

Инструкция по установке

Содержание

1. Сервер контроля	3
1.1. Установка PostgreSQL 15 и Docker	3
1.1.1. Установка PostgreSQL 15	3
1.1.2. Установка Docker	4
1.2. Установка сервера контроля	5
1.3. Настройка сервера контроля	6
1.4. Настройка Сервера Контроля для работы с доменными источниками	9
1.5. Установка https сертификата	10
1.6. Запуск и выключение сервера контроля	10
1.7. Обновление сервера контроля	11
1.8. Удаление сервера контроля	11
1.9. Логирование сервера контроля	12
2. Сервер анализа	13
2.1. Установка Docker	13
2.2. Установка сервера анализа	13
2.3. Настройка	14
2.3.1. Настройка локализации	14
2.3.2. Настройка сервера анализа	16
2.4. Установка https сертификата	21
2.5. Запуск и выключение сервера анализа	21
2.6. Обновление сервера анализа	22
2.7. Удаление сервера анализа	23
2.8. Логирование сервера анализа	23
3. Режим резервирования сервера контроля	24
3.1. Общая информация	24
3.2. Переключение режимов резервирования Сервера Контроля (Основной/Резервный)	25
3.3. Настройка серверов контроля для работы в режиме резервирования	26
3.4. Настройка сервера анализа	27
3.5. Пример настройки системы ORBOX в режиме резервирования сервера контроля	27
3.5.1. Настройка Сервера БД	30
3.5.2. Настройка СК1	31
3.5.3. Настройка СА	35
3.5.4. Настройка СК2	37

3.5.5. Тестирование работы режима резервирования сервера контроля	41
4. Архив брака	44
4.1. Общая информация	44
4.2. Установка архива брака	44
4.3. Настройка	45
4.4. Установка https сертификата	46
4.5. Запуск и выключение архива брака	46
5. ORBOX плеер	48
5.1. Общие требования	48
5.2. Первичная установка	48
5.3. Повторная установка	51
5.3.1. Полная установка	51
5.3.2. Обновление программы	53
6. Adobe Premier Plugin	55
6.1. Установка и настройка ORBOX Connector	55
7. Высокодоступный кластер PostgreSQL	59
7.1. Общие данные	59
7.2. Составные части кластера	59
7.3. Настройка доменных имён	59
7.4. Установка и настройка Etcd	60
7.5. Установка и настройка PostgreSQL	61
7.6. Установка и настройка Patroni	62
7.7. Установка и настройка HAProxy	65
7.8. Настройка сервера контроля ORBOX	66

1. Сервер контроля

1.1. Установка PostgreSQL 15 и Docker

В разделе 1.1. «Установка PostgreSQL 15 и Docker» указаны команды для установки PostgreSQL 15 и Docker для Операционной Системы Debian 11 и могут отличаться на других ОС.

Рекомендуем проверить правильность команд при установке на ОС, отличной от Debian 11, на официальных сайтах PostgreSQL и Docker:

Официальный сайт PostgreSQL: www.postgresql.org

Официальный сайт Docker: www.docker.com

1.1.1. Установка PostgreSQL 15

Важно! В разделе 1.1.1. «Установка PostgreSQL 15» команды выполняются из-под пользователя **root**, если не указано другое.

1. Переключиться на пользователя **root**:

```
su -
```

2. Установить PostgreSQL 15:

```
cd /tmp
apt update -y
apt upgrade -y
apt install lsb-release gnupg2 wget curl -y
sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt \
$(lsb_release -cs)-pgdg main" > \
/etc/apt/sources.list.d/pgdg.list'
wget --quiet -O - \
https://www.postgresql.org/media/keys/ACCC4CF8.asc \
| apt-key add -
apt update -y
apt install postgresql-15 -y
```

3. Запустить консольный клиент **psql** (запускается только от пользователя postgres) :

```
su - postgres
psql
```

4. Создать параметры доступа для базы данных (они так же будут указаны в файле конфигурации сервера контроля) можно с помощью команды в **psql**:

```
ALTER USER user_name PASSWORD 'new_password'
```

Например, чтобы создать параметры, которые указаны в файле конфигурации сервера контроля по умолчанию, пользователь postgres пароль 'postgres':

```
ALTER USER postgres PASSWORD 'postgres';  
exit
```

1.1.2. Установка Docker

Важно! В разделе 1.1.2. «Установка Docker» команды выполняются из-под пользователя **root**, если не указано другое.

1. Установить Docker версии не ниже 20.10.24.

Информацию по Docker смотри на официальном сайте docs.docker.com.

Важно! Нужно установить Docker именно с официального сайта, потому что в репозитории вашего дистрибутива может быть старая версия.

Перед тем, как вы установите Docker Engine в первый раз, нужно настроить репозиторий для Docker **apt**:

```
# Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl  
sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/  
keyrings/docker.asc  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
# Add the repository to Apt sources:  
echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/  
docker.asc] https://download.docker.com/linux/debian \  
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

Установить пакеты Docker:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

После успешной установки необходимо запустить сервис Docker и добавить его в автозагрузку:

```
systemctl enable docker --now
```

Проверить, что сервис запущен:

```
systemctl status docker
```

В статусе должно быть отображено 'active (running)'.

Проверить информацию об установленной версии docker можно командой :

```
docker version
```

Проверить системную информацию docker можно командой :

```
docker info
```

2. После установки Docker нужно добавить своего пользователя, под которым будет запускаться ORBOX, в группу docker.

Важно! В данном руководстве предполагается, что этот пользователь имеет имя **user**. Установка ORBOX из-под пользователя **root** или с помощью команды **sudo** не поддерживается.

Добавить пользователя **user** в группу docker:

```
sudo usermod -aG docker user
```

, где **user** - имя вашего пользователя.

3. Для применения настроек и прав для пользователя требуется перезапустить пользовательскую сессию или перезагрузить компьютер.

1.2. Установка сервера контроля

Важно! В разделе 1.2. «Установка сервера контроля» все команды выполняются из-под пользователя **user**, если не указано другое. Установка из-под пользователя **root** или с помощью **sudo** не поддерживается.

Установка сервера контроля производится следующим образом:

1. Если на компьютере установлена предыдущая версия ORBOX (не docker), то её требуется удалить. Следуйте шагам по удалению сервера контроля из инструкции для установленной версии.

Важно! Если для настройки межсетевого экрана (firewall) использовался скрипт `setup-firewall.sh` из поставки сервера контроля, то нужно отменить внесённые изменения запустив скрипт и выбрав пункт 3) `Remove all rules`.

2. Скопировать на машину, где планируется установить сервер контроля, 2 файла:

- `setup-orbox-server-docker-x.x.x.x-lic.sh`
- `orbox-server-docker-x.x.x.x-lic.tar.gz`

3. Установить для скрипта `setup-orbox-server-docker-x.x.x.x-lic.sh` права на выполнение:

