
Визирь.СКУД

Программный комплекс системы видеоидентификации

STC-S857

Версия программного обеспечения: 4.23

Руководство по установке

ЦВАУ.01026-01 92



СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ	4
1.1 Общие положения	4
1.2 Соглашения и обозначения	4
1.3 Правообладатель	5
2 ОБЩИЕ СВЕДЕНИЯ	6
2.1 Сведения о программе и изготовителе	6
2.2 Техническая поддержка	6
3 УСТАНОВКА ПРОГРАММЫ	7
3.1 Установка в ОС Ubuntu	7
3.2 Установка в ОС Astra Linux	13
4 АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ	21
4.1 Активация ключа (при наличии файла .v2c)	21
4.2 Активация ключа (при отсутствии файла .v2c)	22
4.3 Обновление ключа защиты	25
4.4 Активация ключа для vl FaceSDK	26
4.5 Активация ключей Guardant	28
4.5.1 Активация лицензий	29
4.6 Обновление ключа Guardant	32
4.6.1 Online-обновление ключа	32
4.6.2 Offline-обновление ключа	32
4.7 Активация ключа для sb FaceSDK	33
5 ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ	35
5.1 Авторизация в сервисах Nomad и Consul	35
5.2 Требования к настройкам сети	37
5.3 Рекомендации по сохранению исходных файлов системы	39
5.4 Добавление мобильного приложения на стартовую страницу	40
5.5 Добавление нового узла в кластер	42
5.6 Перенос сервисов	42
5.7 Выделение хоста под определенные сервисы	43
5.8 Удаление системы	44



5.9 Восстановление кластера.....	45
6 УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ.....	46
6.1 Закрытие незавершенных сессий GCC.....	46
7 ОБНОВЛЕНИЕ СИСТЕМЫ.....	47
7.1 Обновление с версии 4.8.....	47
7.2 Обновление с версии 4.9 - 4.12.....	50
7.3 Обновление с версии 4.13 / 4.14.....	54
7.4 Обновление с версии 4.15 и выше.....	58
7.5 Дополнительная информация.....	60
7.5.1 Обновление БД сервиса events-service	60
7.5.2 Настройка сервиса Events Service	61



1 ВВЕДЕНИЕ

1.1 Общие положения

Данное руководство предназначено для администраторов программного комплекса системы видеопериферии **Визирь.СКУД** (далее – **Визирь.СКУД**) и содержит сведения о разворачивании компонентов **Визирь.СКУД**.

Настоящее руководство не заменяет учебную, справочную литературу и руководства от производителей операционной системы и общего программного обеспечения.

1.2 Соглашения и обозначения

В руководстве приняты следующие типографские соглашения:

Формат	Описание
Обычный	Основной текст документа.
<i>Курсив</i>	Применяется для выделения первого появления <i>термина</i> , значение которого поясняется здесь же или даётся в приложении. Также применяется для оформления <i>примечаний</i> .
Полужирный	Применяется для написания наименований программных компонентов , наименований управляющих и информационных элементов интерфейса (заголовки , кнопки и т.п.), а также компонентов системы (Центральный сервер , Подсистема обработки видеопотоков и т.п.). Также применяется для привлечения внимания .
<i>Полужирный курсив</i>	Применяется для написания имен файлов и путей доступа к ним.
<u>Подчёркнутый синий</u>	Применяется для обозначения <u>внешних ссылок</u> , <u>ссылок на разделы</u> текущего документа, а также <u>адресов URL-запросов</u> для работы с API.

Словосочетание «выбрать, выделить, нажать объект (или нажать на объект)» означает: «навести указатель манипулятора типа «мышь» на объект, и нажать кнопку манипулятора».

Выбор меню, который показан при помощи стрелки **>**, например, текст **Файл > Выход**, должен пониматься так: выбрать меню **Файл**, затем команду **Выход** из меню **Файл**.

Ниже приведены примеры оформления материала руководства, указывающие на важность сведений.



Ссылки на другие документы в основном тексте.



Примечания; важные сведения; указания на действия, которые необходимо выполнить в обязательном порядке.



Требования, несоблюдение которых может привести к некорректной работе, повреждению или выходу из строя изделий или программного обеспечения.

1.3 Правообладатель

Правообладателем **Программного комплекса системы видеоидентификации «Визирь»** является ООО «ЦРТ».

Все остальные названия компаний и названия продуктов, упомянутые в документе, являются собственностью их соответствующих владельцев.

Ни одна из частей этого документа не подлежит воспроизведению, передаче, хранению в поисковой системе или переводу на какой-либо язык в любой форме, любыми средствами без письменного разрешения общества с ограниченной ответственностью «Центр речевых технологий».



2 ОБЩИЕ СВЕДЕНИЯ

2.1 Сведения о программе и изготовителе

Наименование: Программный комплекс системы видеоидентификации физических лиц
Визирь.СКУД

Условное наименование: STC-S857

Обозначение: ЦВАУ.01026-01

Изготовитель: Общество с ограниченной ответственностью «Центр речевых технологий»

Адрес: 194044, г. Санкт-Петербург, Выборгская наб., д. 45, лит. Е

Телефон: +7 (812) 325-88-48

Факс: +7 (812) 327-92-97

2.2 Техническая поддержка

Адрес службы сервисного обслуживания и технической поддержки в интернете:

Email: st-support@speechpro.com

Веб-сайт: <http://www.speechpro.ru/>

При обращении в службу технической поддержки необходимо предоставить подробное описание возникшей проблемы.

Предварительно подготовьте следующую информацию:

- тип компьютера и сведения об его конфигурации;
- название и версия используемой операционной системы;
- log-файлы.



3 УСТАНОВКА ПРОГРАММЫ



Минимальные требования к аппаратным ресурсам (приведенные требования являются ориентировочными и применимы к тестовым установкам системы с целью ознакомления с ее функционалом). Требования указаны из расчета подключения не более 2 камер:

- Оперативная память – 48 ГБ (нижний предел – 32 ГБ);
- Процессор – не менее 8 ядер **Intel Core i7 / Xeon Silver** с частотой не менее 2,2 ГГц;
- SSD накопитель – не менее 150 ГБ.



Поддерживаемые операционные системы для установки **Визирь.СКУД** – **Ubuntu 22.04, Astra Linux 1.7.5-1.7.7** (с отключенным режимом ЗПС).



Перед установкой **Визирь.СКУД** при необходимости уточните у поставщика тип используемого компонента **FaceSDK**. От этого будет зависеть порядок активации лицензий.

3.1 Установка в ОС Ubuntu

Перед установкой

1. Перед установкой системы **обязательно** убедитесь, что ваша сеть имеет корректную конфигурацию (см. [Требования к настройкам сети](#)).



Обратите внимание!

- **Docker** по умолчанию использует сеть **172.17.0.0/16**;
- В настройках заданий (job) всех сервисов системы в **Nomad** по умолчанию указан DNS-сервер **172.17.0.1**.

Таким образом, для корректной работы системы **не используйте** адреса хостов из диапазона **172.17.0.0/16**.

2. Если на компьютерах (далее – хостах), куда будет устанавливаться **Визирь.СКУД**, настроен доступ через прокси-сервер – произведите следующие настройки в файле **environment** (отредактируйте файл на **каждом** хосте, куда будет устанавливаться **Визирь.СКУД**):

- Откройте файл **environment**, расположенный в каталоге **/etc/**;
- Добавьте в файл следующие параметры:

```
no_proxy=127.0.0.1,localhost,Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP,*.consul  
NO_PROXY=127.0.0.1,localhost,Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP,*.consul
```

где **Host_1_Name_or_IP, Host_2_Name_or_IP, Host_N_Name_or_IP** – имена или IP-адреса хостов, на которые будет устанавливаться **Визирь.СКУД**.

- Переподключитесь к системе.



3. Распакуйте дистрибутив **Визирь.СКУД** в любой каталог. При распределенной установке на несколько хостов – распакуйте дистрибутив в любой каталог на одном из хостов. Для распаковки используйте хост, на котором будет устанавливаться компонент **Nexus**.



Рекомендуемый каталог для распаковки дистрибутива системы указан в разделе [Рекомендации по сохранению исходных файлов системы](#).



После установки системы Визирь.СКУД не удаляйте каталог с распакованным дистрибутивом. Данный каталог может быть использован в дальнейшем для настройки системы и установки дополнительных сервисов.

4. Настройте лицензирование. Для этого:

– При использовании ключей **Guardant**:

– На хосте, который будет использоваться для активации лицензий, установите сервис **Guardant Control Center** и активируйте лицензии (см. [Активация ключей Guardant](#)).

– На хосте, с которого будет производиться установка **Визирь.СКУД**:

- В распакованном дистрибутиве **Визирь.СКУД** перейдите в каталог **`./ansible/inventory/group_vars`**;
- Откройте файл **`all.yml`** с помощью текстового редактора;
- В файле **`all.yml`** в параметре **`guardant_license_server`** укажите имена или IP-адреса хостов, которые будут использоваться для активации лицензий.

– При использовании ключей **HASP**:

– На хосте, который будет использоваться для активации лицензий:

- Перейдите в распакованный дистрибутив **Визирь.СКУД**;
- Выполните следующие команды:

```
tar xf hasp.tar.gz
cd aksusbd-8.11.1 && sudo ./dinst && cd ..
cp HaspTool /usr/bin/
sudo chmod +x /usr/bin/HaspTool
sudo sh -c 'echo ACCremote = 1 >> /etc/hasplm/hasplm.ini'
sudo service hasplmd restart
```

– На хосте, с которого будет производиться установка **Визирь.СКУД**:

- В распакованном дистрибутиве **Визирь.СКУД** перейдите в каталог **`./ansible/roles/hasp/files`**;
- Откройте файл **`hasplm.ini`** с помощью текстового редактора;
- В файле **`hasplm.ini`** в блоке настроек **REMOTE** укажите имена или IP-адреса хостов, которые будут использоваться для активации лицензий (параметр **`serveraddr`**, см. рис. ниже).



```
[REMOTE]
broadcastsearch = 1
aggressive = 0
serversearchinterval = 30
serveraddr = Hostname_1
serveraddr = Hostname_2
```

Рисунок 1 – Файл *hasplm.ini*

– Активируйте лицензии (см. [АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ](#)).

Установка

Для установки **Визирь.СКУД** на компьютер под управлением ОС **Ubuntu** выполните следующие действия:

1. Установите **sshpas**. При этом:

– При **наличии** доступа в интернет используйте следующую команду:

```
sudo apt install -y sshpass
```

– При **отсутствии** доступа в интернет используйте следующую команду:

```
sudo dpkg -i offline/ubuntu/*.deb
```

2. Установите **ansible** и **pip**:

```
sudo bash ansible.sh
```

3. В файле **hosts** (после распаковки дистрибутива файл расположен в каталоге **/ansible/inventory**) укажите имена или IP-адреса хостов, на которые необходимо установить сервисы **Визирь.СКУД**. Заполните следующие блоки настроек (в каждом блоке также укажите нужные имя пользователя и пароль для доступа по ssh). Если доступ по ssh производится не по стандартному порту 22, укажите используемый порт ssh в каждой строке, где указываются имена или IP-адреса хостов, на которые необходимо установить сервисы **Визирь.СКУД**, например, **Host_name:3222 ansible_ssh_user=root ansible_ssh_pass=P2sssw0rd**, где **3222** – используемый порт ssh:

– **[sdk]** – укажите имена или IP-адреса хостов, на которые необходимо установить компонент **FaceSDK**;

– **[nomad_servers]** – укажите имена или IP-адреса хостов, на которых необходимо развернуть кластер **Nomad** и **Consul**. Данные хосты будут отвечать за работоспособность (отказоустойчивость) кластера системы;

– **[nomad_clients]** – укажите имена или IP-адреса хостов, на которых необходимо запустить основные сервисы платформенной части **Визирь.СКУД**. Сервисы **Визирь.СКУД** будут автоматически распределены по указанным хостам. **Обратите внимание!** Обязательно укажите в блоке **[nomad_clients]** имена или IP-адреса хостов, на которые будут установлены **Minio** и **Postgres** (см. блоки **[minio]**, **[postgres]** ниже). При необходимости также укажите в блоке **[nomad_clients]** имена или IP-адреса хостов с GPU (хосты, на которые будет установлен плагин **nomad-device-nvidia**, см. блок **[gpu]** ниже);



Для группировки хостов (для последующего выделения хостов под определенные сервисы, см. [Выделение хоста под определенные сервисы](#)) задайте для нужных хостов произвольное название пула в параметре **node_pool**.



Обратите внимание:

- Хост, указанный в блоке **[nomad_clients]** – это хост, на котором могут быть запущены сервисы системы. Таким образом, если хост **не был указан** в блоке **[nomad_clients]**, то на нем **не будет запущен ни один сервис системы**, даже если этот хост указан в блоке **[nomad_servers]**;
- Один и тот же хост может выполнять роль сервера / клиента или обе роли одновременно, то есть может быть указан в одном из блоков **[nomad_clients]** / **[nomad_servers]** или в обоих блоках;
- Рекомендации по заполнению блока **[nomad_servers]**:
 - Если в кластере меньше 3 хостов – укажите в блоке **[nomad_servers]** 1 сервер;
 - Если в кластере меньше 5 хостов – укажите в блоке **[nomad_servers]** 3 сервера. В этом случае кластер продолжит работу, если один из указанных хостов выйдет из строя;
 - Если в кластере больше 5 хостов – укажите в блоке **[nomad_servers]** 5 серверов. В этом случае кластер продолжит работу, если 2 из указанных хостов выйдут из строя.

- **[rts]** – укажите имена или IP-адреса хостов, на которые необходимо установить компонент **RTS**;
- **[srts]** – укажите имена или IP-адреса хостов, на которые необходимо установить компонент **SRTS**;



Обратите внимание:

- Возможна одновременная работа системы только с **одним** из сервисов **RTS** / **SRTS** (не указывайте хосты одновременно в группах **[rts]** и **[srts]**);
- При установке системы с **SDK SB** возможна установка **только** компонента **SRTS** (блок **[srts]**); при установке системы с **SDK VL** – возможна установка **только** компонента **RTS** (блок **[rts]**);
- Для каждого хоста в переменной **host_with_gpu** укажите **True** или **False** в зависимости от наличия или отсутствия на хосте видеокарты. На хосты с установленной переменной **host_with_gpu=True** будет установлен плагин **nvidia-toolkit**.

- **[minio]** – укажите имя или IP-адрес хоста, на который необходимо установить файловое хранилище **Minio**. **Не устанавливайте Minio на несколько хостов**;



- **[postgres]** – укажите имя или IP-адрес хоста, на который необходимо установить **Postgres**.
Не устанавливайте Postgres на несколько хостов;
- **[gpu]** – укажите имена или IP-адреса хостов с видеокартой (GPU). На данные хосты будет установлен **nvidia-toolkit**, а также плагин **nomad-device-nvidia** (если хосты с GPU указаны в блоке **[nomad_clients]**);
- **[nexus]** – укажите имя или IP-адрес хоста, на который необходимо установить **Nexus** (укажите хост, где был распакован дистрибутив).

**Обратите внимание:**

- Блок настроек **[platform]** является системным и **не предназначен для редактирования**;
- **Не допускается** запуск **ansible playbook** без заполнения **хотя бы одного** основного блока настроек в файле **hosts** (см. шаг 3);
- **Не допускается** указание в качестве имени / адреса хоста значений **localhost** или **127.0.0.1**.

4. При **отсутствии** доступа в интернет установите значение параметра **offline_install: "true"** в файле **all.yml**, расположенном в каталоге **./ansible/inventory/group_vars**.

5. Запустите **ansible playbook**. При этом:

- Используйте ключ **-K** для предоставления текущему пользователю root-прав для запуска команды (для возможности использования повышения привилегий **sudo**);
- Используйте ключ **-k**, чтобы не указывать в файле **hosts** пароль пользователя (см. предыдущий шаг).

```
cd ansible/  
ansible-playbook -i inventory/hosts run.yml -K
```

**Обратите внимание:**

- При возникновении ошибок **No cluster leader** восстановите кластер **Nomad / Consul** согласно разделу [Восстановление кластера](#). После восстановления кластера продолжите установку системы;
- При запуске **ansible playbook** происходит автоматическая проверка соответствия версии ОС требованиям для установки. Для отключения данной проверки установите значение параметра **os_check=false** в файле **all.yml**, расположенном в каталоге **./ansible/inventory/group_vars**.



Обратите внимание! Добавление новых хостов в кластер через **ansible playbook** **невозможно**. Таким образом, если в процессе первичной установки системы вы указали не все нужные хосты для установки – полностью завершите процесс первичной установки и затем добавьте хосты вручную (см. [Добавление нового узла в кластер](#)).



