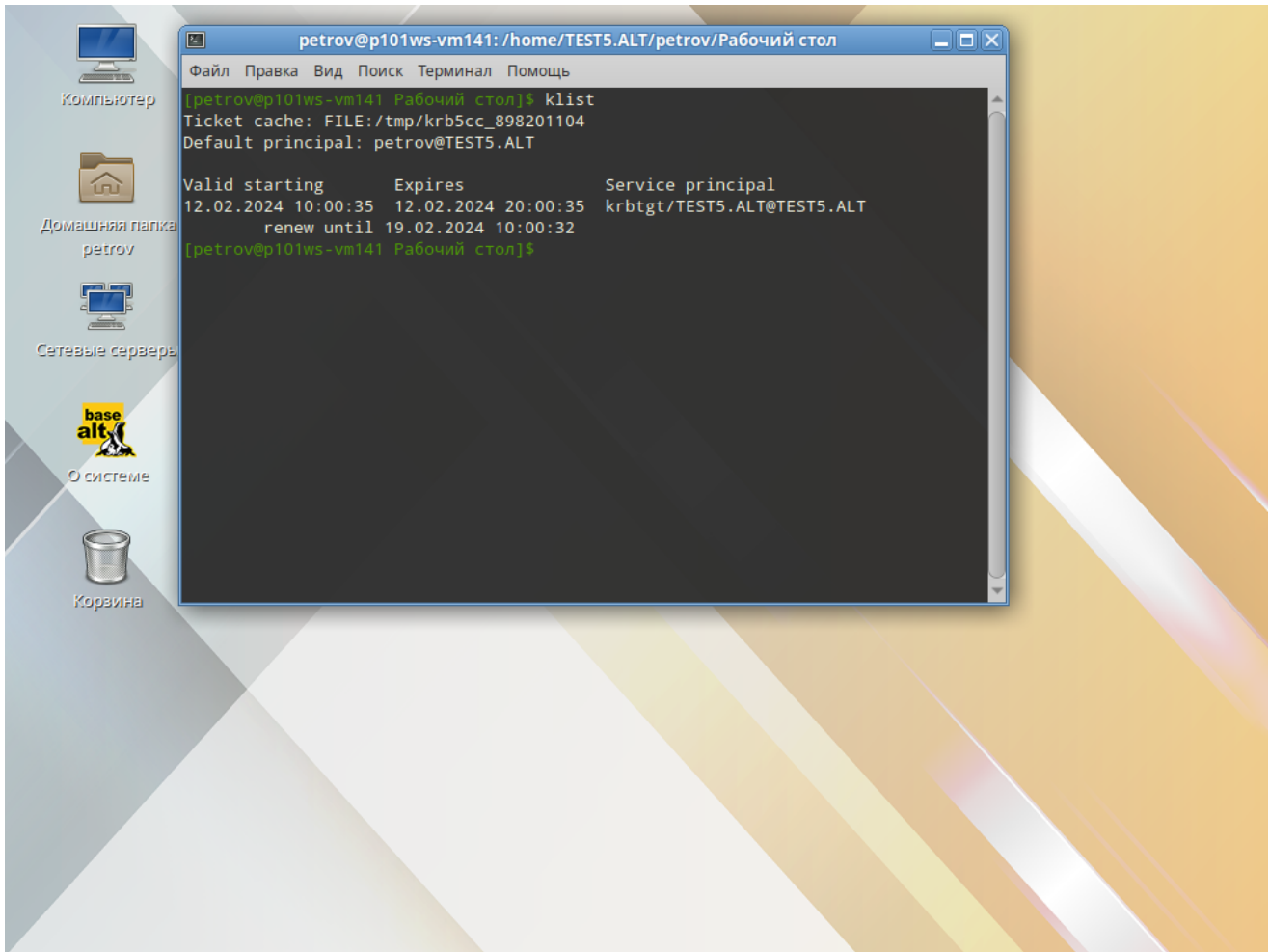
 petrov@test5.alt

 PIN-код

Войти



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



## 9. Удаление AeCA

Для ЦС:

```
# bash /opt/aecaCa/scripts/uninstall.sh
=====
ALADDIN ENTERPRISE CERTIFICATE AUTHORITY
=====
Вы уверены, что хотите продолжить? Изменения необратимы.
1) Yes
2) No
#? 1
...
Очистка / удаление...
1: аеса-са-2.0.1-367 ##### [100%]
Завершено.
=====
Удаление завершено!
```

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
...

```





## **Контакты службы обеспечения совместимости**

**Электронная почта:**

[gost@basealt.ru](mailto: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.