```
chmod +x ./setup-orbox-server-docker-x.x.x.x-lic.sh
```

Для установки сервера контроля из папки, где находятся файлы, выполнить команду:

```
./setup-orbox-server-docker-x.x.x.x-lic.sh --install
```

После успешной установки файл `setup-orbox-server-docker-x.x.x.x-lic.sh` будет скопирован в папку `/home/user/.config/orbox-server/`.

4. Поместить файл лицензии `license.dat` рядом со скриптом `setup-orbox-server-docker-x.x.x.x-lic.sh` и выполнить команду:

```
./setup-orbox-server-docker-x.x.x.x-lic.sh --install-license license.dat
```

Эта команда копирует файл лицензии в папку `/home/user/.config/orbox-server/volume/orbox-server`

5. Обновить конфигурационные файлы командой:

```
./setup-orbox-server-docker-x.x.x.x-lic.sh --update-config all
```

Установка завершена.

1.3. Настройка сервера контроля

Настройки сервера контроля находятся в файле `appsettings.json`, который находится в папке `/home/user/.config/orbox-server/volume/orbox-server/conf/`.

Для быстрого открытия этого файла на редактирование можно воспользоваться командой:

```
./setup-orbox-server-docker-x.x.x.x-lic.sh --edit-config server
```

Описание настроек сервера из файла `appsettings.json`:

- `HttpPort` – Порт веб-сервера при использовании HTTP;
- `HttpsPort` – Порт веб-сервера при использовании HTTPS;
- `UseSSL` – Использовать защищенное соединение (HTTPS);
- `ApiTokenExpiresDays` – Число дней, после которого API-токен считается невалидным и требуется повторная авторизация. Установите 0, если токен выдаётся бессрочно;
- `CertificatePath` – Путь к *.pfx сертификату, оставьте пустым или null, если будет использоваться HTTPS сертификат по умолчанию;
- `CertificatePassword` – Пароль к файлу сертификата;
- `DatabaseConnectionString` – Имя строки подключения к базе данных;
- `NormalizationOutputFolder` – Имя подпапки для нормализованных файлов;
- `Language` – Язык интерфейса;
- `DarkMode` – Тёмный режим;
- `DateFormat` – Отображаемый формат даты;
- `LDAPAuthEnabled` – Разрешить аутентификацию LDAP;
- `LDAPServer` – Сервер для подключения к Active Directory. Используется для валидации доменных параметров доступа и для поиска пользователей при включенной LDAP аутентификации.
- `LDAPDomain` – Домен для подключения к Active Directory. Используется для поиска пользователей при включенной LDAP аутентификации.
- `LDAPUserName` – Имя доменного пользователя для подключения к Active Directory. Используется для поиска пользователей при включенной LDAP аутентификации.
- `LDAPUserPassword` – Пароль доменного пользователя для подключения к Active Directory. Используется для поиска пользователей при включенной LDAP аутентификации;
- `FileWatcher` – Вотчер для файловой системы, который будет использован. Возможные значения:

System Системный .NET File Watcher

Tecom Собственный File Watcher (Tecom)

MyOddWeb Сторонний File Watcher (myoddweb.directorywatcher)

- `LastWriteTimeWaitSeconds` – Минимальное время, которое должно пройти после записи файла до момента, когда он станет доступным для анализа;
- `MaxAgeOfFilesWhenSourceRestart` – Файлы, старше чем указанное значение (в днях) не будут добавлены в очередь при перезагрузке источника или старте сервера;

- `NotProcessAlreadyProcessedFiles` - Не обрабатывать уже обработанные файлы повторно;
- `MxfAtomWaitingByTimer` - Выполнять сбор файлов MXF OP-Atom бандла по таймеру. Если опция активна, то составные элементы бандла MXF Atom будут собираться до истечения таймера указанного в настройках системы (по общему Material Umid, который должен быть вшит в каждый файл). Если опция не активна, то составные элементы бандла MXF Atom будут определяться исходя из структуры, зашитой в каждый из файлов;
- `MxfAtomProcessMissingMaterialStructure` - Продолжить анализ MXF OP-Atom файлов с отсутствующей структурой материала. Если опция активна, то при отсутствии структуры материала внутри файла задача всё равно будет проанализирована (как составная задача, состоящая из единственного файла). В результате анализа будет добавлено предупреждение о том, что не найдена структура материала. Если опция не активна, то при отсутствии структуры материала внутри файла задача будет завершена с ошибкой «Не найдена структура материала»;
- `TimeoutDeclinedTask` - Время (в миллисекундах), которое задача со статусом Declined будет висеть в списке необработанных файлов, до следующей отправке серверу анализа;
- `RetryErrorJobsErrorCodes` - Список кодов ошибок, при которых требуется отправка задачи на повторный анализ:

1	Файл не найден	<code>FILE_NOT_FOUND</code>
2	Доступ к файлу запрещён	<code>FILE_ACCESS_DENIED</code>
3	Ошибка открытия файла	<code>FILE_OPEN_ERROR</code>
4	Ошибка подключения	<code>CONNECTION_ERROR</code>
5	Неверный аргумент	<code>INVALID_ARGUMENT</code>
6	Ошибка ввода-вывода	<code>IO_ERROR</code>
- `MaxResultsInReport` - Максимальное число результатов в сводном отчёте;
- `VideoDefinitions` - Список предустановленных разрешений видео;
- `Containers` - Список поддерживаемых видеоконтейнеров;
- `VideoCodecs` - Список поддерживаемых семейств видеокодеков, используется для выбора шаблона анализа файлов;
- `VideoCodecsCommercialNames` - Список специфических версий видеокодеков, поддерживаемых для выбора шаблона анализа файлов;
- `AudioCodecs` - Список поддерживаемых аудиокодеков;
- `Tests` - Тесты по категориям;
- `UseSmpteDuration` - Использовать формат SMPTE Timecode при отображении длительности видео;
- `Replication` - Настройки резервирования;

- `ServerReplicationEnabled` – Резервирование сервера контроля включено (`true`) или отключено (`false`);
- `Port` – Порт для синхронизации резервирования (для текущего узла);
- `ServerNodes` – Узлы резервирования сервера контроля в формате АДРЕС:ПОРТ, включая текущий узел. Порядок указания узлов соответствует приоритету при выборе основного узла (первый имеет высший приоритет);
- `DatabaseReplicationEnabled` – Резервирование БД включено (`true`) или отключено (`false`);
- `DatabaseNodes` – Узлы резервирования БД CooroachDB в формате ХОСТ:ПОРТ;

1.4. Настройка Сервера Контроля для работы с доменными источниками

Для работы с источниками, доступных с использованием доменной учетной записи, в настройках Сервера Контроля необходимо заполнить следующие параметры: `LDAPServer` и `LDAPDomain`.

Для изменений параметров Сервера Контроля нужно остановить Сервер контроля и отредактировать настройки используя команды :

```
cd /home/user/.config/orbox-server/
./setup-orbox-server-docker-x.x.x.x-lic.sh --disable
./setup-orbox-server-docker-x.x.x.x-lic.sh --edit-config server
```

Например, если для доступа требуется пользователь `user` из домена `mydomain` и адрес сервера `mydomain.ru`, то необходимо указать :

Ниже приведена часть файла `appsettings.json` с нужными параметрами.