6. Дождитесь завершения процесса установки. После завершения установки будет отображен информационный блок **Play recap** (рис. 2).

```
PLAY RECAP *****
localhost : ok=15   changed=6    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
Host_1 : ok=106  changed=69   unreachable=0    failed=0    skipped=37   rescued=0    ignored=2
Host_2 : ok=60   changed=44   unreachable=0    failed=0    skipped=11   rescued=0    ignored=2
Host_3 : ok=57   changed=41   unreachable=0    failed=0    skipped=14   rescued=0    ignored=2
```

Рисунок 2 – Информационный блок **Play recap** после завершения установки

После установки

После установки **Визирь.СКУД**:

1. Если на хостах, где была установлена система, настроен доступ через прокси-сервер – выполните следующие действия на **каждом** хосте кластера системы:

- В каталоге **/etc/systemd/system/** создайте подкаталог **docker.service.d**:

```
sudo mkdir -p /etc/systemd/system/docker.service.d/
```

- В созданном каталоге **/etc/systemd/system/docker.service.d** создайте файл **http-proxy.conf**, содержащий следующий текст:

```
Environment="NO_PROXY=127.0.0.1,localhost,Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP,*.consul"
```

где **Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP** – имена или IP-адреса хостов, на которых установлена система.

- Перезапустите **docker**:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

2. Авторизуйтесь в web-интерфейсах **Nomad** и **Consul** (см. [Авторизация в сервисах Nomad и Consul](#)).

3. При использовании ключей **HASP**:

- Откройте конфигурационный файл **licensetype.env**, расположенный в каталоге **configuration-service** web-интерфейса **Consul**;

- В файле **licensetype.env** исправьте содержимое на **hasp**. Не допускайте переноса строки в файле. После сохранения изменений в файле **licensetype.env** все сервисы системы будут автоматически перезапущены.

4. Для корректной работы в web-интерфейсе **обязательно отключите режим энергосбережения** в настройках браузера согласно официальной документации на соответствующий браузер (актуально для устройств, работающих от батарей):

- Для **Google Chrome** – согласно <https://support.google.com/chrome/answer/12929150?hl=en#zippy=%2Cturn-energy-saver-on-or-off>;

- Для **Yandex Browser** – согласно <https://yandex.ru/support/browser/ru/economy-mode/energy-saving.html>.



При количестве ядер процессора сервера более **196** необходимо ограничить количество используемых хранилищем **Minio** ядер значением **не более 196**. Для этого:

- В web-интерфейсе **Nomad** перейдите на вкладку **Jobs** и выберите задание **minio** из списка;
- Перейдите на вкладку **Definition**, установите переключатель **Job Spec / Full Definition** в положение **Full Definition** и нажмите кнопку **Edit**;
- В блок параметров **Env** добавьте параметр **GOMAXPROCS** со значением не более **196** (например, "**GOMAXPROCS**": "**196**", где **196** – количество ядер процессора, которое будет использовать **Minio**).



В составе дистрибутива системы поставляются сервисные скрипты, позволяющие выполнять дополнительные действия (например, перезапускать сервисы, удалять задания **Nomad** и т.д.). Скрипты и текстовые файлы с их описанием расположены в каталоге **/opt/frs/scripts**).

3.2 Установка в ОС Astra Linux

Перед установкой

Перед установкой необходимо настроить репозитории ОС **Astra Linux** на **всех хостах**. Для этого:

1. При **наличии** доступа в интернет воспользуйтесь одним из следующих способов:
 - Используйте репозиторий с диска ОС **Astra Linux** (рекомендуемый способ при установке на **один** хост):
 - Убедитесь, что в системе есть смонтированный диск, содержащий образ ОС **Astra Linux**.
 - Проверьте, что в файле **/etc/apt/sources.list** есть запись следующего вида:

deb cdrom:[OS Astra Linux 1.7.5 1.7_x86-64 DVD]/ 1.7_x86-64 contrib main non-free

- Настройте локальные репозитории согласно официальной документации <https://wiki.astralinux.ru/pages/viewpage.action?pageId=61575159> (рекомендуемый способ при установке на **несколько** хостов)
- Настройте сетевые репозитории согласно официальной документации <https://wiki.astralinux.ru/pages/viewpage.action?pageId=158598882>.

2. При **отсутствии** доступа в интернет воспользуйтесь одним из следующих способов:

- Используйте репозиторий с диска ОС **Astra Linux** (рекомендуемый способ при установке на **один** хост):
 - Убедитесь, что в системе есть смонтированный диск, содержащий образ ОС **Astra Linux**.
 - Проверьте, что в файле **/etc/apt/sources.list** есть запись следующего вида:

deb cdrom:[OS Astra Linux 1.7.5 1.7_x86-64 DVD]/ 1.7_x86-64 contrib main non-free



- Настройте локальные репозитории согласно официальной документации <https://wiki.astralinux.ru/pages/viewpage.action?pageId=61575159> (рекомендуемый способ при установке на **несколько** хостов).

Также перед установкой:

1. **Обязательно** убедитесь, что ваша сеть имеет корректную конфигурацию (см. [Требования к настройкам сети](#)).



Обратите внимание!

- **Docker** по умолчанию использует сеть **172.17.0.0/16**;
- В настройках заданий (job) всех сервисов системы в **Nomad** по умолчанию указан DNS-сервер **172.17.0.1**.

Таким образом, для корректной работы системы **не используйте** адреса хостов из диапазона **172.17.0.0/16**.

2. Убедитесь, что в файле **/etc/hosts** отсутствует (или закомментирована) строка в формате **127.0.1.1 Имя_хоста**. При наличии указанной строки закомментируйте / удалите ее, или замените адрес **127.0.1.1** на IP-адрес хоста.



Обратите внимание! При установке системы с SDK VL в ОС Astra Linux 1.7.6 / 1.7.7, на всех хостах, куда устанавливается система, **обязательно**:

1. В файле **/etc/default/grub** в строку **GRUB_CMDLINE_LINUX_DEFAULT** добавьте параметр **parsec.execstack=1**:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet net.ifnames=0 parsec.mac=0 parsec.execstack=1"
```

2. Выполните следующую команду:

```
update-grub
```

3. Перезагрузите хосты.

3. Если на компьютерах (далее – хостах), куда будет устанавливаться **Визирь.СКУД**, настроен доступ через прокси-сервер – произведите следующие настройки в файле **environment** (отредактируйте файл на **каждом** хосте, куда будет устанавливаться **Визирь.СКУД**):

- Откройте файл **environment**, расположенный в каталоге **/etc/**;
- Добавьте в файл следующие параметры:

```
no_proxy=127.0.0.1,localhost,Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP,*.consul  
NO_PROXY=127.0.0.1,localhost,Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP,*.consul
```

где **Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP** – имена или IP-адреса хостов, на которые будет устанавливаться **Визирь.СКУД**.

- Переподключитесь к системе.

4. На всех хостах, где будет производиться установка, отключите режим ЗПС с помощью следующих команд:



```
sudo astra-digsig-control disable  
sudo reboot
```

5. Распакуйте дистрибутив **Визирь.СКУД** в любой каталог. При распределенной установке на несколько хостов – распакуйте дистрибутив в любой каталог на одном из хостов. Для распаковки используйте хост, на котором будет устанавливаться компонент **Nexus**.



Рекомендуемый каталог для распаковки дистрибутива системы указан в разделе [Рекомендации по сохранению исходных файлов системы](#).



После установки системы Визирь.СКУД не удаляйте каталог с распакованным дистрибутивом. Данный каталог может быть использован в дальнейшем для настройки системы и установки дополнительных сервисов.

6. Настройте лицензирование. Для этого:

– При использовании ключей **Guardant**:

– На хосте, который будет использоваться для активации лицензий, установите сервис **Guardant Control Center** и активируйте лицензии (см. [Активация ключей Guardant](#)).

– На хосте, с которого будет производиться установка **Визирь.СКУД**:

- В распакованном дистрибутиве **Визирь.СКУД** перейдите в каталог **`./ansible/inventory/group_vars`**;
- Откройте файл **`all.yml`** с помощью текстового редактора;
- В файле **`all.yml`** в параметре **`guardant_license_server`** укажите имена или IP-адреса хостов, которые будут использоваться для активации лицензий.

– При использовании ключей **HASP**:

– На хосте, который будет использоваться для активации лицензий:

- Перейдите в распакованный дистрибутив **Визирь.СКУД**;
- Выполните следующие команды:

```
tar xf hasp.tar.gz  
cd aksusbd-8.11.1 && sudo ./dinst && cd ..  
cp HaspTool /usr/bin/  
sudo chmod +x /usr/bin/HaspTool  
sudo sh -c 'echo ACCremote = 1 >> /etc/hasplm/hasplm.ini'  
sudo service hasplmd restart
```

– На хосте, с которого будет производиться установка **Визирь.СКУД**:

- В распакованном дистрибутиве **Визирь.СКУД** перейдите в каталог **`./ansible/roles/hasp/files/`**;
- Откройте файл **`hasplm.ini`** с помощью текстового редактора;



- В файле **hasplm.ini** в блоке настроек **REMOTE** укажите имена или IP-адреса хостов, которые будут использоваться для активации лицензий (параметр **serveraddr**, см. рис. ниже).

```
[REMOTE]
broadcastsearch = 1
aggressive = 0
serversearchinterval = 30
serveraddr = Hostname_1
serveraddr = Hostname_2
```

Рисунок 3 – Файл **hasplm.ini**

- Активируйте лицензии (см. [АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ](#)).

Установка

Для установки **Визирь.СКУД** на компьютер под управлением ОС **Astra Linux** выполните следующие действия:

1. Установите **python** и **sshpas**:

```
sudo apt update
sudo apt install python3.7 python3-distutils curl sshpass
```



Обратите внимание:

- На хосте, с которого производится установка, **обязательно** должен быть установлен **python** версии **3.7**;
- На всех остальных хостах кластера **обязательно** должен быть установлен **python** версии **2.7** или **3.5 – 3.11**.
- Для проверки установки **python** на хосте выполните следующую команду:
 - Для версии **python 3.*** используйте команду **python3 --version**;
 - Для версии **python 2.*** используйте команду **python --version**.

Для установки **python** на хосты кластера:

- Убедитесь, что на **всех хостах** настроены репозитории ОС **Astra Linux** (см. п. **Перед установкой** выше);
- Выполните следующие команды:

```
sudo apt update
sudo apt install -y python3.7
```

2. Установите **ansible** и **pip**:

```
sudo bash ansible.sh
```

3. В файле **hosts** (после распаковки дистрибутива файл расположен в каталоге **/ansible/inventory**) укажите имена или IP-адреса хостов, на которые необходимо установить сервисы **Визирь.СКУД**. Заполните следующие блоки настроек (в каждом блоке также укажите нужные имя пользователя и пароль для доступа по ssh). Если доступ по ssh производится не



по стандартному порту 22, укажите используемый порт `ssh` в каждой строке, где указываются имена или IP-адреса хостов, на которые необходимо установить сервисы **Визирь.СКУД**, например, `Host_name:3222 ansible_ssh_user=root ansible_ssh_pass=P2sssw0rd`, где **3222** – используемый порт `ssh`:

- **[sdk]** – укажите имена или IP-адреса хостов, на которые необходимо установить компонент **FaceSDK**;
- **[nomad_servers]** – укажите имена или IP-адреса хостов, на которых необходимо развернуть кластер **Nomad** и **Consul**. Данные хосты будут отвечать за работоспособность (отказоустойчивость) кластера системы;
- **[nomad_clients]** – укажите имена или IP-адреса хостов, на которых необходимо запустить основные сервисы платформенной части **Визирь.СКУД**. Сервисы **Визирь.СКУД** будут автоматически распределены по указанным хостам. **Обратите внимание!** Обязательно укажите в блоке **[nomad_clients]** имена или IP-адреса хостов, на которые будут установлены **Minio** и **Postgres**, а также хостов с GPU (см. блоки **[minio]**, **[postgres]** и **[gpu]** ниже);



Для группировки хостов (для последующего выделения хостов под определенные сервисы, см. [Выделение хоста под определенные сервисы](#)) задайте для нужных хостов произвольное название пула в параметре **node_pool**.



Обратите внимание:

- Хост, указанный в блоке **[nomad_clients]** – это хост, на котором могут быть запущены сервисы системы. Таким образом, если хост **не был указан** в блоке **[nomad_clients]**, то на нем **не будет запущен ни один сервис системы**, даже если этот хост указан в блоке **[nomad_servers]**;
- Один и тот же хост может выполнять роль сервера / клиента или обе роли одновременно, то есть может быть указан в одном из блоков **[nomad_clients]** / **[nomad_servers]** или в обоих блоках;
- Рекомендации по заполнению блока **[nomad_servers]**:
 - Если в кластере меньше 3 хостов – укажите в блоке **[nomad_servers]** 1 сервер;
 - Если в кластере меньше 5 хостов – укажите в блоке **[nomad_servers]** 3 сервера. В этом случае кластер продолжит работу, если один из указанных хостов выйдет из строя;
 - Если в кластере больше 5 хостов – укажите в блоке **[nomad_servers]** 5 серверов. В этом случае кластер продолжит работу, если 2 из указанных хостов выйдут из строя.

- **[rts]** – укажите имена или IP-адреса хостов, на которые необходимо установить компонент **RTS**;



- **[srts]** – укажите имена или IP-адреса хостов, на которые необходимо установить компонент **SRTS**;



Обратите внимание:

- Возможна одновременная работа системы только с **одним** из сервисов **RTS / SRTS** (не указывайте хосты одновременно в группах **[rts]** и **[srts]**);
- При установке системы с **SDK SB** возможна установка **только** компонента **SRTS** (блок **[srts]**); при установке системы с **SDK VL** – возможна установка **только** компонента **RTS** (блок **[rts]**);
- Для каждого хоста в переменной **host_with_gpu** укажите **True** или **False** в зависимости от наличия или отсутствия на хосте видеокарты. На хосты с установленной переменной **host_with_gpu=True** будет установлен плагин **nvidia-toolkit**.

- **[minio]** – укажите имя или IP-адрес хоста, на который необходимо установить файловое хранилище **Minio**. **Не устанавливайте Minio на несколько хостов**;
- **[postgres]** – укажите имя или IP-адрес хоста, на который необходимо установить **Postgres**. **Не устанавливайте Postgres на несколько хостов**;
- **[gpu]** – укажите имена или IP-адреса хостов с видеокартой (GPU). На данные хосты будет установлен плагин **nomad-device-nvidia** и **nvidia-toolkit**;
- **[nexus]** – укажите имя или IP-адрес хоста, на который необходимо установить **Nexus** (укажите хост, где был распакован дистрибутив на шаге 3).



Обратите внимание:

- Блок настроек **[platform]** является системным и **не предназначен для редактирования**;
- **Не допускается** запуск **ansible playbook** без заполнения **хотя бы одного** основного блока настроек в файле **hosts** (см. шаг 3);
- **Не допускается** указание в качестве имени / адреса хоста значений **localhost** или **127.0.0.1**.

4. Запустите **ansible playbook**. При этом:

- Используйте ключ **-K** для предоставления текущему пользователю root-прав для запуска команды (для возможности использования повышения привилегий **sudo**);
- Используйте ключ **-k**, чтобы не указывать в файле **hosts** пароль пользователя (см. предыдущий шаг).

```
cd ansible/  
sudo ansible-playbook -i inventory/hosts run.yml -K
```



Обратите внимание:

- При возникновении ошибок **No cluster leader** восстановите кластер **Nomad / Consul** согласно разделу [Восстановление кластера](#). После восстановления кластера продолжите установку системы;
- При запуске **ansible playbook** происходит автоматическая проверка соответствия версии ОС требованиям для установки. Для отключения данной проверки установите значение параметра **os_check=false** в файле **all.yml**, расположенном в каталоге **./ansible/inventory/group_vars**.



Обратите внимание! Добавление новых хостов в кластер через **ansible playbook** **невозможно**. Таким образом, если в процессе первичной установки системы вы указали не все нужные хосты для установки – полностью завершите процесс первичной установки и затем добавьте хосты вручную (см. [Добавление нового узла в кластер](#)).

5. Дождитесь завершения процесса установки. После завершения установки будет отображен информационный блок **Play recap** (рис. 4).