```
/* Сервер для подключения к Active Directory.Используется
для валидации доменных параметров доступа и для поиска пользователей при
включенной LDAP аутентификации.Если
значение не указано, в Windows будет использован адрес текущего LDAP сервера,
если машина находится в домене. */
"LDAPServer": "mydomain.ru",

/* Домен для подключения к Active Directory.Используется
для поиска пользователей при включенной LDAP аутентификации.Если
значение не указано, в Windows будет использовано имя текущего домена, если
машина находится в домене. */
"LDAPDomain": "mydomain",
```

1.5. Установка https сертификата

- Разместить сертификат по пути без пробелов, например /home/user/cert.pfx. Выполнить команду:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --install-cert /home/user/cert.pfx
```

Эта команда скопирует сертификат в папку /home/user/.config/orbox-server/volume/orbox-server/ под именем cert и пропишет его в конфигурационном файле сервера контроля.

- Открыть конфигурационный файл сервера контроля командой:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --edit-config server
```

Параметр UseSSL установить в значение true. Если сертификат защищён паролем, то в параметре CertificatePassword указать пароль от сертификата.

1.6. Запуск и выключение сервера контроля

Запуск сервера контроля в режиме консольного приложения производится следующим образом:

Перейти в папку /home/user/.config/orbox-server/ и запустить скрипт с ключом --start:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --start
```

Если сервер контроля запускается на одном компьютере с сервером анализа, то нужно дополнительно указывать ключ --limit:

```
./setup-orbox-server-docker-x.x.x.x-lic.sh --start --limit
```

В этом режиме сервер контроля работает пока открыта консоль. Чтобы завершить работу нужно нажать сочетание клавиш Ctrl+C.

Запуск сервера контроля в фоновом режиме производится следующим образом:

Перейти в папку /home/user/.config/orbox-server/ и запустить скрипт с ключом --enable:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --enable
```

Если сервер контроля запускается на одном компьютере с сервером анализа, то нужно дополнительно указывать ключ --limit:

```
./setup-orbox-server-docker-x.x.x.x-lic.sh --enable --limit
```

При этом так же будет включен автозапуск при включении компьютера.

Чтобы выключить сервер контроля нужно выполнить команду:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --disable
```

Это также отключит автозапуск.

1.7. Обновление сервера контроля

Обновление сервера контроля производится следующим образом:

1. Остановить сервер контроля:

- Если сервер запущен как консольное приложение, то необходимо нажать CTRL+C;
- Если сервер запущен в фоновом режиме, то необходимо выполнить команду:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --disable
```

2. Скопировать на машину файлы с новой версией сервера контроля:

- setup-orbox-server-docker-y.y.y.y-lic.sh
- orbox-server-docker-y.y.y.y-lic.tar.gz

3. Для установки новой версии сервера контроля запустить команду:

```
./setup-orbox-server-docker-y.y.y.y-lic.sh --install
```

Команду нужно вызывать из под обычного пользователя. Вызов из-под root или через sudo не поддерживается.

4. Обновить конфигурационные файлы:

```
./setup-orbox-server-docker-y.y.y.y-lic.sh --update-config all
```

5. Выполнить повторную настройку сервера контроля:

```
./setup-orbox-server-docker-y.y.y.y-lic.sh --edit-config server
```

1.8. Удаление сервера контроля

Для удаления сервера контроля, с сохранением файлов конфигурации, необходимо выполнить команду:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --remove
```

Для полного удаления сервера контроля выполнить команду:

```
cd /home/user/.config/orbox-server/  
sudo ./setup-orbox-server-docker-x.x.x.x-lic.sh --purge
```

1.9. Логирование сервера контроля

Логи сервера контроля находятся в папке

/home/user/.config/orbox-server/volume/orbox-server/log

2. Сервер анализа

Для установки сервера анализа требуется установленный компонент Docker версии не ниже 20.10.24. Информацию по Docker смотри на официальном сайте docs.docker.com.

2.1. Установка Docker

На сервер анализа компонент Docker устанавливается по аналогии с сервером контроля, подробнее можно посмотреть в разделе 1.1.2. «Установка Docker» на стр. 4.

2.2. Установка сервера анализа

Важно! В разделе 2.2. «Установка сервера анализа» все команды выполняются из-под пользователя **user**, если не указано другое. Установка из-под пользователя **root** или с помощью **sudo** не поддерживается.

Установка сервера анализа производится следующим образом:

1. Если на компьютере установлена предыдущая версия ORBOX (не docker), то её требуется удалить. Следуйте шагам по удалению сервера анализа из инструкции для установленной версии.
2. Скопировать на машину, где планируется установить сервер анализа 2 файла:
 - `setup-orbox-analyzer-docker-x.x.x.x.sh`
 - `orbox-analyzer-docker-x.x.x.x.tar.gz`

3. Установить для скрипта `setup-orbox-analyzer-docker-x.x.x.x.sh` права на выполнение:

```
chmod +x ./setup-orbox-analyzer-docker-x.x.x.x.sh
```

```
./setup-orbox-analyzer-docker-x.x.x.x.sh --install
```

После успешной установки файл `setup-orbox-analyzer-docker-x.x.x.x.sh` будет скопирован в папку `/home/user/.config/orbox-analyzer/`.

4. Поместить файл лицензии `license.dat` рядом со скриптом `setup-orbox-analyzer-docker-x.x.x.x.sh` и выполнить команду:

```
./setup-orbox-analyzer-docker-x.x.x.x.sh --install-license license.dat
```

Эта команда скопирует файл лицензии в папку
/home/user/.config/orbox-analyzer/volume/orbox

5. Обновить конфигурационные файлы командой:

```
./setup-orbox-analyzer-docker-x.x.x.x.sh --update-config all
```

Установка завершена.

2.3. Настройка

2.3.1. Настройка локализации

Проверка текущей локализации

Для того чтобы проверить текущую локализацию нужно выполнить команду `locale`.

```
LANG=ru_RU.UTF-8
LANGUAGE=
LC_CTYPE="ru_RU.UTF-8"
LC_NUMERIC="ru_RU.UTF-8"
LC_TIME="ru_RU.UTF-8"
LC_COLLATE="ru_RU.UTF-8"
LC_MONETARY="ru_RU.UTF-8"
LC_MESSAGES="ru_RU.UTF-8"
LC_PAPER="ru_RU.UTF-8"
LC_NAME="ru_RU.UTF-8"
LC_ADDRESS="ru_RU.UTF-8"
LC_TELEPHONE="ru_RU.UTF-8"
LC_MEASUREMENT="ru_RU.UTF-8"
LC_IDENTIFICATION="ru_RU.UTF-8"
LC_ALL=
```

В данном случае видно, что установлена русская локаль (значение переменной `LANG` установлено в `ru_RU.UTF-8`). В случае английской локали должно быть установлено значение `en_US.UTF-8`.