```
PLAY RECAP *****
localhost : ok=15  changed=6  unreachable=0  failed=0  skipped=5  rescued=0  ignored=0
Host_1 : ok=106  changed=69  unreachable=0  failed=0  skipped=37  rescued=0  ignored=2
Host_2 : ok=60  changed=44  unreachable=0  failed=0  skipped=11  rescued=0  ignored=2
Host_3 : ok=57  changed=41  unreachable=0  failed=0  skipped=14  rescued=0  ignored=2
```

Рисунок 4 – Информационный блок **Play recap** после завершения установки

После установки

После установки **Визирь.СКУД**:

1. Если на хостах, где была установлена система, настроен доступ через прокси-сервер – выполните следующие действия на **каждом** хосте кластера системы:

- В каталоге **/etc/systemd/system/** создайте подкаталог **docker.service.d**:

```
sudo mkdir -p /etc/systemd/system/docker.service.d/
```

- В созданном каталоге **/etc/systemd/system/docker.service.d** создайте файл **http-proxy.conf**, содержащий следующий текст:

```
Environment="NO_PROXY=127.0.0.1,localhost,Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP,*.consul"
```

где **Host_1_Name_or_IP,Host_2_Name_or_IP,Host_N_Name_or_IP** – имена или IP-адреса хостов, на которых установлена система.

- Перезапустите **docker**:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

2. В конфигурационном файле **/etc/dnsmasq.conf** и (при наличии) в файле **/etc/dnsmasq.d/20-owndns** укажите корректные DNS-серверы вашей сети в параметрах **server** и перезапустите сервис **dnsmasq** с помощью команды **service dnsmasq restart**;



3. Авторизуйтесь в web-интерфейсах **Nomad** и **Consul** (см. [Авторизация в сервисах Nomad и Consul](#));
4. При использовании ключей **HASP**:
 - Откройте конфигурационный файл **licensetype.env**, расположенный в каталоге **configuration-service** web-интерфейса **Consul**;
 - В файле **licensetype.env** исправьте содержимое на **hasp**. Не допускайте переноса строки в файле. После сохранения изменений в файле **licensetype.env** все сервисы системы будут автоматически перезапущены.
5. Для корректной работы в web-интерфейсе **обязательно отключите режим энергосбережения** в настройках браузера согласно официальной документации на соответствующий браузер (актуально для устройств, работающих от батареи):
 - Для **Google Chrome** – согласно <https://support.google.com/chrome/answer/12929150?hl=en#zippy=%2Cturn-energy-saver-on-or-off>;
 - Для **Yandex Browser** – согласно <https://yandex.ru/support/browser/ru/economy-mode/energy-saving.html>.



В составе дистрибутива системы поставляются сервисные скрипты, позволяющие выполнять дополнительные действия (например, перезапускать сервисы, удалять задания **Nomad** и т.д.). Скрипты и текстовые файлы с их описанием расположены в каталоге **/opt/frs/scripts**).



При количестве ядер процессора сервера более **196** необходимо ограничить количество используемых хранилищем **Minio** ядер значением **не более 196**. Для этого:

- В web-интерфейсе **Nomad** перейдите на вкладку **Jobs** и выберите задание **minio** из списка;
- Перейдите на вкладку **Definition**, установите переключатель **Job Spec / Full Definition** в положение **Full Definition** и нажмите кнопку **Edit**;
- В блок параметров **Env** добавьте параметр **GOMAXPROCS** со значением не более **196** (например, **"GOMAXPROCS": "196"**, где **196** – количество ядер процессора, которое будет использовать **Minio**).



4 АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ

Возможны следующие сценарии активации ключей:

- При наличии файла лицензии **.v2c** активируйте лицензии согласно разделу [Активация ключа \(при наличии файла .v2c\)](#);
- При отсутствии файла лицензии **.v2c** активируйте лицензии согласно разделу [Активация ключа \(при отсутствии файла .v2c\)](#);
- При необходимости обновления ключа защиты выполните действия согласно разделу [Обновление ключа защиты](#);
- При работе **Визирь.СКУД** на базе компонента **vl FaceSDK** установите **дополнительный** ключ защиты согласно разделу [Активация ключа для vl FaceSDK](#);
- При работе **Визирь.СКУД** на базе компонента **sb FaceSDK** установите **дополнительный** ключ защиты согласно разделу [Активация ключа для sb FaceSDK](#);
- При работе с ключами **Guardant** активируйте лицензии согласно разделу [Активация ключей Guardant](#).

4.1 Активация ключа (при наличии файла .v2c)

При наличии файла лицензии **.v2c** (может быть получен у поставщика системы) активируйте ключи защиты следующим образом:

1. Перейдите по адресу <http://servername.domain:1947/>.
2. Перейдите во вкладку **Update/Attach** и активируйте файл лицензии **.v2c** (см. рис ниже).

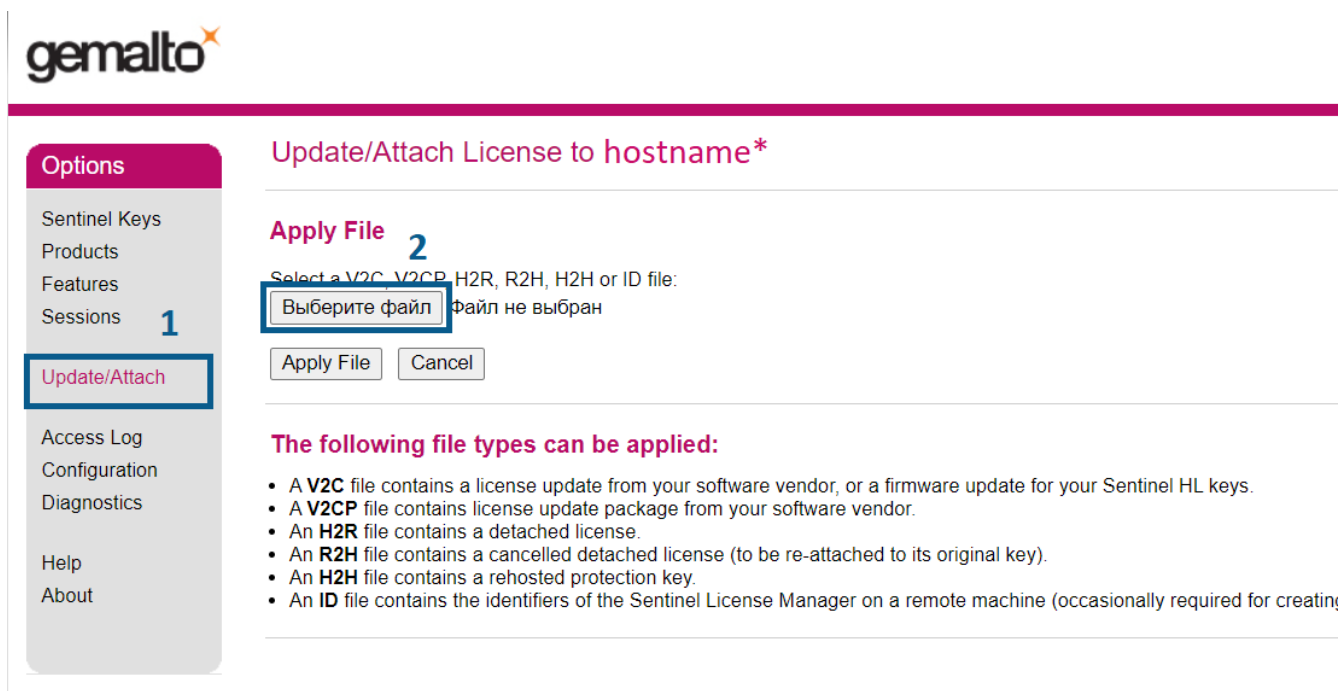


Рисунок 5 – Активация лицензии

3. После добавления лицензии перезапустите сервисы:



- Перезапустите сервисы системы с помощью скрипта **nomad_restart_services**;
- Перезапустите сервис **RTS** с помощью команды **sudo service rts restart**.

4.2 Активация ключа (при отсутствии файла .v2c)

При отсутствии файла лицензии **.v2c** активируйте ключи защиты следующим образом:

1. Снимите слепок (сгенерируйте файл **.c2v**). Для этого:



Перед снятием слепка убедитесь, что служба **/etc/init.d/aksusbd** имеет статус **Работает**. Для этого выполните команду **/etc/init.d/aksusbd status**. Отсутствие данной службы означает, что на компьютере не установлен драйвер **HASP**. Установка драйвера **HASP** описана в разделе [УСТАНОВКА ПРОГРАММЫ](#).

- Выполните команду **HaspTool**:

HaspTool

- Выполните команду **f**, чтобы подготовить слепок системы;
- Введите полное имя **.c2v** файла, который будет сгенерирован, например:

filename.c2v

- Для выхода из утилиты **HaspTool** выполните команду **e** и нажмите клавишу **Enter**.

2. Получите файл лицензии **.v2c** одним из следующих способов:

- При отсутствии данных о **Product Key** передайте сгенерированный на шаге 1 файл **.c2v** сотруднику технической поддержки или менеджеру поставщика системы;
- При наличии данных о **Product Key** выполните следующие действия:
 - Перейдите по ссылке <http://activation.speechpro.com:8080/ems/customerLogin.html>;
 - Введите полученный **Product/Licence Key** в открывшемся диалогом окне (рис. 6) и нажмите **Login**;

Sentinel EMS
ENTITLEMENT MANAGEMENT SYSTEM 7.8

Product/License Key Login

Product/License Key :

Login

Рисунок 6 – Окно ввода Product Key

- При запросе персональных данных можно пропустить данный пункт, нажав кнопку **Register Later** (рис. 7).



Register to activate the Product key. Already registered? [click here](#) Register Later

Customer Information

* First Name: Middle Name:
 * Last Name: * E-mail:
 Locale:
 Telephone: Fax:
 CRM ID: Ref ID:
 Description:

Billing Details

Street:
 City: ZIP/Postal Code:
 State: Country:
☒ Shipping details are same as billing details

Рисунок 7 – Окно ввода персональных данных

– Нажмите кнопку **Offline Activation** в правом верхнем углу (рис. 8);

Product/License Key : 634ce76f-c123-77f5-33e3-c43ff76b8d55 Online Activation Offline Activation

Download and install the Latest RTE Installer by clicking here [EXE](#).

Product/License Key:	634ce76f-c123-77f5-33e3-c43ff76b8d55		
Customer Name:	Alex	E-mail:	
Channel Partner:	-	E-mail:	-
Activations:	1	Remaining Activations:	1
Previous Activations:	0	Enabled:	true
Enforcement:	Sentinel LDK		

Products	Lock Type
FaceSDK	HL or SL (AdminMode or UserMode)

Рисунок 8 – Кнопка **Offline Activation**

– Загрузите сгенерированный на шаге 1 файл **.c2v**. Для этого в строке **Upload C2V** нажмите кнопку (рис. 9).

– Нажмите кнопку **Generate** (рис. 9).



Generate License

Order Details

Product/License Key: 634ce76f-c123-77f5-33e3-c43ff76b8d55

Customer: Alex Email:

Activations: 1 Remaining Activations: 1

Ref ID 1: Ref ID 2:

Entitlement

Comments:

Products:

Product	Lock Type
FaceSDK	HL or SL (AdminMode or UserMode)

Download **RUS**, a tool to generate C2V ➡

Upload C2V

Upload C2V:

Comments:

Рисунок 9 – Загрузка файла **.c2v**

- Автоматически будет сгенерирован **.v2c**-файл. Нажмите кнопку **Download V2C File**, чтобы сохранить его на свой компьютер (рис. 10).



Generate License ? ✕

V2C generated successfully

Order Details

Product Key: [REDACTED]
 Customer: Internal E-mail: -
 Activations: 1 Remaining Activations: 1
 Ref ID 1: Ref ID 2:
 Entitlement Comments: SIS2 for descri...
 Products:

Product	Lock Type
SIS2_HL_SL	HL or SL (AdminMode or UserMode)

[Download **RUS**, a tool to generate C2V →](#)

Activation Details

Key ID	Key Type	Activation Date	Comments
773007258381348572	SL-AdminMode	2013-06-17	Download V2C File

Рисунок 10 – Загрузка .v2c-файла на компьютер

- После получения .v2c-файла активируйте лицензию согласно алгоритму, описанному в разделе [Активация ключа \(при наличии файла .v2c\)](#).

4.3 Обновление ключа защиты

Для обновления ключа защиты:

- Снимите слепок (сгенерируйте файл .c2v). Для этого:



Перед снятием слепка убедитесь, что служба **/etc/init.d/aksusbd** имеет статус **Работает**. Для этого выполните команду **/etc/init.d/aksusbd status**. Отсутствие данной службы означает, что на компьютере не установлен драйвер **HASP**. Установка драйвера **HASP** описана в разделе [УСТАНОВКА ПРОГРАММЫ](#).

- Выполните команду **HaspTool**:

HaspTool

- Выполните команду **i**, чтобы получить информацию о **HASP ID** уже установленного ключа;
- Выполните команду **c** для сохранения данных;
- Введите полное имя .c2v файла, который будет сгенерирован, например:

filename.c2v

- Для выхода из утилиты **HaspTool** выполните команду **e** и нажмите клавишу **Enter**.

- Передайте сгенерированный на шаге 1 файл .c2v сотруднику технической поддержки или менеджеру поставщика системы. В ответ вы получите файл лицензии .v2c.
- После получения .v2c-файла активируйте лицензию согласно алгоритму, описанному в разделе [Активация ключа \(при наличии файла .v2c\)](#).



4.4 Активация ключа для vl FaceSDK



Начиная с версии **vl FaceSDK 5.3** (версия 5.4 является исключением) лицензия привязывается не к аппаратному обеспечению, а к сгенерированному слепку (**Fingerprint**), который сохраняется в файл. Таким образом, если данный файл будет утерян или удален, потребуются заново проходить процедуру активации ключа (генерировать новый **Fingerprint** с использованием еще одной лицензии).

При работе **Визирь.СКУД** на базе компонента **vl FaceSDK** установите дополнительный ключ защиты. Для этого:

1. После установки **Визирь.СКУД** на базе компонента **vl FaceSDK** перейдите в каталог **/opt/stc/SDK/FaceSDK/bin/data** и откройте текстовый файл **license.conf**.
2. В файле **license.conf** (рис. 11):
 - В поле **EID** укажите полученный от поставщика **EID**;
 - В поле **ProductID** укажите полученный от поставщика **ProductID**;
 - В поле **VendorData** укажите полученный от поставщика **VendorData**;
 - Также обязательно убедитесь, что все остальные параметры имеют значения, соответствующие приведенным на рисунке ниже;
 - Сохраните изменения в файле.

```
<param name="Server" type="Value::String" text="https://license.visionlabs.ai/" />
<param name="EID" type="Value::String" text=" " />
<param name="ProductID" type="Value::String" text=" " />
<param name="Filename" type="Value::String" text="license.dat" />
<param name="ActivationMode" type="Value::String" text="Hardware" />
<param name="Storage" type="Value::String" text="/opt/stc/SDK/FaceSDK/SDK_data/license_storage.dat" />
<param name="ContainerMode" type="Value::Int1" x="1" />
<!-- 0 - first variant, 1 - second variant -->
<param name="FingerprintSource" type="Value::Int1" x="2" />
<param name="ConnectionTimeout" type="Value::Int1" x="15" />
<param name="ServerRetriesCount" type="Value::Int1" x="3" />
<param name="VendorData" type="Value::String" text=" " />
```

Рисунок 11 – Файл **license.conf**

3. Запустите исполняемый файл **FingerprintViewer** (расположен в каталоге **/opt/stc/SDK/FaceSDK/bin**):

```
cd /opt/stc/SDK/FaceSDK/bin
./FingerprintViewer
```

4. Сохраните полученный в результате выполнения команды ID (выделенная область на рис. 12). Он потребуется в дальнейшем.



5. После выполнения команды в директории `/opt/stc/SDK/FaceSDK/SDK_data/` будет автоматически создан файл **license_storage.dat**. Если данный файл будет утерян или удален, потребуется заново проходить процедуру активации ключа.

```
[root@... /]# cd /opt/stc/SDK/FaceSDK/bin/
[root@... bin]# ./FingerprintViewer
Using license file: ./data/license.conf
To run with default license file path:
    ./FingerprintViewer
If you want to specify license file:
    ./FingerprintViewer <path to license.conf>
[26.09.2022 10:52:22] [Debug] [createLicense] creation of the FitLicense license!
Fingerprint for the current device:
RnRtZgEAAACi8t8RPXLYIIPDfP8PleDR
```

Рисунок 12 – Результат запуска исполняемого файла **FingerprintViewer**

6. Скопируйте измененный на шаге 2 файл **license.conf** в директорию `/opt/stc/SDK/FaceSDK/SDK_data/`.
7. На компьютере с доступом в интернет перейдите по ссылке <http://visionlabsbv.prod.sentinelcloud.com:80/ems/customerLogin.html>.
8. Авторизуйтесь в системе по полученному **EID** (рис. 13).

Customer Login

EID

4a587dcv-s33b-7eg1-7ne3-5b675bf4565d

Login

Рисунок 13 – Авторизация

9. В открывшемся окне нажмите кнопку **Register Later**.
10. В открывшемся окне нажмите кнопку **Activate**.
11. В открывшемся окне (рис. 14):
- Укажите **e-mail**, произвольное **имя устройства** (имя позволит идентифицировать устройство, на котором будет активирована лицензия) и выберите нужный **часовой пояс** в соответствующих полях;
 - В поле **Device FingerPrint** укажите ID, полученный с помощью командной строки на шагах 3-4.
 - Нажмите кнопку **Activate**.



Рисунок 14 – Активация лицензии

12. Скачайте сгенерированный файл **licenseFile.v2c** и сохраните его в директорию **/opt/stc/SDK/FaceSDK/SDK_data/**.

13. Переименуйте сохраненный файл **licenseFile.v2c** в **license.dat**:

```
cd /opt/stc/SDK/FaceSDK/SDK_data/
mv licenseFile.v2c license.dat
```

14. Перезапустите сервисы:

- Перезапустите сервисы системы с помощью скрипта **nomad_restart_services**;
- Перезапустите сервис **RTS** с помощью команды **sudo service rts restart**.



При установке **Визирь.СКУД** на базе компонента **vl FaceSDK** обязательно сохраняйте в надежном месте следующие данные:

- Файл **license.conf**;
- Файл **license.dat**;
- Файл **license_storage.dat**.

4.5 Активация ключей Guardant



Для работы с ключами **Guardant** используется сервис **Guardant Control Center (GCC)** и утилита **license_wizard**.



4.5.1 Активация лицензий

Возможны 2 сценария активации лицензий:

- **Online** – если на компьютере, где развернута система, есть доступ в интернет (см. [Online-активация лицензий](#));
- **Offline** – если на компьютере, где развернута система, отсутствует доступ в интернет (см. [Offline-активация лицензий](#)).

4.5.1.1 Online-активация лицензий

Для активации лицензий:

1. Запросите ключ для активации у менеджера продукта.
2. На хосте, который будет использоваться для активации лицензий, установите **Guardant Control Center (GCC)**. Для этого:
 - Загрузите установочный пакет. Установочный пакет доступен по ссылке **http://nexus_host:8081/repository/raw/grdcontrol.deb**, где **nexus_host** – адрес сервера, где установлен **Nexus**;
 - Установите **Guardant Control Center** с помощью следующей команды:

```
dpkg -i grdcontrol.deb
```

При установке **GCC** утилита **license_wizard** будет установлена **автоматически**.

После установки web интерфейс **GCC** будет доступен по адресу **<http://hostname:3189>** (рис. 15).

По умолчанию пароль для входа в раздел **Настройки** – **admin**.

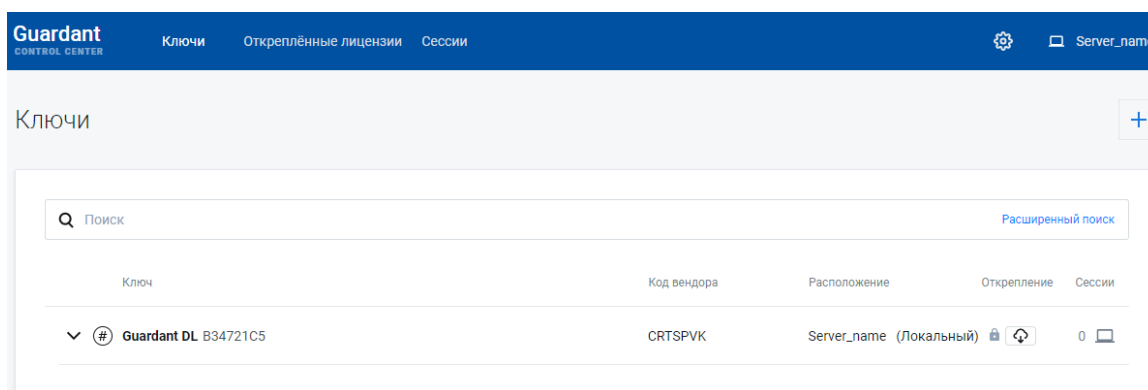


Рисунок 15 – Web-интерфейс **Guardant Control Center**

3. Активируйте запрошенный на шаге 1 ключ, выполнив команду в следующем формате:
license_wizard --console --activate key_id --host https://guardant_server:guardant_port
где:

- **key_id** – ключ активации, запрошенный на шаге 1;
- **guardant_server** – адрес сервера лицензирования **Guardant** (используйте адрес **<https://guardant-api.speechpro.com>**);



– **guardant_port** – порт сервера лицензирования **Guardant** (используйте порт **443**);

Пример выполнения команды:

```
license_wizard --console \  
--activate ztuHjX-3u7etA-1eweti-1t3zsA-Cg6C7R \  
--host https://guardant-api.speechpro.com:443
```

При успешной активации вывод результата выполнения команды будет содержать фразу **License activation completed.**



Итоговый файл с лицензией будет сохранен в каталоге **/var/guardant/DL/Licenses**.



Обратите внимание! При перезагрузке сервиса **Guardant Control Center (GCC)** могут оставаться незавершенные сессии. В этом случае требуется завершить их вручную (см. [Закрытие незавершенных сессий GCC](#)).

4.5.1.2 Offline-активация лицензий

Для активации лицензий:

1. Запросите ключ для активации у менеджера продукта.
2. На хосте, входящем в кластер **Визирь.СКУД**, установите **Guardant Control Center (GCC)**. Для этого:
 - Загрузите установочный пакет. Установочный пакет доступен по ссылке **http://nexus_host:8081/repository/raw/grdcontrol.deb**, где **nexus_host** – адрес сервера, где установлен **Nexus**;
 - Установите **Guardant Control Center** с помощью следующей команды:

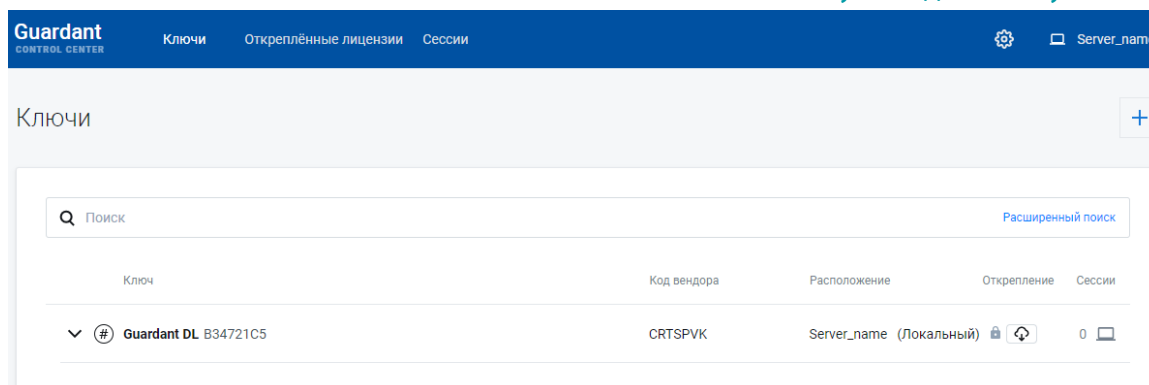
```
dpkg -i grdcontrol.deb
```

3. Перенесите скачанный установочный пакет **grdcontrol.deb** на хост с доступом в интернет (хост, который будет использоваться для запроса лицензии).
4. Установите **Guardant Control Center** на хосте с доступом в интернет с помощью следующей команды:

```
dpkg -i grdcontrol.deb
```

При установке **GCC** утилита **license_wizard** будет установлена **автоматически**.

После установки web интерфейс **GCC** будет доступен по адресу **http://hostname:3189** (рис. 16). По умолчанию пароль для входа в раздел **Настройки** – **admin**.

Рисунок 16 – Web-интерфейс **Guardant Control Center**

5. Снимите «слепок» (fingerprint) системы, выполнив команду в следующем формате:

license_wizard --console --activate-request fingerprint_file

где **fingerprint_file** – полный путь к файлу слепка, который будет сгенерирован, например, **/opt/stc/fingerprint**.

6. Перенесите сгенерированный файл слепка на компьютер с доступом в интернет.

7. На компьютере с доступом в интернет сгенерируйте файл лицензии, выполнив команду в следующем формате:

license_wizard --console --activate-response key_id fingerprint_file key_file --host https://guardant_server:guardant_port

где:

- **key_id** – ключ активации, запрошенный на шаге 1;
- **fingerprint_file** – полный путь к файлу слепка, сгенерированного на шаге 4;
- **key_file** – полный путь к файлу лицензии, который будет сгенерирован, например, **/opt/stc/licensefile**;
- **guardant_server** – адрес сервера лицензирования **Guardant** (используйте адрес **https://guardant-api.speechpro.com**);
- **guardant_port** – порт сервера лицензирования **Guardant** (используйте порт **443**);

Пример выполнения команды:

```
license_wizard --console --activate-response ztuHjX-3u7etA-1eweti-1t3zsA-Cg6C7R /opt/stc/fingerprint /opt/stc/licensefile --host https://guardant-api.speechpro.com:443
```

8. Перенесите сгенерированный файл лицензии на исходный компьютер, где его необходимо активировать.

9. Активируйте файл лицензии:

```
license_wizard --console --activate-offline key_file
```

где **key_file** – полный путь к файлу лицензии, который необходимо активировать, например, **/opt/stc/licensefile**;



Обратите внимание! При перезагрузке сервиса **Guardant Control Center (GCC)** могут оставаться незавершенные сессии. В этом случае требуется завершить их вручную (см. [Заккрытие незавершенных сессий GCC](#)).

4.6 Обновление ключа Guardant

Возможны 2 сценария обновления ключа:

- **Online** – если на компьютере, где развернута система, есть доступ в интернет (см. [Online-обновление ключа](#));
- **Offline** – если на компьютере, где развернута система, отсутствует доступ в интернет (см. [Offline-обновление ключа](#)).

4.6.1 Online-обновление ключа

Для обновления ключа:

1. Для просмотра списка установленных ключей выполните следующую команду:

```
$ license_wizard --console --list
```

Установленные ключи будут отображены в выводе команды в строке **Guardant DL Net** (в примере ниже ключ **CEEE71DD**):

```
Guardant DL Net #CEEE71DD
FaceSDK (Net) (no MAC address) (209)
FaceSDK (85)
No restrictions
Work station, 50
```

2. Обновите ключ, выполнив команду в следующем формате:

```
license_wizard --console --update license_id --host https://guardant_server:guardant_port
```

где:

- **license_id** – установленный ключ, полученный на шаге 1;
- **guardant_server** – адрес сервера лицензирования **Guardant** (используйте адрес **https://guardant-api.speechpro.com**);
- **guardant_port** – порт сервера лицензирования **Guardant** (используйте порт **443**);

Пример выполнения команды:

```
license_wizard --console --update CEEE71DD --host https://guardant-api.speechpro.com:443
```

При успешной активации вывод результата выполнения команды будет содержать фразу **License update completed**.

4.6.2 Offline-обновление ключа

Для обновления ключа:

1. Для просмотра списка установленных ключей выполните следующую команду:

```
$ license_wizard --console --list
```



Установленные ключи будут отображены в выводе команды в строке **Guardant DL Net** (в примере ниже ключ **CEEE71DD**):

```
Guardant DL Net #CEEE71DD
FaceSDK (Net) (no MAC address) (209)
FaceSDK (85)
No restrictions
Work station, 50
```

2. Создайте файл, содержащий информацию о текущей лицензии, выполнив команду в следующем формате:

```
$ license_wizard --console --update-request #CEEE71DD file.request
```

где **file.request** – имя файла, который будет сгенерирован в текущем каталоге.

3. Перенесите сгенерированный на предыдущем шаге файл на компьютер с доступом в интернет.

4. На компьютере с доступом в интернет и установленным сервисом **GCC** выполните команду в следующем формате:

```
license_wizard --console --update-response request_file_path license_file_path --host
https://guardant_server:guardant_port
```

где:

- **request_file_path** – путь к файлу с информацией о текущей лицензии (**file.request**), сгенерированному на шаге 2;
- **license_file_path** – путь к файлу лицензии, который будет сгенерирован;
- **guardant_server** – адрес сервера лицензирования **Guardant** (используйте адрес **https://guardant-api.speechpro.com**);
- **guardant_port** – порт сервера лицензирования **Guardant** (используйте порт **443**);

Пример выполнения команды:

```
license_wizard --console --update-response file.request licensefile --host
https://guardant_server:guardant_port
```

5. Перенесите сгенерированный файл лицензии на исходный компьютер, где его необходимо активировать.

6. Активируйте файл лицензии, выполнив команду в следующем формате:

```
license_wizard --console --update-offline license_file_path
```

Пример выполнения команды:

```
license_wizard --console --update-offline licensefile
```

При успешной активации вывод результата выполнения команды будет содержать фразу **License update completed**.

4.7 Активация ключа для sb FaceSDK

При работе **Визирь.СКУД** на базе компонента **sb FaceSDK**:



1. Установите дополнительный ключ защиты. Установка ключа производится аналогично описанной в разделе [Активация ключей Guardant](#). При этом используется адрес сервера лицензирования **https://guardant-face-api.speechpro.com:443**.

Пример выполнения команды активации лицензии:

```
license_wizard --console --activate 3u7etA-z9yizh-kX3Cgh-jYr7rX-utuRjX \  
--host https://guardant-face-api.speechpro.com:443
```

2. Перейдите в каталог **/opt/stc/SDK/FaceSDK/SDK_data/**.
3. В файле **sblicense.json** в параметре **hostName** укажите адрес хоста, который использовался для активации лицензий (адрес хоста, где был установлен **Guardant Control Center**). Адрес хоста необходимо указывать в **двойных кавычках**. Пример указания адреса:

```
"hostName": ["10.10.10.10"]
```



5 ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

5.1 Авторизация в сервисах Nomad и Consul

Для авторизации в web-интерфейсах **Nomad** и **Consul**:

1. Откройте файл **acl.yml**. После установки **Визирь.СКУД** файл доступен в каталоге **/ansible/inventory/group_vars**.
2. Скопируйте в буфер обмена токены для авторизации в **Nomad** и **Consul** (параметры **nomad_token** и **consul_token**, см. рис. ниже).

```
[root@Host_1 ansible]# cat inventory/group_vars/acl.yml
---
nomad_token: 'f5dd14fe-ac0f-8cdc-a91c-ae83ce104d9c'
consul_token: '703ed906-7d8a-3e65-2831-ef5f5fc4ae5e'
```

Рисунок 17 – Файл **acl.yml**

3. Перейдите в web-интерфейс **Nomad**. Для этого введите в адресной строке браузера адрес в формате **Адрес_или_Имя_компьютера:4646** (например, **10.10.10.10:4646**).
4. В открывшемся окне нажмите кнопку **Sign In** (рис. 18).

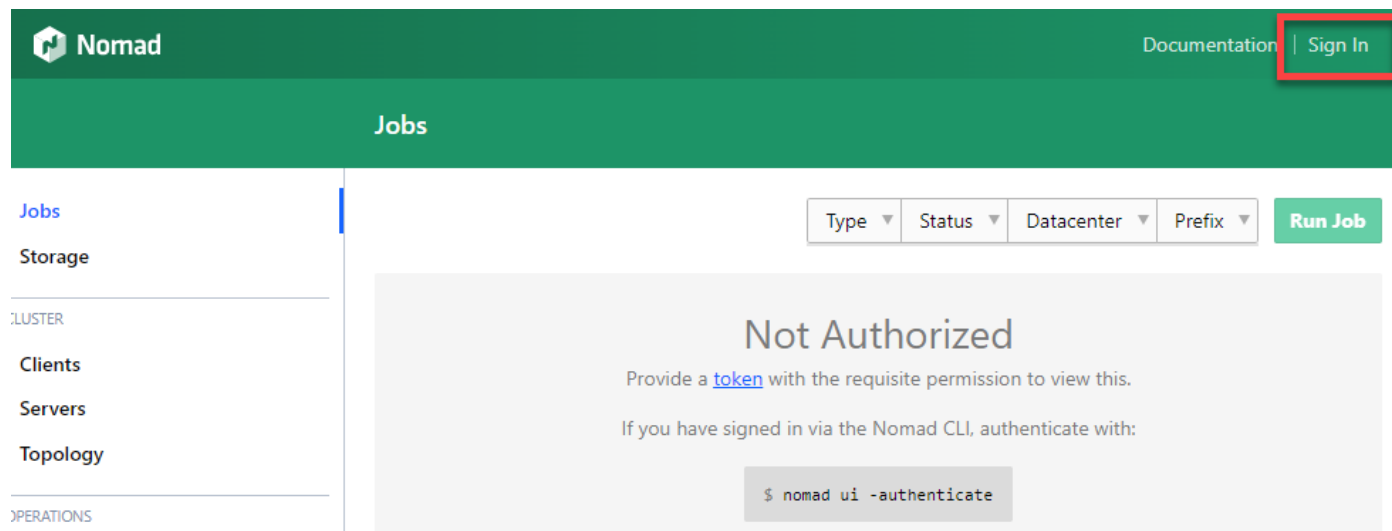


Рисунок 18 – Авторизация в web-интерфейсе **Nomad**

5. В открывшемся окне введите в поле **Secret ID** токен, скопированный из файла **acl.yml** (параметр **nomad_token**), и нажмите кнопку **Set Token** (рис. 19).

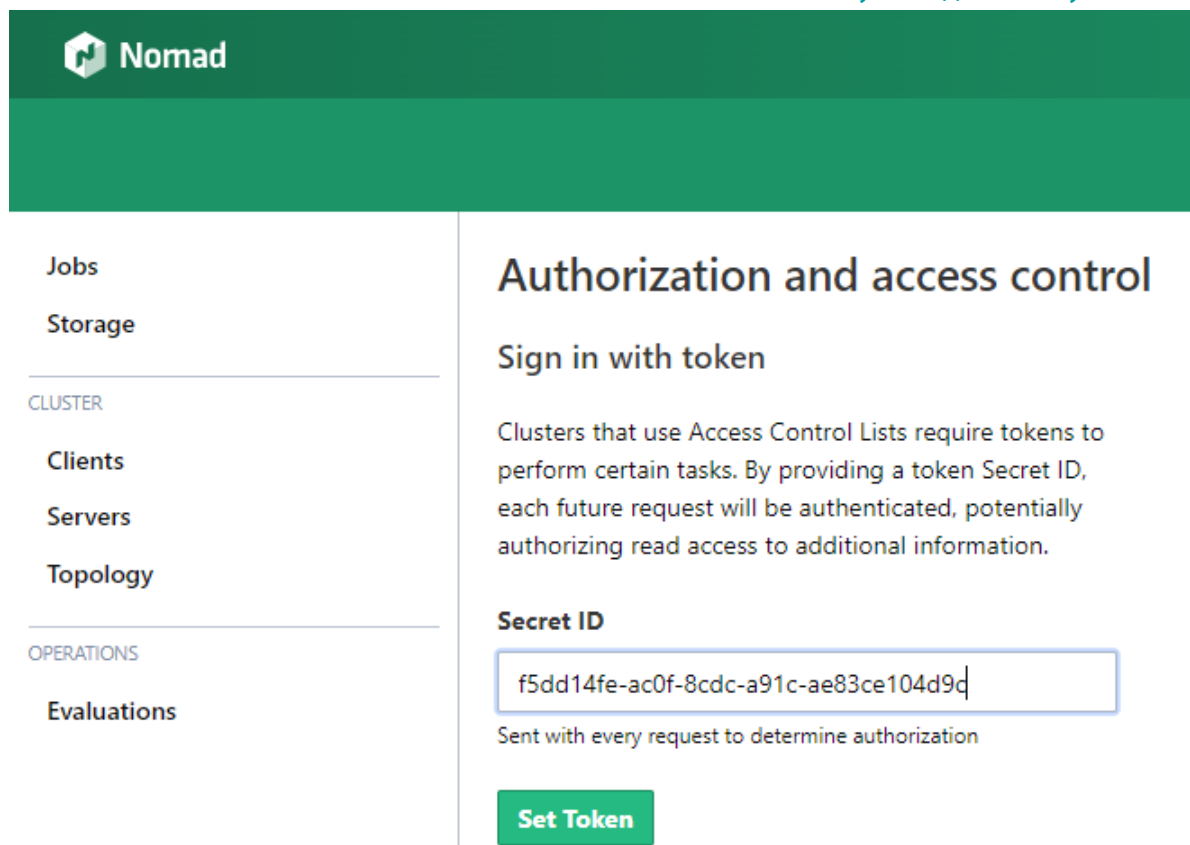


Рисунок 19 – Авторизация в web-интерфейсе **Nomad**. Ввод токена

6. Перейдите в web-интерфейс **Consul**. Для этого в адресной строке браузера введите адрес в формате **Адрес_или_Имя_компьютера:8500** (например, **10.10.10.10:8500**).
7. В открывшемся окне нажмите кнопку **Log In** (рис. 20).

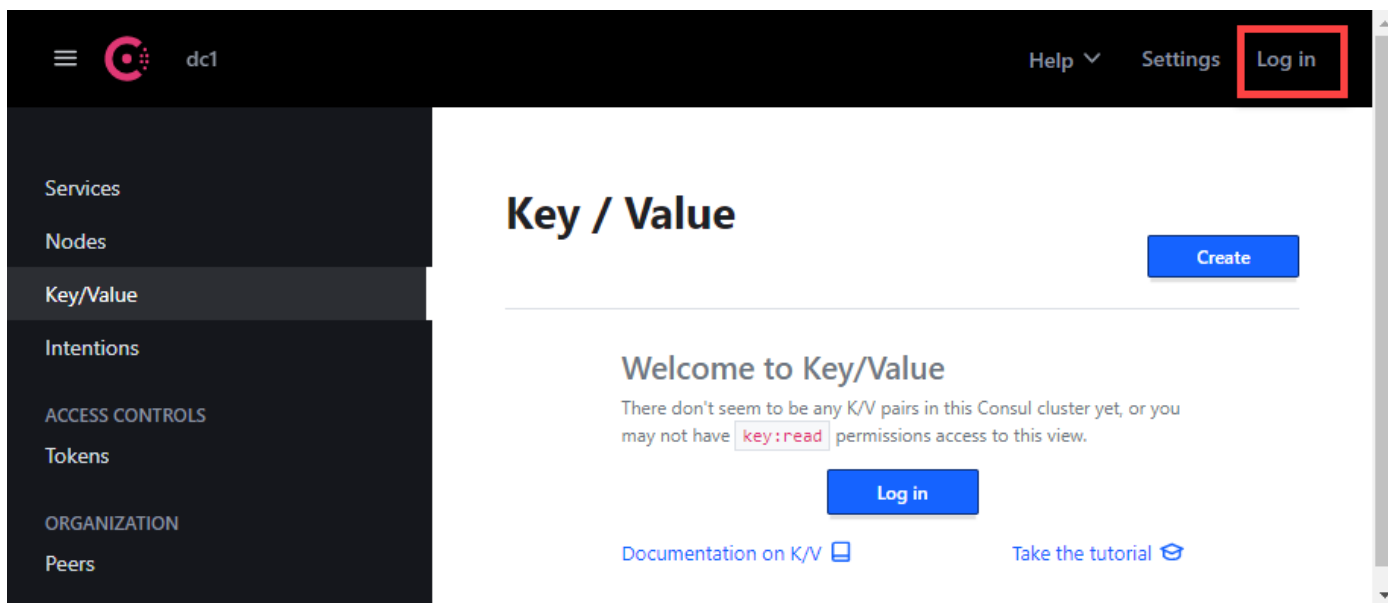


Рисунок 20 – Авторизация в web-интерфейсе **Consul**

8. В открывшемся окне введите токен, скопированный из файла **acl.yml** (параметр **consul_token**), и нажмите кнопку **Log In** (рис. 21).



Рисунок 21 – Авторизация в web-интерфейсе **Consul**. Ввод токена

5.2 Требования к настройкам сети



Обратите внимание:

- Не используйте несколько сетевых интерфейсов;
- В системе обязательно должен быть настроен маршрут по умолчанию (**default route**). При наличии нескольких сетевых интерфейсов, **Визирь.СКУД** будет доступен на интерфейсе, указанном в маршруте по умолчанию;
- Не изменяйте сетевой адрес в течение всего срока работы системы;
- IP-адреса всех хостов кластера должны быть доступны для рабочих мест администраторов / операторов системы;
- При необходимости смены адресов или доступа к нескольким сетям (контурам) используйте отдельный маршрутизатор или устройство, которое может выполнять его функции;
- Протокол IPv6 **не поддерживается**. Таким образом, после установки системы необходимо убедиться, что имя хоста, на котором установлен компонент **RTS**, имеет IP-адрес в формате протокола IPv4. Для этого выполните команду **ping имя_хоста**. Если имя хоста имеет IP-адрес в формате протокола IPv4, то результат выполнения команды будет содержать строку в формате **PING server_name (10.10.1.10)**. Если вывод команды содержит адрес в формате с **двоеточиями** или содержит ошибку **Name or service not known** – добавьте в файл **/etc/hosts** строку в формате **ip имя_хоста** (где **имя_хоста** – имя хоста с установленным компонентом **RTS**).



В процессе установки **Визирь.СКУД** (см. [Установка в ОС Ubuntu](#) / [Установка в ОС Astra Linux](#)) настройки DNS в системе будут приведены к следующему виду:

- Для перенаправления запросов к сервису **Consul** и DNS-серверам вашей сети в системе на каждом узле (хосте) кластера будет установлен DNS-сервер **dnsmasq** (актуально для ОС **Astra Linux**);
- В настройках DNS-сервера **dnsmasq** будут указаны DNS-серверы вашей сети (актуально для ОС **Astra Linux**);
- В настройках сети операционной системы будет указан DNS-сервер **127.0.0.1**.

Таким образом, для корректной установки **Визирь.СКУД**, до начала установки убедитесь, что сеть в ОС сконфигурирована **одним из** следующих способов:

1. За настройки сети отвечает компонент ОС **NetworkManager**:

- В ОС **присутствует и включен** компонент **NetworkManager**. Для проверки наличия **NetworkManager** выполните команду **sudo systemctl status NetworkManager**. При наличии **NetworkManager** вывод команды будет содержать строку **Active: active (running)**.
- В ОС **присутствует** утилита **nmcli**. Для проверки наличия утилиты выполните команду **nmcli -v**. При наличии утилиты в системе вывод команды будет содержать текущую версию утилиты;
- Вывод команды **nmcli dev status** содержит список сетевых интерфейсов;

2. Настройки сети указаны вручную в файле **/etc/network/interfaces**. В этом случае, в зависимости от типа настроек сети (статические или динамические адреса):

- Если указан **статический** IP адрес:
 - В ОС **отсутствует или отключен** компонент **NetworkManager**;
 - В каталоге **/etc/network/** присутствует файл **interfaces**;
 - В файле **interfaces** присутствует строка **inet static**;
 - В файле **/etc/resolv.conf** указаны корректные DNS-серверы вашей сети.
- Если настройки вашей сети **динамически** назначаются DHCP-сервером:
 - В ОС **отсутствует или отключен** компонент **NetworkManager**;
 - В каталоге **/etc/network/** присутствует файл **interfaces**;
 - В файле **interfaces** присутствует строка **inet dhcp**;
 - В файле **/etc/resolv.conf** указаны корректные DNS-серверы вашей сети.

Также обязательно убедитесь, что:



- После перезагрузки сети (команда **service networking restart**) в файле **/etc/resolv.conf** указан **только** адрес **127.0.0.1**;
- В системе настроен маршрут по умолчанию (**default route**). Для проверки выполните команду **ip route**. Вывод команды должен содержать строку **default via**.



Не рекомендуется использовать несколько сетевых интерфейсов. В этом случае может потребоваться дополнительная настройка для обеспечения доступности системы с необходимых сетевых интерфейсов.

При наличии в системе нескольких сетевых интерфейсов:

- Настройте **все** сетевые интерфейсы таким образом, чтобы для каждого из них в качестве DNS-сервера был указан адрес **127.0.0.1**;
- После настройки сетевых интерфейсов перезагрузите сеть ОС (команда **service networking restart**);
- Убедитесь, что после перезагрузки сети в файле **/etc/resolv.conf** указан только адрес **127.0.0.1**.

5.3 Рекомендации по сохранению исходных файлов системы

Во избежание потери важных системных данных рекомендуется сохранять исходные файлы системы в следующих каталогах:

- Распакованный архив дистрибутива (при первичной установке системы) – в каталоге **/opt/frs/distr/product_name/product_version** (где **product_name** – имя устанавливаемого продукта, **product_version** – версия продукта);
- Распакованный архив дистрибутива при обновлении системы – в каталоге **/opt/frs/distr/product_name/product_version**;
- Распакованный архив с сервисами интеграции – в каталоге **/opt/frs/distr/integration_service_name/product_version** (где **integration_service_name** – имя устанавливаемого сервиса интеграции или группы сервисов);
- Рекомендуемый каталог для хранения резервных копий БД – **/backup/**.

При этом, следующие данные будут скопированы в соответствующие каталоги **автоматически** после установки системы:

1. В каталог **/opt/frs/installs/дата_время/** будут скопированы:

- Лог-файлы процесса установки системы (файлы **ansible/ansible.log**);
- Файл **ansible/inventory/hosts**;
- Каталог **jobs** из **ansible/jobs**;



- Архив с диагностическими данными выполнения скрипта **get_info.sh**.
- 2. В каталог **/opt/frs/scripts** будут скопированы файлы скриптов из каталога **scripts**.
- 3. В каталог **/opt/frs/scripts/recovery** будут скопированы **.yaml**-файлы, предназначенные для удаления и восстановления системы (см. [Удаление системы](#) и [Восстановление кластера](#)).

5.4 Добавление мобильного приложения на стартовую страницу

Для обеспечения возможности скачивания мобильного приложения со стартовой страницы системы выполните следующие действия:

1. Перейдите по адресу в формате **Server_IP_or_Name:8081/#browse/browse:raw**, где **Server_IP_or_Name** – адрес сервера, где установлен Nexus.
2. Авторизуйтесь – для этого нажмите кнопку **Sign in** в правом верхнем углу экрана (по умолчанию используются логин – **admin**; пароль – **admin**).
3. **Обновите страницу браузера.**
4. Находясь в репозитории **raw** (**Browse > raw**), нажмите кнопку **Upload component**.
5. В открывшемся окне (рис. 22):
 - Нажмите кнопку **Browse** и выберите установочный файл мобильного приложения, который будет добавлен на стартовую страницу системы;
 - В параметре **Directory** укажите имя каталога, в который необходимо добавить файл.

Choose assets for this component

File	Filename
C:\fakepath\vizirClient_4.22	vizirClient_4.22.0.557.apk

 **Browse**

 **Add another asset**

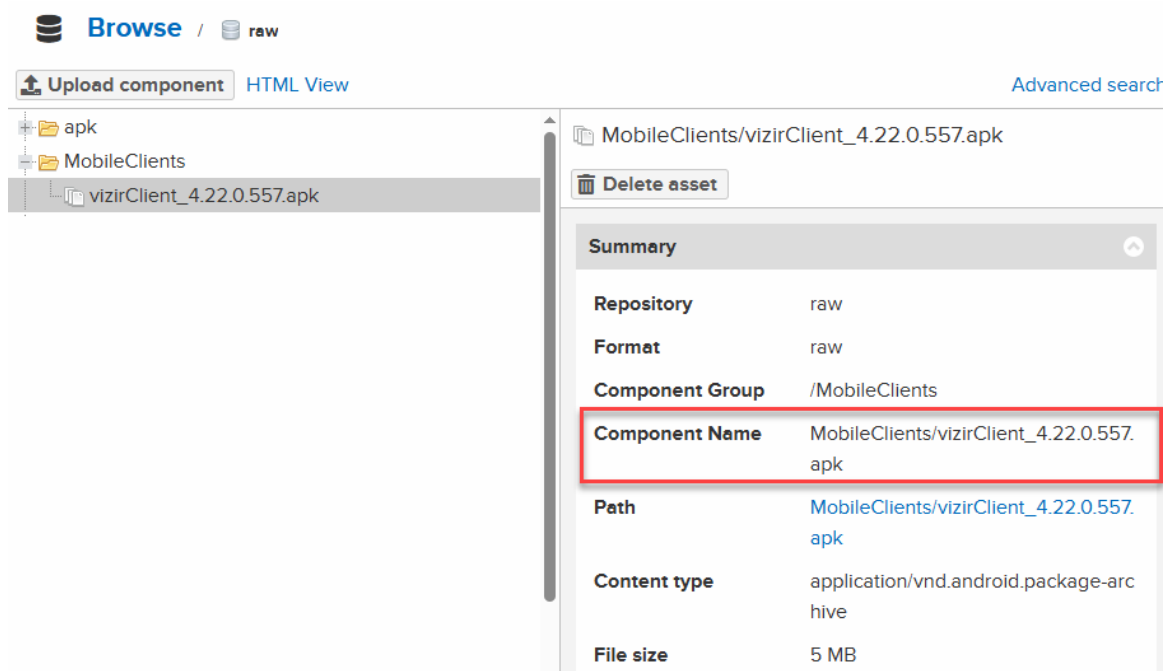
Component attributes

Directory:
Destination for uploaded files (e.g. /path/to/files/)

Upload **Cancel**

Рисунок 22 – Загрузка файла

6. Перейдите в созданный каталог, выберите загруженный файл и скопируйте значение параметра **Component name** (рис. 23).

Рисунок 23 – Параметр **Component name**

7. В конфигурационном файле **appsettings.json** сервиса **serviceslanding**, расположенном в web-интерфейсе **Consul**:

- Укажите путь к загруженному файлу (скопированное на шаге 6 значение параметра **Component name**) в параметре **MobileApkPath**;
- Установите значение параметра **Landing.MobileClient=true**.

В результате:

- На стартовой странице системы будет отображена кнопка **Мобильное приложение**, при нажатии на которую на локальное устройство будет скачан загруженный установочный файл мобильного приложения (рис. 24);
- Скачиваемый при нажатии на кнопку **Мобильное приложение** файл будет иметь имя **MobileClient.apk**.

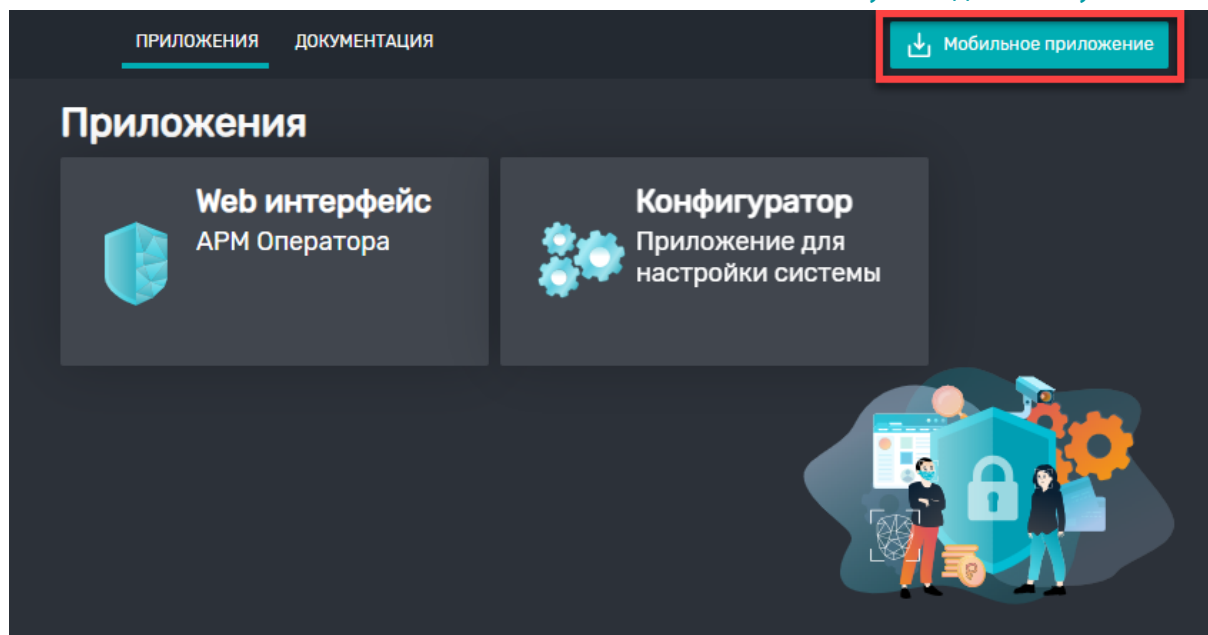


Рисунок 24 – Стартовая web-страница > кнопка **Мобильное приложение**

5.5 Добавление нового узла в кластер



Обратите внимание!

Все действия производятся на хосте с распакованным дистрибутивом системы (хост, с которого производилась установка системы).

Для добавления нового узла в кластер **Nomad**:

1. В файле **hosts** (расположен в каталоге **/ansible/inventory**) добавьте в группу параметров **[nomad_clients]** / **[nomad_servers]** имя или IP-адрес хоста, который необходимо добавить в кластер. **Не удаляйте уже имеющиеся в файле данные.**
2. Запустите **ansible playbook**. Для этого выполните следующие команды:

```
cd ansible/  
ansible-playbook -i inventory/hosts run.yml --tags install_nomad,install_consul -K
```
3. Дождитесь завершения процесса установки.
4. При возникновении ошибок **No cluster leader** восстановите кластер **Nomad / Consul** согласно разделу [Восстановление кластера](#). После восстановления кластера повторно запустите **ansible playbook** (см. шаг 2).

5.6 Перенос сервисов

Nomad позволяет освободить узел кластера от размещенных на узле сервисов системы. Для этого:

1. В web-интерфейсе **Nomad** перейдите на вкладку **Clients**.
2. Выберите узел кластера, сервисы с которого необходимо перенести на другой узел.
3. В открывшемся окне:



- Отключите опцию **Eligible** (рис. 25, **1**);
- В выпадающем меню **Drain** нажмите кнопку **Drain** (рис. 25, **2**).

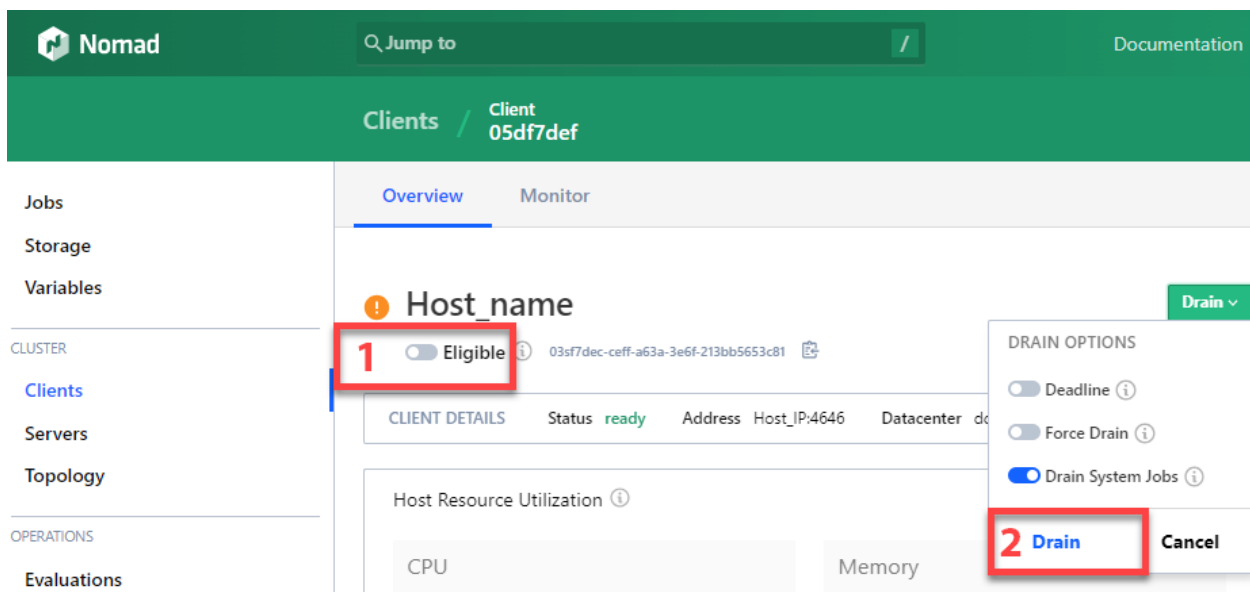


Рисунок 25 – Параметры узла кластера в интерфейсе Nomad

4. Перейдите на вкладку **Topology**.

5. На вкладке **Topology** соответствующий узел кластера будет иметь метку  и статус **Empty Client** (см. **Host_2** на рис. 26).

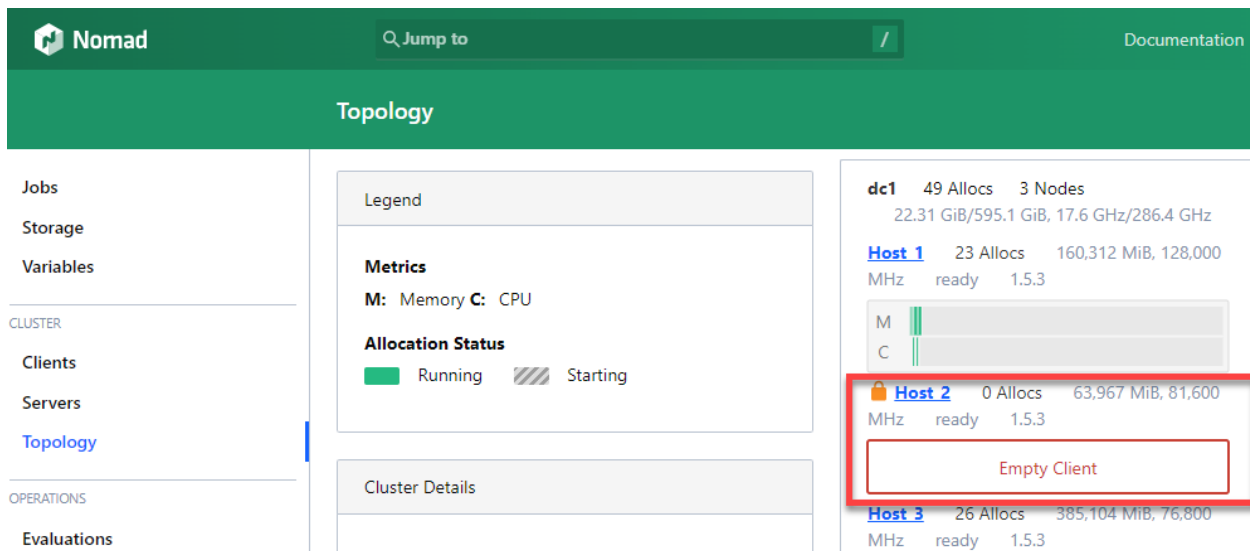


Рисунок 26 – Вкладка Topology

5.7 Выделение хоста под определенные сервисы



Описанный в данном разделе способ позволяет привязать сервисы к определенному хосту кластера. При этом запуск других сервисов на этом же хосте **будет запрещен**.

Для привязки сервиса к определенному узлу кластера:



1. Откройте файл **nomad_client.hcl**, расположенный в каталоге **/etc/nomad.d/** на хосте, **к которому необходимо привязать сервисы.**
2. В файле **nomad_client.hcl** добавьте в блок параметров **client** следующий блок кода:

```
node_pool = "pool_name"
```

где **pool_name** – произвольное название добавляемого пула.

3. Перезапустите **Nomad**:

```
systemctl restart nomad
```

4. Освободите узел кластера от размещенных на узле сервисов (см. [Перенос сервисов](#)).
5. В web-интерфейсе **Nomad** перейдите на вкладку **Jobs**.
6. Выберите нужное задание из списка (имя задания соответствует имени сервиса).
7. В открывшемся окне перейдите на вкладку **Definition** и нажмите кнопку **Edit**.
8. В параметре **NodePool** укажите название пула, добавленного на шаге 2.

```
Job Definition

1 {
2   "Stop": false,
3   "Region": "global",
4   "Namespace": "default",
5   "ID": "tasks-service",
6   "ParentID": "",
7   "Name": "tasks-service",
8   "Type": "service",
9   "Priority": 60,
10  "AllAtOnce": false,
11  "Datacenters": [
12    "dc1"
13  ],
14  "NodePool": "pool name",
```

Рисунок 27 – Параметр **NodePool**

где **pool_name** – имя пула, добавленного на шаге 2.

9. Повторите шаги 5-8 для всех нужных сервисов.

После приведенных выше действий на хосте будут запущены только указанные сервисы, а также **Nginx** и **Harproxy**.

5.8 Удаление системы



Обратите внимание!

При удалении будут потеряны **все** данные системы, включая данные в хранилище **Minio** и БД сервисов.



Для удаления системы выполните следующие действия на хосте с распакованным дистрибутивом системы (хост, с которого производилась **установка** системы):

1. Убедитесь, что в файле **hosts** (расположен в распакованном дистрибутиве системы в каталоге **./ansible/inventory**) указаны корректные имена или IP-адреса хостов кластера системы. **Все данные и сервисы системы будут удаляться с хостов, указанных в файле hosts.**
2. Перейдите в каталог **ansible** и запустите **ansible playbook**:

```
cd ansible/
sudo ansible-playbook -i inventory/hosts uninstall.yml -K
```

3. Дождитесь завершения процесса удаления системы. После завершения удаления будет отображен информационный блок **Play recap** (рис. 28).

```
PLAY RECAP *****
localhost : ok=15   changed=6    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
Host_1 : ok=106  changed=69   unreachable=0    failed=0    skipped=37   rescued=0    ignored=2
Host_2 : ok=60   changed=44   unreachable=0    failed=0    skipped=11   rescued=0    ignored=2
Host_3 : ok=57   changed=41   unreachable=0    failed=0    skipped=14   rescued=0    ignored=2
```

Рисунок 28 – Информационный блок **Play recap**



После удаления, перед повторной установкой системы **обязательно вручную удалите** файл **ansible/inventory/group_vars/acl.yml** из дистрибутива системы (или убедитесь, что файл отсутствует).

5.9 Восстановление кластера

При возникновении сбоев в работе кластера, связь с хостами, входящими в кластер системы, может быть потеряна. В этом случае лог-файлы сервисов **Nomad / Consul** будут содержать строки с ошибкой **No cluster leader**.

Для восстановления работоспособности кластера системы выполните следующие действия на хосте с распакованным дистрибутивом системы (хост, с которого производилась **установка** системы):

1. Убедитесь, что в файле **hosts** (расположен в распакованном дистрибутиве системы в каталоге **./ansible/inventory**) указаны корректные имена или IP-адреса хостов кластера системы;
2. Перейдите в каталог **ansible** и запустите **ansible playbook** с тегом **nomad** и/или **consul** в зависимости от того, в каких лог-файлах (сервиса **Nomad** или **Consul**) была ошибка **No cluster leader**. Возможно одновременное использование тегов **nomad** и **consul**:

– Для восстановления кластера **Nomad**:

```
sudo ansible-playbook -i inventory/hosts cluster-recovery.yml -K --tags nomad
```

– Для восстановления кластера **Consul**:

```
sudo ansible-playbook -i inventory/hosts cluster-recovery.yml -K --tags consul
```

– Для восстановления кластеров **Nomad** и **Consul**:

```
sudo ansible-playbook -i inventory/hosts cluster-recovery.yml -K --tags nomad, consul
```



6 УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ

6.1 Заккрытие незавершенных сессий GCC

При перезагрузке сервиса **Guardant Control Center (GCC)** могут оставаться незавершенные сессии.

В этом случае необходимо вручную завершить сессии. Для этого:

1. После перезагрузки сервиса **GCC** выполните следующую команду для проверки наличия незавершенных сессий:

```
netstat -tulnap | grep LISTEN | grep 3189
```

2. Если вывод команды **содержит** данные – система работает корректно, и закрытие сессий **не требуется**. Пример вывода команды:

```
tcp 0 0 0.0.0.0:3189 0.0.0.0:* LISTEN 1201514/grdcontrold
```

3. Если вывод команды **не содержит** данных – требуется ручное закрытие незавершенных сессий **GCC**. Для этого:

– Выполните следующую команду:

```
netstat -tulnap | grep 3189
```

Вывод команды будет содержать незакрытые сессии **GCC**. Пример вывода команды:

```
tcp 0 0 10.10.10.10:43414 10.10.10.11:3189 ESTABLISHED 4108186/java
```

где **4108186** – уникальный ID сессии **GCC**.

– Для каждой сессии **GCC** выполните команду **kill -9**, например:

```
kill -9 4108186
```



7 ОБНОВЛЕНИЕ СИСТЕМЫ

7.1 Обновление с версии 4.8



Для обновления системы необходимы:

1. Архив **pgupdate.tar**, содержащий скрипты обновления **Postgres**.
2. Дистрибутив **Визирь.СКУД**.

Алгоритм обновления системы:

Перед обновлением

Перед обновлением системы **обязательно**:

1. Проверьте количество записей в таблице **events-service.Tracks**, и при необходимости, произведите дополнительные настройки (см. [Настройка сервиса Events Service](#));
2. Выполните следующий SQL-запрос для БД сервиса **events-service**:

```
DELETE FROM "___EFMigrationsHistory" WHERE "___EFMigrationsHistory"."MigrationId" =  
'20230608103903_AddExternalId';  
ALTER TABLE "CombinedMatchLists" DROP COLUMN "ExternalId";
```

Обновление

1. Остановите сервисы **rts**, **platform**, **logcleaner**:

```
systemctl stop rts platform logcleaner
```

2. Удалите службу **systemd service platform**:

```
rm -rf /etc/systemd/system/platform.service  
systemctl daemon-reload
```

3. Распакуйте архив **pgupdate.tar** в любой каталог.

4. Перейдите в распакованный каталог:

```
cd pgupdate/
```

5. Откройте файл **env.sh**, расположенный в распакованном на шаге 4 каталоге, с помощью текстового редактора.

6. В файле **env.sh** при необходимости измените следующие параметры:

- **OLD** – текущая мажорная версия **Postgres** (**12** или **15**);
- **NEW** – версия **Postgres**, на которую производится обновление (оставьте значение **16**);
- **PGDATAOLD** – путь к текущей базе данных **Postgres**;
- **PGDATATMP** – путь к временной базе данных **Postgres**, создаваемой в процессе обновления версии **Postgres**. После обновления версии **Postgres** данная директория будет **автоматически удалена**;



- **PGDATABACKUP** – путь к резервной копии текущей базы данных **Postgres**. Резервная копия текущей базы данных автоматически будет создана в процессе выполнения скрипта **pgupdate.sh**. После обновления версии **Postgres** данная директория **не будет** удалена;

7. Запустите скрипт **pgupdate.sh**:

```
bash pgupdate.sh
```

После обновления **Postgres** будет отображена фраза **Upgrade Complete**.

8. Установите новую версию **Визирь.СКУД** (см. [УСТАНОВКА ПРОГРАММЫ](#)). При установке **обязательно** распределите хосты по блокам **[nomad_clients]** / **[nomad_servers]**, а также замените группу **[nomad]** в файле **hosts** на следующую:

```
[nomad:children]
nomad_servers
nomad_clients
```

При возникновении ошибок **No cluster leader** восстановите кластер **Nomad / Consul** согласно разделу [Восстановление кластера](#). После восстановления кластера продолжите установку системы.



Обратите внимание! На этапе **распаковки** дистрибутива с новой версией системы (см. [УСТАНОВКА ПРОГРАММЫ](#)) **обязательно**:

- Убедитесь, что в разархивированном каталоге с дистрибутивом **новой** версии системы в **./ansible/inventory/group_vars** присутствует файл **acl.yml**;
- При **отсутствии** файла – скопируйте его из каталога **./ansible/inventory/group_vars**, расположенного в распакованном дистрибутиве **старой (уже установленной)** версии системы в аналогичный каталог с распакованным дистрибутивом **новой** версии системы;
- Убедитесь, что в файле **acl.yml** присутствует параметр **DNS_token_AccessorID** (например, **DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'**). При отсутствии параметра:

- Выполните следующие команды:

```
export CONSUL_HTTP_TOKEN=<consul_token>
```

```
consul acl token list
```

где **consul_token** – токен для доступа к **Consul**. Токен расположен в файле **acl.yml** (параметр **consul_token**).

- В выводе команды **consul acl token list** из группы параметров с **Description: DNS-requests** скопируйте значение параметра **AccessorID**;
- Добавьте в файл **acl.yml** параметр **DNS_token_AccessorID** со значением параметра **AccessorID** в формате **DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'**.

9. При необходимости обновите лицензии (см. [АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ](#)).



10. Запустите скрипт **refreshCollation.sh**. При запуске скрипта потребуется ввести текущий пароль доступа к **Postgres**.