Установленные в системе локали (доступные для выбора)

Посмотреть установленные в системе локали можно с помощью команды `locale -a`:

```
C
C.UTF-8
en_US.utf8
```

```
POSIX
ru_RU.utf8
```

Данный вывод означает, что в системе доступны и русская и английская локали. Если необходимая локаль отсутствует в списке, ее нужно установить согласно разделу «Установка новой локали в систему».

Установка новой локали в систему

Если в результате выполнения команды `locale -a`, нужная локаль отсутствует в списке, ее нужно установить:

1. Открыть файл `/etc/locale.gen` с правами root:

```
sudo nano /etc/locale.gen
```

В файле можно увидеть список доступных для установки локалей:

```
# This file lists locales that you wish to have built.
# You can find a list of valid supported locales at
# /usr/share/i18n/SUPPORTED, and you can add user
# defined locales to /usr/local/share/i18n/SUPPORTED.
# If you change this file, you need to rerun locale-gen.
# aa_DJ ISO-8859-1
# aa_DJ.UTF-8 UTF-8
# aa_ER UTF-8
# aa_ER@saaho UTF-8
# aa_ET UTF-8
# af_ZA ISO-8859-1
# af_ZA.UTF-8 UTF-8
# ak_GH UTF-8
# am_ET UTF-8
# an_ES ISO-8859-15
# an_ES.UTF-8 UTF-8
# anp_IN UTF-8
...
```

2. Раскомментировать нужную локаль, убрав `#` в начале строки.
3. Выполнить команду:

```
sudo locale-gen
```

4. Выполнить команду:

```
sudo update-locale
```

После выполнения данных шагов должна стать доступной новая локаль. Проверка доступности производится выполнением команды:

```
locale -a
```

Выбор локали

После генерации локалей нужно прописать её в файле `/etc/default/locale` и перезагрузиться. Прописывать нужно три переменные: `LANG`, `LC_ALL` и `LANGUAGE`:

```
LANG=ru_RU.UTF-8  
LANGUAGE=ru_RU.UTF-8  
LC_ALL=ru_RU.UTF-8
```

В данном случае выбирается русская локаль. Для выбора английской локали переменной «`LANG`» нужно присвоить значение «`en_US.UTF-8`».

2.3.2. Настройка сервера анализа

Настройки сервера анализа находятся в файле:

```
/home/user/.config/orbox-analyzer/volume/orbox/conf/analyzer.conf.
```

Для быстрого открытия этого файла на редактирование можно воспользоваться командой:

```
cd /home/user/.config/orbox-analyzer  
./setup-orbox-analyzer-docker-x.x.x.x.sh --edit-config analyzer
```

Настройка журналирования открываются с помощью команды:

```
cd /home/user/.config/orbox-analyzer  
./setup-orbox-analyzer-docker-x.x.x.x.sh --edit-config logger
```

Описание настроек сервера анализа:

- `ControlServerAddress1` – IP адрес основного сервера контроля;
- `ControlServerPort` – Порт для коммуникации с основным сервером контроля;
- `ControlServerCertificatePath1` – Путь до SSL сертификата основного сервера контроля;
- `ControlServerCertificateEcdh1` – Опциональный параметр, определяющий тип эллиптической кривой в случае использования соответствующего SSL сертификата для основного сервера контроля;
- `ControlServerAddress2` – IP адрес резервного сервера контроля. Если использование резервного сервера контроля не планируется, то эту настройку необходимо удалить или указать в ней IP адрес основного сервера контроля;

- ControlServerPort2 – Порт для коммуникации с резервным сервером контроля. Если использование резервного сервера контроля не планируется, то эту настройку необходимо удалить или указать в ней порт основного сервера контроля;
- ControlServerCertificatePath2 – Путь до SSL сертификата резервного сервера контроля;
- ControlServerCertificateEcdh2 – Опциональный параметр, определяющий тип эллиптической кривой в случае использования соответствующего SSL сертификата для резервного сервера контроля;
- UseHTTPS – Использовать (true) или нет (false) HTTPS для взаимодействия СА и СК;
- CountEmptyResponseToFlush – Количество пустых ответов от СК для очистки внутренних буферов;
- ThreadsCount – Размер пулла потоков создаваемых модулей;
- ColorComponentsLevelTesNumThreads – Количество потоков, которое будет использоваться в тесте «Уровень цветовых компонент»;
- VectorscopeTestNumThreads – Количество потоков, которое будет использоваться в тесте «Уровень насыщенности»;
- InterlacementDetectionTestNumThreads – Количество потоков, которое будет использоваться в тестах «Гребёнка» и «Тип развёртки»;
- VideoDecThreadCount – Количество потоков, которое будет использоваться в процессе декодирования кадров;
- VideoDecThreadTypeMap – Метод декодирования, который будет использоваться (slice или frame). Можно задать индивидуально для каждого кодека;
- FrameBufferMaxSize – Максимальное количество кадров в очереди;
- ParseSpeed – Параметр, регулирующий скорость чтения метаданных и их детализацию (чем меньше, тем быстрее считываются метаданные, но с меньшей детализацией);
- CompressionTestNumThreads – Количество потоков, которое будет использоваться в тесте «Артефакты сжатия»;
- CompressionBlockingTest_BatchSize – Количество кадров, которые будут обрабатываться параллельно в тесте «Артефакты сжатия»;
- ArtifactsTestUseYUVcomponentCheck – Включение/выключение проверки YUV компоненты в тесте на артефакты потери данных;
- DataLossArtifactsTest_BatchSize – Количество кадров, которые будут обрабатываться параллельно в тесте «Артефакты потери данных»;
- DataLossMacroblockTestNumThread – Количество потоков, которые будут использоваться в тесте «Артефакты потери данных в макроблоках»;
- DataLossMacroblockTest_BatchSize – Количество кадров, которые будут обрабатываться параллельно в тесте «Артефакты потери данных в макроблоках»;
- MemLimitInMB – Ограничение на память. Если количество свободной памяти меньше

чем указанное значение, то текущая задача завершается с ошибкой;

- AllowQsvDecoding – использовать QSV декодер если есть такая возможность. Всегда отключено для HDR файлов;
- DropoutLineThresholdValue – Параметр регулирующий метрику теста «Выпадающие линии». Чем значение ближе к нулю, тем более вероятно найти артефакт;

Параметры теста «Калибровочные таблицы»:

- TestCardResizeImageWidth – Ширина, до которой будет сжат кадр перед обработкой;
- TestCardResizeImageHeight – Высота, до которой будет сжат кадр перед обработкой;
- TestCardImageSimilarityPercent – Минимальный процент «схожести» между анализируемым кадром и шаблоном таблицы, для того чтобы принять решение о том, что задетектирована калибровочная таблица;
- TestCardImageDilatingSize – Специфический параметр, используемый в алгоритме;
- UseTestCardFastMode – Включение/выключение быстрого режима в тесте «Калибровочные таблицы»;
- TestCardTest_BatchSize – Количество кадров, которые будут обрабатываться параллельно в тесте «Калибровочные таблицы»;

Параметры теста «Микропланы»:

- ShotTransitionHistogramDiffThreshold – Параметр, устанавливающий допустимую разницу гистограммы между кадрами в тесте «Микропланы»;
- ShotTransitionMinimalSequenceLength – Минимальная длина последовательности кадров, которая признана содержащей некорректные результаты. В дальнейшем эти результаты будут удалены из итоговой последовательности;
- ShotTransitionSobelMetricThreshold – Специфический параметр, используемый в алгоритме, чем выше значение, тем больше допускается разность кадров;
- ShotTransitionMssimDiffThresholdValue – Параметр, используемый в алгоритме, чем выше значение, тем больше допускается разность кадров;

Параметры теста «Цветные кадры»:

- ColorFramesResizeImageWidth – ширина, до которой будет сжат кадр перед обработкой;
- ColorFramesResizeImageHeight – высота, до которой будет сжат кадр перед обработкой;

- `ColorFramesPossiblePixelsInaccuracy` – допустимая неточность при сравнении пикселей. То есть, если значения соответствующих компонент двух пикселей отличаются на значение, меньшее данного параметра, то пиксели считаются одинаковыми;
- `PossibleDifferentColorsInFrame` – допустимое количество различных цветов в кадре;
- `PercentOfDominantColor` – процент доминирующего цвета в кадре;

Параметры теста «Тестовый сигнал»:

- `AudioTestSignalTestCalcWindowShift` – сдвиг окна, в процентах;
- `AudioTestSignalTestPercentOfMagnitude` – процент мощности сигнала 1000 Гц, для того чтобы считать звук в анализируемом окне тестовым сигналом.

Параметры теста «Gamut-ошибки»:

- `GamutErrorsTest_Batchsize` – Количество кадров, которые будут обрабатываться параллельно;
- `GamutErrorsTest_NumThreads` – Количество потоков, которые будет обрабатывать каждый кадр параллельно;
- `GamutErrorsTest_LowPassFilter_NumThreads` – Количество потоков для низкочастотного фильтра;
- `GamutErrorsTest_HorizontalQuarterBandFilter_NumThreads` – Количество потоков для горизонтального фильтра;
- `GamutErrorsTest_FiltersActivation_LeftThreshold` – Левое пороговое значение для низкочастотного фильтра;
- `GamutErrorsTest_FiltersActivation_RightThreshold` – Правое пороговое значение для низкочастотного фильтра;
- `GamutErrorsTest_Dev_Mode_enabled` – Режим разработчика. Вы можете включить этот флаг для рендеринга неудачных кадров в отдельные изображения. Также для каждого кадра файла будет напечатана информация о количестве битых пикселей;
- `GamutErrorsTest_Temp_folder` – Папка, куда будут сохранены изображения;

Детектирование шума:

- `NoiseDetectionTest_BatchSize` – Количество кадров, которые будут обрабатываться параллельно;
- `NoiseDetectionTest_BlockSize` – Размер блока для теста «Детектирование шума»;
- `NoiseDetectionTest_NoiseThreshold` – Пороговое значение для теста «Детектирование шума»;

Очередь сегментов Hls:

- `HlsSegmentsQueue_MaxBufferSizeMb` – размер буфера очереди Hls сегментов;

- `HlsSegmentsQueue_MaxOutBlockSizeKb` – максимальный размер блока на выходе;

Параметры теста «Интегральная метрика качества»:

- `PictureQualityScore_DevModeEnabled` – включения режима разработчика;
- `PictureQualityScore_TestsWeights` – веса зависимых тестов для расчета качества;

Параметры теста «Корректность заголовка»:

- `TitleCorrectnessTest_Regex` – регулярное выражение для проверки заголовка в тесте «Корректность заголовка»;

Параметры теста «Несоответствие стерео/моно сигнала»:

- `MonoStereoTest_WindowSizeMsec` – размер окна для поиска дефектов;
- `MonoStereoTest_WindowShiftPercent` – значение сдвига для перемещения по окну;
- `MonoStereoTest_PercentMonoSamplesInWindow` – если процент моно-сэмплов в окне равен или больше этого значения, окно является моно;

Описание настроек журналирования(файл `logger.properties`):

- `logging.loggers.root.channel` – Название класса для журналирования;
- `logging.loggers.root.level` – Уровень детальности логов (`trace`, `debug`, `notice`, `information`, `warning`, `error`, `critical`, `fatal`);
- `logging.formatters.f1.class` – Класс для форматирования;
- `logging.formatters.f1.pattern` – Шаблон вывода сообщений в логах;
- `logging.formatters.f1.times` – Время, которое будет использоваться в логах;
- `logging.channels.c1.class` – Класс для вывода в консоль;
- `logging.channels.c1.formatter` – Указывает на использование класса `f1` для форматирования;
- `logging.channels.c2.class` – Класс для вывода в файл;
- `logging.channels.c2.path` – Путь к файлу логов;
- `logging.channels.c2.formatter` – Указывает на использование класса `f1` для форматирования;
- `logging.channels.c2.rotation` – Максимальный размер файла;
- `logging.channels.c2.archive` – Имя архива;
- `logging.channels.c2.compress` – Использовать сжатие;
- `logging.channels.c2.purgeCount` – Максимальное число заархивированных файлов;
- `logging.channels.splitter.class` – Класс для режима вывода;
- `logging.channels.splitter.channels` – Режим вывода логов (`c1` – вывод в консоль, `c2` – вывод в файл).

2.4. Установка https сертификата

- Разместить сертификат по пути без пробелов, например /home/user/cert.pem. Выполнить команду:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --install-cert /home/user/cert.pem
```

Эта команда скопирует сертификат в папку /home/user/.config/orbox-analyzer/volume/orbox/ под именем cert и пропишет его в конфигурационном файле сервера анализа.

Если сертификат сделан на основе эллиптических кривых, то дополнительно нужно указать тип кривой:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --install-cert /home/user/cert.pem  
secp384r1
```

где secp384r1 тип кривой.