```
bash refreshCollation.sh
```

11. В web-интерфейсе **Nomad** перезапустите задание **postgresql**.

12. Обновите БД сервиса **events-service** (см. [Обновление БД сервиса events-service](#)).

13. Если после обновления лог-файл сервиса **Person Card Service** содержит записи об ошибках, приведенные на рисунке ниже, выполните следующие действия:

```
PersonCardsService
[06:44:41 INF] Starting host for person-cards-service 4.12.0.1736
[06:45:13 ERR] Failed executing DbCommand (0ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
UPDATE "PersonCardModels" SET "Timestamp" = "PersonCardImageMetadatas"."Timestamp" FROM "PersonCardImageMetadatas" WHERE "PersonCardModels"."PersonCardImageId" = "PersonCardImageMetadatas"."Id" AND "PersonCardModels"."Timestamp" is null
[06:45:13 ERR] Failed to execute command MigrateDatabaseCommand<PersonCardsContext>:38f17188-55ca-452d-ba76-a2862831c802
```

Рисунок 29 – Ошибка в лог-файле сервиса **Person Card Service**

- Остановите задание (job) сервиса **Person Card Service** в web-интерфейсе **Nomad** (работа с **Nomad** подробно описана в документе **Руководство администратора**).
- Подключитесь к контейнеру **postgresql**. Для этого:
 - В web-интерфейсе **Nomad** перейдите на вкладку **Jobs**;
 - Выберите задание **postgresql** из списка;
 - В открывшемся окне нажмите кнопку **Exec** (рис. 30).

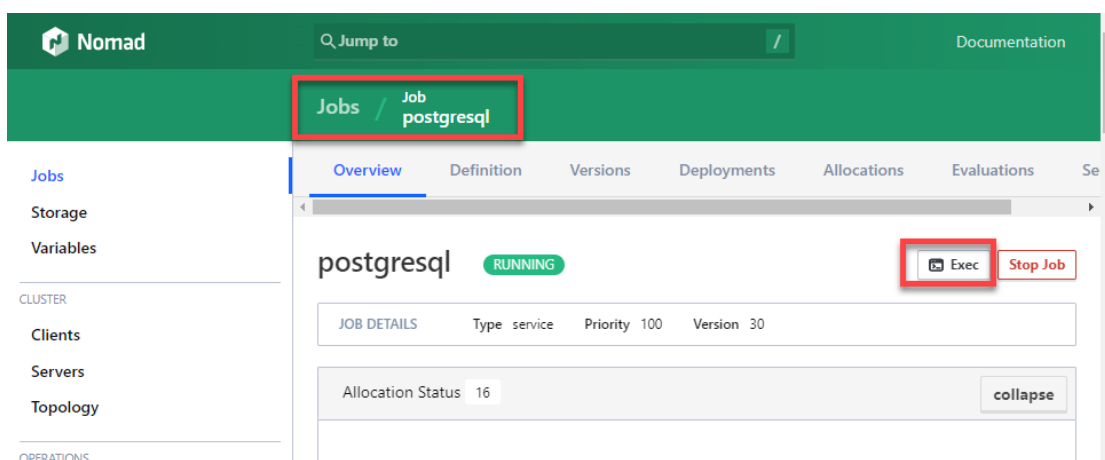


Рисунок 30 – Задание **postgresql** в web-интерфейсе **Nomad**

- В открывшемся окне (рис. 31):
 - Перейдите на вкладку **postgresql > postgresql** (рис. 31, **1**);
 - Последовательно выполните следующие команды (рис. 31, **2**):

```
psql -Upostgres
\c person-cards-service;
UPDATE "PersonCardModels" SET "Timestamp" = "PersonCardImageMetadatas"."Timestamp" FROM
"PersonCardImageMetadatas" WHERE "PersonCardModels"."PersonCardImageId" =
"PersonCardImageMetadatas"."Id" AND "PersonCardModels"."Timestamp" is null;
```



The screenshot shows the Nomad web interface. On the left, under 'TASKS', the 'postgresql' task is selected and highlighted with a red box and the number '1'. The main panel shows the task configuration and a terminal session. The terminal session is also highlighted with a red box and the number '2'. The terminal output shows the command to allocate and exec the postgresql task, followed by the postgresql command prompt and a SQL query.

```
root@00ebbb6fa55f8:/# psql -Upostgres
psql (15.2 (Debian 15.2-1.pgdg110+1))
Type "help" for help.

postgres=# \c person-cards-service;
You are now connected to database "person-cards-service" as user "postgres".
person-cards-service=# UPDATE "PersonCardModels" SET "Timestamp" = "PersonCardImageMetad
atas"."Timestamp" FROM "PersonCardImageMetadatas" WHERE "PersonCardModels"."PersonCardIm
ageId" = "PersonCardImageMetadatas"."Id" AND "PersonCardModels"."Timestamp" is null;
```

Рисунок 31 – Подключение к контейнеру postgresql

После обновления

После обновления системы **обязательно**:

- Выполните перестроение индексов галерей, используемых для поиска по архиву (см. Перестроение индексов);
- Войдите в **Конфигуратор** системы под учетной записью пользователя с ролью **sysadmins**. Это необходимо для автоматического создания системных данных;
- Удалите БД сервиса **video-analytics-tasks-processor-service** и перезапустите задание **video-analytics-tasks-processor-service** в интерфейсе **Nomad**;
- При работе новой версии системы через сервис **SRTS**:
 - В интерфейсе **Конфигуратора** удалите все пресеты IP-камер и обновите страницу браузера. В открывшемся после обновления страницы окне **Настройка БД** установите опцию **Создать базовые пресеты для камер** и нажмите кнопку **Запустить**;
 - В конфигурационном файле **settings.Presets.json** (расположен в каталоге **/configuration-service/config/** web-интерфейса **Consul**) в блоке **Presets > IP** установите значение параметра **frameBufferSize=10**;
- Для всех БД сервисов системы выполните команду **vacuum analyze**.

7.2 Обновление с версии 4.9 - 4.12



Для обновления системы необходимы:

1. Архив **pgupdate.tar**, содержащий скрипты обновления **Postgres**.
2. Дистрибутив **Визирь.СКУД**.

Алгоритм обновления системы:



Перед обновлением

Перед обновлением системы **обязательно** проверьте количество записей в таблице **events-service.Tracks**, и при необходимости, произведите дополнительные настройки (см. [Настройка сервиса Events Service](#)).

Обновление

1. Остановите сервисы **rts, platform, logcleaner**:

```
systemctl stop rts platform logcleaner
```

2. При обновлении с версии системы **4.11** и ранее удалите службу **systemd service platform**:

```
rm -rf /etc/systemd/system/platform.service  
systemctl daemon-reload
```

3. Распакуйте архив **pgupdate.tar** в любой каталог.

4. Перейдите в распакованный каталог:

```
cd pgupdate/
```

5. Откройте файл **env.sh**, расположенный в распакованном на шаге 4 каталоге, с помощью текстового редактора.

6. В файле **env.sh** при необходимости измените следующие параметры:

- **OLD** – текущая мажорная версия **Postgres** (**12** или **15**);
- **NEW** – версия **Postgres**, на которую производится обновление (оставьте значение **16**);
- **PGDATAOLD** – путь к текущей базе данных **Postgres**;
- **PGDATATMP** – путь к временной базе данных **Postgres**, создаваемой в процессе обновления версии **Postgres**. После обновления версии **Postgres** данная директория будет **автоматически удалена**;
- **PGDATABACKUP** – путь к резервной копии текущей базы данных **Postgres**. Резервная копия текущей базы данных автоматически будет создана в процессе выполнения скрипта **pgupdate.sh**. После обновления версии **Postgres** данная директория **не будет удалена**;

7. Запустите скрипт **pgupdate.sh**:

```
bash pgupdate.sh
```

После обновления **Postgres** будет отображена фраза **Upgrade Complete**.

8. Установите новую версию **Визирь.СКУД** (см. [УСТАНОВКА ПРОГРАММЫ](#)). При установке **обязательно** распределите хосты по блокам **[nomad_clients]** / **[nomad_servers]**, а также замените группу **[nomad]** в файле **hosts** на следующую:

```
[nomad:children]  
nomad_servers  
nomad_clients
```

При возникновении ошибок **No cluster leader** восстановите кластер **Nomad / Consul** согласно разделу [Восстановление кластера](#). После восстановления кластера продолжите установку системы.



Обратите внимание! На этапе **распаковки** дистрибутива с новой версией системы (см. [УСТАНОВКА ПРОГРАММЫ](#)) **обязательно**:

- Убедитесь, что в разархивированном каталоге с дистрибутивом **новой** версии системы в **`./ansible/inventory/group_vars`** присутствует файл **`acl.yml`**;
- При **отсутствии** файла – скопируйте его из каталога **`./ansible/inventory/group_vars`**, расположенного в распакованном дистрибутиве **старой (уже установленной)** версии системы в аналогичный каталог с распакованным дистрибутивом **новой** версии системы;
- Убедитесь, что в файле **`acl.yml`** присутствует параметр **`DNS_token_AccessorID`** (например, **`DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'`**). При отсутствии параметра:

- Выполните следующие команды:

```
export CONSUL_HTTP_TOKEN=<consul_token>
```

```
consul acl token list
```

где **`consul_token`** – токен для доступа к **Consul**. Токен расположен в файле **`acl.yml`** (параметр **`consul_token`**).

- В выводе команды **`consul acl token list`** из группы параметров с **Description: DNS-requests** скопируйте значение параметра **`AccessorID`**;
- Добавьте в файл **`acl.yml`** параметр **`DNS_token_AccessorID`** со значением параметра **`AccessorID`** в формате **`DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'`**.

9. При необходимости обновите лицензии (см. [АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ](#)).

10. Запустите скрипт **`refreshCollation.sh`**. При запуске скрипта потребуется ввести текущий пароль доступа к **Postgres**.

```
bash refreshCollation.sh
```

11. В web-интерфейсе **Nomad** перезапустите задание **postgresql**.

12. Обновите БД сервиса **events-service** (см. [Обновление БД сервиса events-service](#)).

13. Если после обновления лог-файл сервиса **Person Card Service** содержит записи об ошибках, приведенные на рисунке ниже, выполните следующие действия:

```
PersonCardsService
[06:44:41 INF] Starting host for person-cards-service 4.12.0.1736
[06:45:13 ERR] Failed executing DbCommand (0ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
UPDATE "PersonCardModels" SET "Timestamp" = "PersonCardImageMetadata"."Timestamp" FROM "PersonCardImageMetadata" WHERE "PersonCardModels"."PersonCardImageId" = "PersonCardImageMetadata"."Id" AND "PersonCardModels"."Timestamp" is null
[06:45:13 ERR] Failed to execute command MigrateDatabaseCommand(PersonCardsContext):38f17188-55ca-452d-ba76-a2862831c802
```

Рисунок 32 – Ошибка в лог-файле сервиса **Person Card Service**

- Остановите задание (job) сервиса **Person Card Service** в web-интерфейсе **Nomad** (работа с **Nomad** подробно описана в документе **Руководство администратора**).
- Подключитесь к контейнеру **postgresql**. Для этого:



- В web-интерфейсе **Nomad** перейдите на вкладку **Jobs**;
- Выберите задание **postgresql** из списка;
- В открывшемся окне нажмите кнопку **Exec** (рис. 33).

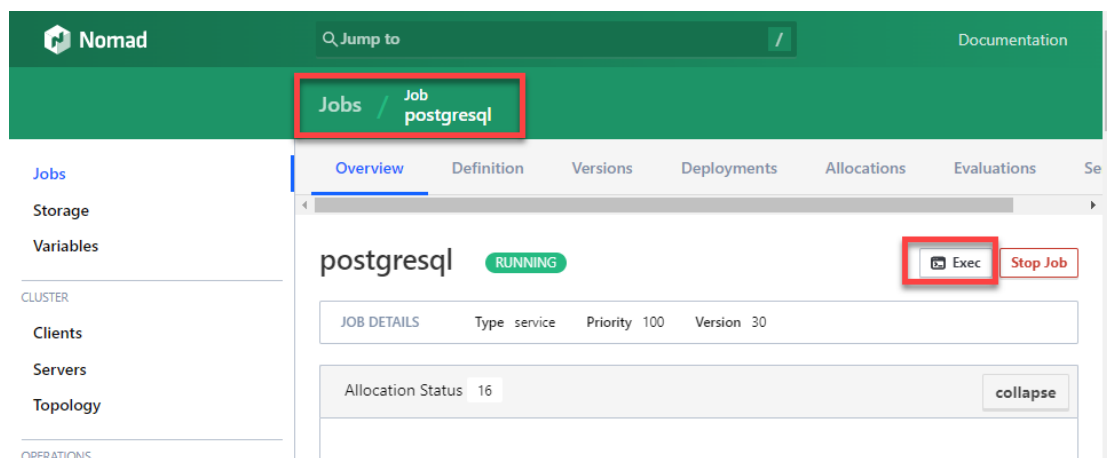


Рисунок 33 – Задание **postgresql** в web-интерфейсе **Nomad**

- В открывшемся окне (рис. 34):
- Перейдите на вкладку **postgresql > postgresql** (рис. 34, 1);
- Последовательно выполните следующие команды (рис. 34, 2):

```
psql -Upostgres
\c person-cards-service;
UPDATE "PersonCardModels" SET "Timestamp" = "PersonCardImageMetadatas"."Timestamp" FROM
"PersonCardImageMetadatas" WHERE "PersonCardModels"."PersonCardImageId" =
"PersonCardImageMetadatas"."Id" AND "PersonCardModels"."Timestamp" is null;
```

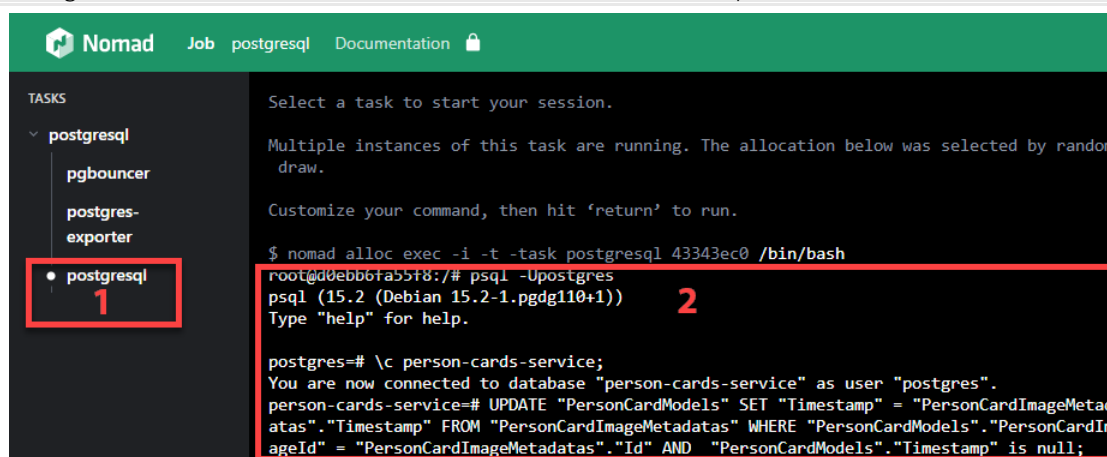


Рисунок 34 – Подключение к контейнеру **postgresql**

После обновления

После обновления системы **обязательно**:

- Выполните перестроение индексов галерей, используемых для поиска по архиву (см. Перестроение индексов);



- Войдите в **Конфигуратор** системы под учетной записью пользователя с ролью **sysadmins**. Это необходимо для автоматического создания системных данных;
- Удалите БД сервиса **video-analytics-tasks-processor-service** и перезапустите задание **video-analytics-tasks-processor-service** в интерфейсе **Nomad**;
- При работе новой версии системы через сервис **SRTS**:
 - В интерфейсе **Конфигуратора** удалите все пресеты IP-камер и обновите страницу браузера. В открывшемся после обновления страницы окне **Настройка БД** установите опцию **Создать базовые пресеты для камер** и нажмите кнопку **Запустить**;
 - В конфигурационном файле **settings.Presets.json** (расположен в каталоге **/configuration-service/config/** web-интерфейса **Consul**) в блоке **Presets > IP** установите значение параметра **frameBufferSize=10**;
- Для всех БД сервисов системы выполните команду **vacuum analyze**.

7.3 Обновление с версии 4.13 / 4.14



Для обновления системы необходимы:

1. Архив **pgupdate.tar**, содержащий скрипты обновления **Postgres**.
2. Дистрибутив **Визирь.СКУД**.

Алгоритм обновления системы:

Перед обновлением

Перед обновлением системы **обязательно** проверьте количество записей в таблице **events-service.Tracks**, и при необходимости, произведите дополнительные настройки (см. [Настройка сервиса Events Service](#)).

Обновление

1. Остановите сервисы **rts**, **platform**, **logcleaner**:

```
systemctl stop rts platform logcleaner
```

2. Распакуйте архив **pgupdate.tar** в любой каталог.

3. Перейдите в распакованный каталог:

```
cd pgupdate/
```

4. Откройте файл **env.sh**, расположенный в распакованном на шаге 3 каталоге, с помощью текстового редактора.

5. В файле **env.sh** при необходимости измените следующие параметры:

- **OLD** – текущая мажорная версия **Postgres** (**12** или **15**);
- **NEW** – версия **Postgres**, на которую производится обновление (оставьте значение **16**);
- **PGDATAOLD** – путь к текущей базе данных **Postgres**;



- **PGDATATMP** – путь к временной базе данных **Postgres**, создаваемой в процессе обновления версии **Postgres**. После обновления версии **Postgres** данная директория будет **автоматически удалена**;
- **PGDATABACKUP** – путь к резервной копии текущей базы данных **Postgres**. Резервная копия текущей базы данных автоматически будет создана в процессе выполнения скрипта **pgupdate.sh**. После обновления версии **Postgres** данная директория **не будет** удалена;

6. Запустите скрипт **pgupdate.sh**:

```
bash pgupdate.sh
```

После обновления **Postgres** будет отображена фраза **Upgrade Complete**.

7. Установите новую версию **Визирь.СКУД** (см. [УСТАНОВКА ПРОГРАММЫ](#)). При установке **обязательно** распределите хосты по блокам **[nomad_clients]** / **[nomad_servers]**, а также замените группу **[nomad]** в файле **hosts** на следующую:

```
[nomad:children]
nomad_servers
nomad_clients
```

При возникновении ошибок **No cluster leader** восстановите кластер **Nomad / Consul** согласно разделу [Восстановление кластера](#). После восстановления кластера продолжите установку системы.



Обратите внимание! На этапе **распаковки** дистрибутива с новой версией системы (см. [УСТАНОВКА ПРОГРАММЫ](#)) **обязательно**:

- Убедитесь, что в разархивированном каталоге с дистрибутивом **новой** версии системы в **`./ansible/inventory/group_vars`** присутствует файл **`acl.yml`**;
- При **отсутствии** файла – скопируйте его из каталога **`./ansible/inventory/group_vars`**, расположенного в распакованном дистрибутиве **старой (уже установленной)** версии системы в аналогичный каталог с распакованным дистрибутивом **новой** версии системы;
- Убедитесь, что в файле **`acl.yml`** присутствует параметр **`DNS_token_AccessorID`** (например, **`DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'`**). При отсутствии параметра:

- Выполните следующие команды:

```
export CONSUL_HTTP_TOKEN=<consul_token>
```

```
consul acl token list
```

где **`consul_token`** – токен для доступа к **Consul**. Токен расположен в файле **`acl.yml`** (параметр **`consul_token`**).

- В выводе команды **`consul acl token list`** из группы параметров с **Description: DNS-requests** скопируйте значение параметра **`AccessorID`**;
- Добавьте в файл **`acl.yml`** параметр **`DNS_token_AccessorID`** со значением параметра **`AccessorID`** в формате **`DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'`**.

8. При необходимости обновите лицензии (см. [АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ](#)).

9. Запустите скрипт **`refreshCollation.sh`**. При запуске скрипта потребуется ввести текущий пароль доступа к **Postgres**.

```
bash refreshCollation.sh
```

10. В web-интерфейсе **Nomad** перезапустите задание **postgresql**.

11. Если после обновления лог-файл сервиса **Person Card Service** содержит записи об ошибках, приведенные на рисунке ниже, выполните следующие действия:

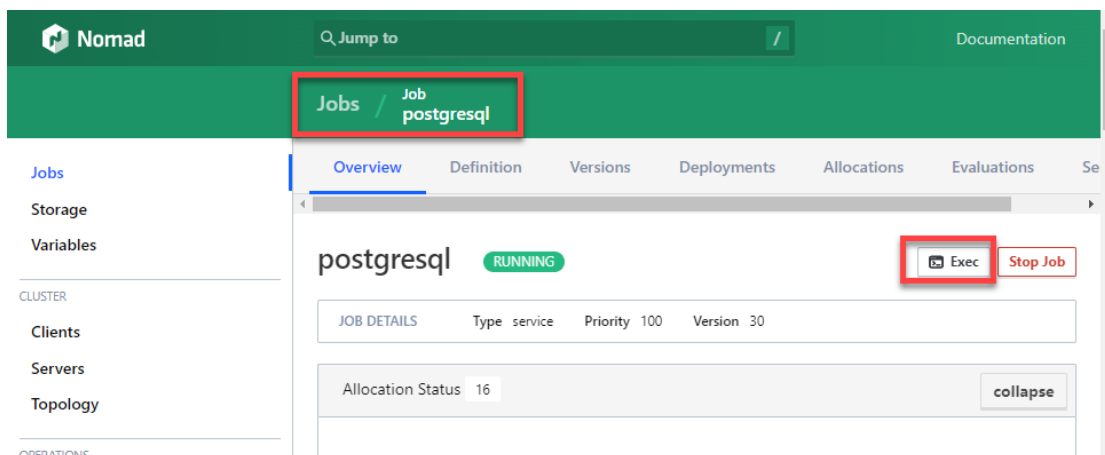
```
PersonCardService
[06:44:41 INF] Starting host for person-cards-service 4.12.0.1736
[06:45:13 ERR] Failed executing DbCommand (0ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
UPDATE "PersonCardModels" SET "Timestamp" = "PersonCardImageMetadata"."Timestamp" FROM "PersonCardImageMetadata" WHERE "PersonCardModels"."PersonCardImageId" = "PersonCardImageMetadata"."Id" AND "PersonCardModels"."Timestamp" is null
[06:45:13 ERR] Failed to execute command MigrateDatabaseCommand(PersonCardsContext):38f17188-55ca-452d-ba76-a2862831c802
```

Рисунок 35 – Ошибка в лог-файле сервиса **Person Card Service**

- Остановите задание (job) сервиса **Person Card Service** в web-интерфейсе **Nomad** (работа с **Nomad** подробно описана в документе **Руководство администратора**).
- Подключитесь к контейнеру **postgresql**. Для этого:
 - В web-интерфейсе **Nomad** перейдите на вкладку **Jobs**;

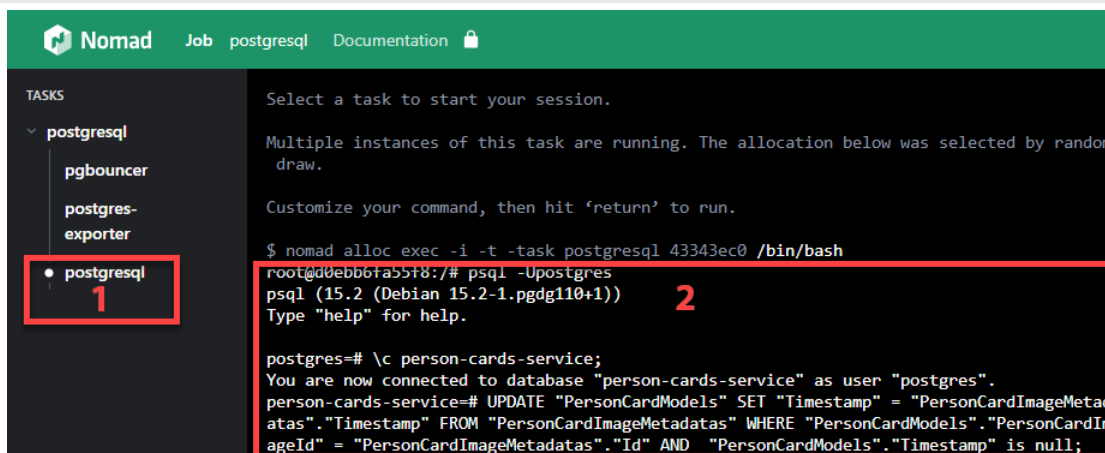


- Выберите задание **postgresql** из списка;
- В открывшемся окне нажмите кнопку **Exec** (рис. 36).

Рисунок 36 – Задание **postgresql** в web-интерфейсе **Nomad**

- В открывшемся окне (рис. 37):
 - Перейдите на вкладку **postgresql > postgresql** (рис. 37, 1);
 - Последовательно выполните следующие команды (рис. 37, 2):

```
psql -Upostgres
\c person-cards-service;
UPDATE "PersonCardModels" SET "Timestamp" = "PersonCardImageMetadatas"."Timestamp" FROM
"PersonCardImageMetadatas" WHERE "PersonCardModels"."PersonCardImageId" =
"PersonCardImageMetadatas"."Id" AND "PersonCardModels"."Timestamp" is null;
```

Рисунок 37 – Подключение к контейнеру **postgresql**

После обновления

После обновления системы **обязательно**:

- Выполните перестроение индексов галерей, используемых для поиска по архиву (см. Перестроение индексов);
- Войдите в **Конфигуратор** системы под учетной записью пользователя с ролью **sysadmins**. Это необходимо для автоматического создания системных данных;



- Удалите БД сервиса **video-analytics-tasks-processor-service** и перезапустите задание **video-analytics-tasks-processor-service** в интерфейсе **Nomad**;
- При работе новой версии системы через сервис **SRTS**:
 - В интерфейсе **Конфигуратора** удалите все пресеты IP-камер и обновите страницу браузера. В открывшемся после обновления страницы окне **Настройка БД** установите опцию **Создать базовые пресеты для камер** и нажмите кнопку **Запустить**;
 - В конфигурационном файле **settings.Presets.json** (расположен в каталоге **/configuration-service/config/** web-интерфейса **Consul**) в блоке **Presets > IP** установите значение параметра **frameBufferSize=10**;
- Для всех БД сервисов системы выполните команду **vacuum analyze**.

7.4 Обновление с версии 4.15 и выше

Алгоритм обновления системы:

1. **Перед** обновлением системы **обязательно** проверьте количество записей в таблице **events-service.Tracks**, и при необходимости, произведите дополнительные настройки (см. [Настройка сервиса Events Service](#)).
2. Установите новую версию **Визирь.СКУД** (см. [УСТАНОВКА ПРОГРАММЫ](#)). При установке **обязательно** распределите хосты по блокам **[nomad_clients]** / **[nomad_servers]**, а также замените группу **[nomad]** в файле **hosts** на следующую:

```
[nomad:children]
nomad_servers
nomad_clients
```

При возникновении ошибок **No cluster leader** восстановите кластер **Nomad / Consul** согласно разделу [Восстановление кластера](#). После восстановления кластера продолжите установку системы.



Обратите внимание! На этапе **распаковки** дистрибутива с новой версией системы (см. [УСТАНОВКА ПРОГРАММЫ](#)) **обязательно**:

- Убедитесь, что в разархивированном каталоге с дистрибутивом **новой** версии системы в **`./ansible/inventory/group_vars`** присутствует файл **`acl.yml`**;
- При **отсутствии** файла – скопируйте его из каталога **`./ansible/inventory/group_vars`**, расположенного в распакованном дистрибутиве **старой (уже установленной)** версии системы в аналогичный каталог с распакованным дистрибутивом **новой** версии системы;
- Убедитесь, что в файле **`acl.yml`** присутствует параметр **`DNS_token_AccessorID`** (например, **`DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'`**). При отсутствии параметра:

- Выполните следующие команды:

```
export CONSUL_HTTP_TOKEN=<consul_token>
```

```
consul acl token list
```

где **`consul_token`** – токен для доступа к **Consul**. Токен расположен в файле **`acl.yml`** (параметр **`consul_token`**).

- В выводе команды **`consul acl token list`** из группы параметров с **Description: DNS-requests** скопируйте значение параметра **`AccessorID`**;
- Добавьте в файл **`acl.yml`** параметр **`DNS_token_AccessorID`** со значением параметра **`AccessorID`** в формате **`DNS_token_AccessorID: 'e4e61076-b95b-302b-59f2-ce7cc0aea3f3'`**.

3. При необходимости обновите лицензии (см. [АКТИВАЦИЯ КЛЮЧА ЗАЩИТЫ](#)).

4. **После** обновления системы **обязательно**:

- Выполните перестроение индексов галерей, используемых для поиска по архиву (см. Перестроение индексов);
- Войдите в **Конфигуратор** системы под учетной записью пользователя с ролью **`sysadmins`**. Это необходимо для автоматического создания системных данных;
- Удалите БД сервиса **`video-analytics-tasks-processor-service`** и перезапустите задание **`video-analytics-tasks-processor-service`** в интерфейсе **Nomad**;
- При работе новой версии системы через сервис **SRTS**:
 - В интерфейсе **Конфигуратора** удалите все пресеты IP-камер и обновите страницу браузера. В открывшемся после обновления страницы окне **Настройка БД** установите опцию **Создать базовые пресеты для камер** и нажмите кнопку **Запустить**;



- В конфигурационном файле **settings.Presets.json** (расположен в каталоге **/configuration-service/config/** web-интерфейса **Consul**) в блоке **Presets > IP** установите значение параметра **frameBufferSize=10**;
- Для всех БД сервисов системы выполните команду **vacuum analyze**.

7.5 Дополнительная информация

7.5.1 Обновление БД сервиса events-service

При обновлении системы с ранних версий на более поздние (**4.13** и позднее) необходимо **обязательно** выполнить скрипты обновления БД. В противном случае все уже имеющиеся события идентификации не будут корректно отображаться в **Журнале событий**.

Алгоритм выполнения скрипта обновления БД (**после обновления** всех сервисов системы):

- Перезапустите задание (Job) **events-service** в web-интерфейсе **Nomad** (работа с **Nomad** подробно описана в документе **Руководство администратора**).
- Выполните следующий SQL-запрос для БД сервиса **events-service**:

```
UPDATE "Tracks" SET "EndTimestamp" =
(SELECT Max("DetectedFaces"."Timestamp")
 FROM "DetectedFaces" WHERE "Tracks"."Id" = "DetectedFaces"."TrackId"
 GROUP BY "TrackId")
WHERE "Tracks"."EndTimestamp" is null
```

- Выполните следующий SQL-запрос для БД сервиса **events-service**:

```
INSERT INTO "IdentCandidates" SELECT
c."Id",
c."Id",
c."CategoryId",
c."PersonCardId",
c."PersonCardImageId",
c."Score",
d."Id",
d."CameraId",
d."ModelConfidence",
d."Timestamp",
d."CropImageUrl",
null,
null,
l."Id",
l."ArealId",
t."Id",
t."StartTimestamp",
t."EndTimestamp",
p."Id",
p."Temperature",
p."Liveness",
p."MedicalMask",
p."Gender",
p."Age",
p."Beard",
```



```
p."Ethnicity",
l."ExternalId"
FROM "CombinedMatchCandidates" AS c
LEFT JOIN "CombinedMatchLists" AS l ON c."MatchListId" = l."Id"
LEFT JOIN "DetectedFaces" AS d ON l."DetectedFaceId" = d."Id"
LEFT JOIN "PortraitCharacteristics" AS p ON p."Id" = d."Id"
LEFT JOIN "Tracks" AS t ON d."TrackId" = t."Id"
WHERE t."MainDetectionId" IS NOT NULL AND t."StartTimestamp" IS NOT NULL AND t."EndTimestamp" IS
NOT NULL;
vacuum analyze "IdentCandidates";
```



Обратите внимание:

- При возникновении ошибок в процессе выполнения скрипта, выполните команды `insert` и `vacuum` отдельно (последовательно);
- Для расчета портретных характеристик используемый компонент SDK должен поддерживать расчет данных характеристик.

7.5.2 Настройка сервиса Events Service

Скорость обновления БД зависит от количества записей в таблицах БД. В частности, обновление таблицы **events-service.Tracks** занимает время, исходя из расчета 1 миллион треков за 72 секунды.

Таким образом, при наличии в таблице **events-service.Tracks** большого количества записей, **перед обновлением всех сервисов системы**, необходимо **обязательно** изменить значение параметров **CommandTimeout** в конфигурационном файле **appsettings.json** сервиса **events-service** исходя из следующей формулы: **CommandTimeout = N*72**, где **N** – количество миллионов треков в таблице **events-service.Tracks**.

После обновления системы необходимо обязательно вернуть исходные значения параметров CommandTimeout.

Примеры расчетов:

1. При количестве треков = 40 млн, **CommandTimeout = 40*72 = 2880** – не превышает установленное по умолчанию значение **3000**, таким образом, установленное по умолчанию значение **3000** менять **не нужно**.
2. При количестве треков = 60 млн, **CommandTimeout = 60*72 = 4320** – превышает установленное по умолчанию значение **3000**, таким образом, необходимо установить значение параметра **CommandTimeout** не менее, чем **4320**.

```
{,
  "AllowedHosts": "*",
  "events-service": {
    "MetricsPrefix": null,
    "DbSettings": {
      "Provider": "PostgreSQL",
      "ConnectionString": "User Id=postgres;Password=123Rty;Host=haproxy.service.consul;Port=5432;Database=events-service;CommandTimeout=3000;",
      "CommandTimeout": 3000
    }
  },
}
```

Рисунок 38 – Параметры **CommandTimeout**

Товарные знаки, упомянутые в документе, являются собственностью их владельцев.

Иллюстративный материал, приведённый в документе, может содержать вымышленные имена, фамилии, номера телефонов, адреса и тому подобные сведения. Любое их совпадение с данными реальных личностей следует считать случайностью.

Визирь.СКУД. Руководство по установке.

Версия документа от 13.04.2026.