- Открыть конфигурационный файл сервера анализа командой:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --edit-config analyzer
```

Параметр UseHTTPS установить в значение true.

Этот способ подходит только для установки сертификата для первого сервера контроля. Установка сертификатов для последующих выполняется вручную. Для этого нужно скопировать файл сертификата в /home/user/.config/orbox-analyzer/volume/orbox/, назвать его уникальным именем и прописать в конфигурационном файле сервера анализа путь до него в параметре ControlServerCertificatePath№, где № это порядковый номер сервера контроля. Если этот сертификат сделан на основе эллиптических кривых, то прописать тип кривой в параметре ControlServerCertificateEcdh№, где № это порядковый номер сервера контроля.

2.5. Запуск и выключение сервера анализа

Запуск сервера анализа в режиме консольного приложения производится следующим образом:

Перейти в папку /home/user/.config/orbox-analyzer/ и запустить скрипт с ключом --start:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --start
```

Если сервер анализа запускается на одном компьютере с сервером контроля, то нужно дополнительно указывать ключ `--limit`:

```
./setup-orbox-analyzer-docker-x.x.x.x.sh --start --limit
```

В этом режиме сервер анализа работает пока открыта консоль. Чтобы завершить работу нужно нажать сочетание клавиш `Ctrl+C`.

Запуск сервера анализа в фоновом режиме производится следующим образом:

Перейти в папку `/home/user/.config/orbox-analyzer/` и запустить скрипт с ключом `--enable`:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --enable
```

Если сервер анализа запускается на одном компьютере с сервером контроля, то нужно дополнительно указывать ключ `--limit`:

```
./setup-orbox-analyzer-docker-x.x.x.x.sh --enable --limit
```

При этом так же будет включен автозапуск при включении компьютера.

Чтобы выключить сервер анализа нужно выполнить команду:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --disable
```

Это также отключит автозапуск.

2.6. Обновление сервера анализа

Обновление сервера анализа производится следующим образом:

1. Остановить сервер анализа:

- Если сервер запущен как консольное приложение, то необходимо нажать `CTRL+C`;
- Если сервер запущен в фоновом режиме, то необходимо выполнить команду:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --disable
```

2. Скопировать на машину файлы с новой версией сервера анализа:

- `setup-orbox-analyzer-docker-y.y.y.y.sh`
- `orbox-analyzer-docker-y.y.y.y.tar.gz`

3. Для установки новой версии сервера анализа запустить команду:

```
./setup-orbox-analyzer-docker-y.y.y.y.sh --install
```

Команду нужно вызывать из под обычного пользователя. Вызов из-под root или через sudo не поддерживается.

4. Обновить конфигурационные файлы:

```
./setup-orbox-analyzer-docker-y.y.y.y.sh --update-config all
```

5. Выполнить повторную настройку сервера анализа:

```
./setup-orbox-analyzer-docker-y.y.y.y.sh --edit-config analyzer
```

2.7. Удаление сервера анализа

Для удаления сервера анализа, с сохранением файлов конфигурации, необходимо выполнить команду:

```
cd /home/user/.config/orbox-analyzer/  
./setup-orbox-analyzer-docker-x.x.x.x.sh --remove
```

Для полного удаления сервера анализа выполнить команду:

```
cd /home/user/.config/orbox-analyzer/  
sudo ./setup-orbox-analyzer-docker-x.x.x.x.sh --purge
```

2.8. Логирование сервера анализа

Логи сервера анализа находятся в папке:

```
/home/user/.config/orbox-analyzer/volume/orbox/logs/
```

, где user – имя вашего пользователя

3. Режим резервирования сервера контроля

3.1. Общая информация

В стандартном режиме сервер БД PostgreSQL обычно установлен на сервере контроля, и Сервера Анализа настроены на подключение только к одному Серверу Контроля.

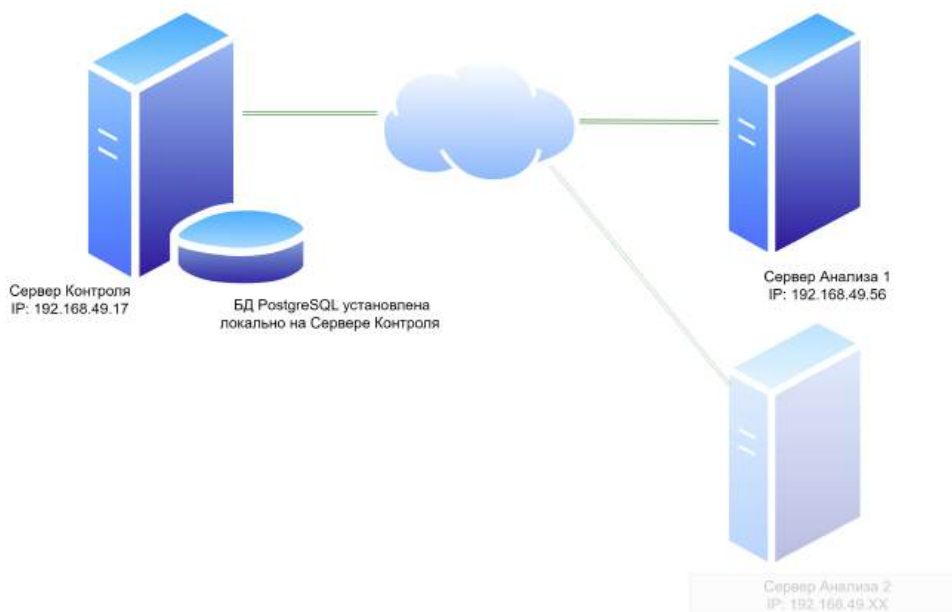


Рис. 3 – Схема подключения компонент системы ORBOX, работающей в стандартном режиме.

Режим резервирования сервера контроля позволяет быстро вернуть в работу систему ORBOX в случае выхода одного из Серверов Контроля из строя.

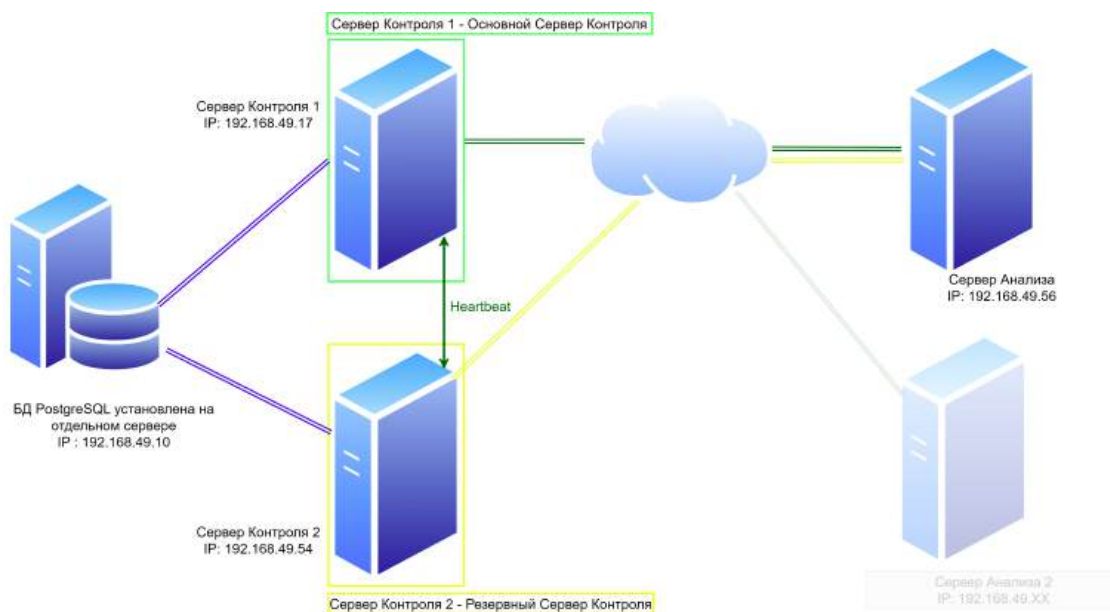


Рис. 4 – Схема подключения компонент системы ORBOX, работающей в режиме резервирования сервера контроля.

Для режима резервирования сервера контроля необходимо иметь:

- Сервер базы данных PostgreSQL, установленный на отдельном сервере (Рисунок 4).
- Два сервера с установленным ПО Сервер Контроля. Оба сервера необходимо настроить для работы в режиме резервирования сервера контроля. Один из серверов будет считаться основным, второй – резервным.
- Сервер Анализа, настроенный с возможностью подключения к 2-м Серверам Контроля.

3.2. Переключение режимов резервирования Сервера Контроля (Основной/Резервный)

Сервер Контроля работает в качестве основного до остановки или выхода из строя, после чего резервный Сервер Контроля переключится в режим основного.

Важно! Для доступа к веб-интерфейсу ORBOX используйте IP адрес Сервера Контроля, работающего в режиме Основного.

Важно! При начальном запуске системы, приоритет основного сервера отдается первому серверу, указанному в списке «ServerNodes» в файле конфигурации ПО Сервера Контроля (первый имеет высший приоритет).

3.3. Настройка серверов контроля для работы в режиме резервирования

Процесс установки ПО Сервера Контроля описан в разделе 1.2. на стр. 5. Предварительно необходимо установить Docker. Процесс установки Docker описан в разделе 1.1.2..

Важно! Все параметры в файлах конфигурации на обоих Серверах Контроля должны совпадать, кроме параметра "Port": "5003", этот порт может быть уникальным для каждого сервера контроля.

В процессе настройки серверов контроля для работы в режиме резервирования на каждом сервере контроля (на основном и резервном), в файле конфигурации (appsettings.json) необходимо изменить следующие настройки:

1. С помощью переменной DatabaseConnectionString необходимо настроить строку подключения к БД, указав таким образом, чтобы все серверы контроля, которые будут использованы в данной конфигурации резервирования, были подключены к одной и той же базе данных, находящейся на одной машине.

Важно! Для корректной работы режима резервирования необходимо установить базу данных PostgreSQL на отдельном сервере.

Пример:

```
"DatabaseConnectionString": "Host=192.168.49.10;Port=5432;Database=orbox;
  SearchPath=public;UserId=postgres;Password=postgres;CommandTimeout=0;
  EntityAdminDatabase=postgres;Include Error Detail=true;Log Parameters=
  true"
```

2. Включить репликацию серверов контроля с помощью параметра ServerReplicationEnabled:

```
"ServerReplicationEnabled": true,
```

3. Указать порт для синхронизации резервирования (для текущего узла) параметра Port в разделе Replication:

```
"Port": "5003",
```

(этот порт может быть одинаковым для всех серверов контроля, если они находятся на разных машинах)

4. Необходимо указать узлы резервирования сервера контроля в формате АДРЕС:ПОРТ, включая текущий узел. Порядок указания узлов соответствует приоритету при выборе основного узла (первый имеет высший приоритет).

```
"ServerNodes": [
  "192.168.49.17:5003",
  "192.168.49.54:5003"
],
```

Порт должен соответствовать порту, указанному в предыдущем пункте для каждого сервера контроля соответственно.

3.4. Настройка сервера анализа

Для работы системы в режиме резервирования необходимо указать все сервера контроля в конфиг файле сервера анализа. В случае если система будет работать по https, необходимо указать соответствующий ssl сертификат отдельно для каждого сервера контроля в конфиг файле сервера анализа.

Важно! При старте Сервера Анализа по очереди пробует подключаться к Серверам Контроля, указанных в файле конфигурации. В случае если связь между Сервером Контроля и Сервером Анализа будет прервана, то Сервер Анализа попытается подключиться к следующему Серверу Контроля, который указан в файле конфигурации.

```
# ip addresses, ports and ssl certificates of control servers
ControlServerAddress1 = 192.168.49.17
ControlServerPort1 = 8080
ControlServerCertificatePath1 = Certificate1.pem

# optional elliptic curve type
# ControlServerCertificateEcdh1 = secp384r1

ControlServerAddress2 = 192.168.49.54
ControlServerPort2 = 8080
ControlServerCertificatePath2 = Certificate2.pem

# optional elliptic curve type
# ControlServerCertificateEcdh2 = secp384r1

# use https for communication with control server
UseHTTPS = false
```

Файл конфигурации `analyzer.config` на Сервере Анализа (192.168.49.56)

3.5. Пример настройки системы ORBOX в режиме резервирования сервера контроля

Для примера, есть настроенная система ORBOX состоящая из 2-х серверов, работающая в стандартном режиме:

1. Сервер с установленным ПО Сервер Контроля ORBOX, далее СК1 (IP 192 . 168 . 49 . 17).
На сервере локально установлена БД PostgreSQL;
2. Сервер с установленным ПО Сервер Анализа ORBOX, далее СА: (IP 192 . 168 . 49 . 56);

Для включения режима резервирования сервера контроля необходимо добавить в систему два сервера:

1. Сервер для установки БД PostgreSQL, далее сервер БД (IP 192 . 168 . 49 . 10);
2. Сервер для установки ПО Сервер Контроля ORBOX, далее СК2 (IP 192 . 168 . 49 . 54);

Схема описанного примера показана ниже (Рис 5):

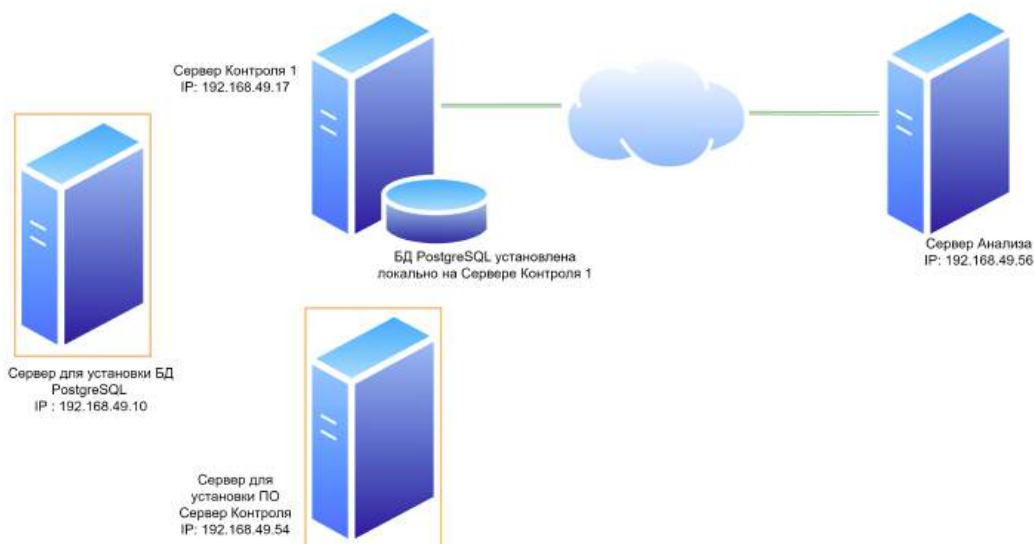


Рис. 5 – Схема подключения компонент системы ORBOX, работающей в стандартном режиме, и два дополнительных сервера.

Ниже приведены файлы настроек СК1 (192.168.49.17) и СА (192.168.49.56):

```
192.168.49.17# cat ${HOME}/.config/orbox-server/volume/orbox-server/conf/appsettings.json
```

```
/* Порт вебсервера- при использовании HTTP порты( с номером ниже 1024 требуют
   root прав на Linux) */
"HttpPort": 8080,

/* Строка подключения к базе данных. Если используется резервирование БД, Host
   и Port не указываются. */
"DatabaseConnectionString": "Host=localhost;Port=5432;Database=orbox;SearchPath=
public;User Id=postgres;Password=postgres;CommandTimeout=0;
EntityAdminDatabase=postgres;Include Error Detail=true;Log Parameters=true",

/* Резервирование */
"Replication": {

   /* Резервирование Сервера Контроля включено (true) или отключено (false) */
   "ServerReplicationEnabled": false,
```

```
},
```

Файл конфигурации `appsettings.json` на СК1 с подключением к локальной БД PostgreSQL, режим резервирования сервера контроля выключен.

```
192.168.49.56# cat ${HOME}/.config/orbox-analyzer/volume/orbox/conf/analyzer.config
```

```
# ip addresses, ports and ssl certificates of control servers
ControlServerAddress1 = 192.168.49.17
ControlServerPort1 = 8080
ControlServerCertificatePath1 = Certificate1.pem

# optional elliptic curve type
# ControlServerCertificateEcdh1 = secp384r1
```

Файл конфигурации `analyzer.config` на СА, настроено подключение только к СК1.

Для того чтобы включить режим резервирования сервера контроля, нам потребуется:

1. Настроить Сервер БД (IP 192.168.49.10):
 - 1.1. Установить БД PostgreSQL на Сервер БД;
 - 1.2. Разрешить удаленный доступ к БД PostgreSQL;
 - 1.3. Сделать бэкап БД на СК1;
 - 1.4. Восстанавливаем бэкап на Сервере БД.
2. Настроить СК1 (IP 192.168.49.17);
 - 2.1. Сменить подключение к локальной БД на подключение к Серверу БД;
 - 2.2. Включить режим резервирования сервера контроля на СК1;
 - 2.3. Проверить работоспособность СК1;
3. Настроить СА (IP 192.168.49.56);
 - 3.1. Настроить подключение к СК1 и СК2;
 - 3.2. Проверить работоспособность СК1+СА;
4. Настроить СК2 (IP 192.168.49.54);
 - 4.1. Установить ПО Сервер Контроля ORBOX на сервер СК2;
 - 4.2. Включить режим резервирования сервера контроля на СК1;
 - 4.3. Проверить систему;
5. Протестировать режим резервирования сервера контроля;

3.5.1. Настройка Сервера БД

Установка БД PostgreSQL на Сервер БД

Для начала необходимо установить БД PostgreSQL на Сервер БД (IP 192.168.49.10), пример установки PostgreSQL под ОС Debian 11 приведен в разделе 1.1.1..

Настройка удаленного доступа к БД PostgreSQL установленной на Сервере БД

После установки БД PostgreSQL, дополнительно требуется настроить доступ для внешних подключений.

Для настройки доступа в PostgreSQL-15 для подключения из нужной подсети XXX.YYY.ZZZ.0/24, необходимо поменять настройки в postgresql.conf и pg_hba.conf, указав, как показано ниже (показаны только нужные параметры, пути для файлов указаны для ОС Debian 11):

```
192.168.49.10# /etc/postgresql/15/main/postgresql.conf
```

```
# - Connection Settings -

listen_addresses = '*'      # what IP address(es) to listen on;
port = 5432                # (change requires restart)
```

```
192.168.49.10# /etc/postgresql/15/main/pg_hba.conf
```

```
#host all all XXX.YYY.ZZZ.0/24 md5
host all all 192.168.49.0/24 md5
```

Создание бэкапа БД на СК1

Для создания бэкапа БД с СК1 (IP 192.168.49.17) на Debian 11 используем утилиту pg_dump для базы orbox и запускаем из-под подпользователя (под которым создавалась база), для примера 'postgres'.

```
192.168.49.17#bash
```

```
sudo -u postgres pg_dump "orbox" > /tmp/backup_current.sql
```

Команда выполняется из-под пользователя root или пользователя включенного в группу sudo users. У пользователя 'postgres' может не быть прав доступа для записи в текущую папку, поэтому указываем папку /tmp.

Проверяем, что файл с бэкапом БД "orbox" /tmp/current_backup.sql создан и он не пустой.

```
192.168.49.17#bash ls -la
```

```
root@192.168.49.17:/tmp# ls -la /tmp
```

```
total 1740
drwxrwxrwt 12 root root    4096 Jul 24 09:13 .
drwxr-xr-x 18 root root    4096 Apr 23 03:37 ..
-rw-r--r--  1 root root 1731436 Jul 24 09:13 backup_current.sql
```

После чего можно скопировать файл бэкапа с СК1 на Сервер БД утилитой scp или любым удобным способом:

```
192.168.49.17#bash scp /tmp/backup_current.sql user@192.168.49.10:/tmp/
```

```
root@orbox-sup-sc-02:/tmp# scp /tmp/backup_current.sql user@192.168.49.10:/tmp/
user@192.168.49.10's password:
backup_current.sql
100% 1691KB 194.2MB/s 00:00
```

Восстанавливаем бэкап на Сервере БД

Для восстановления бэкапа БД "orbox" файл current_backup.sql должен находиться на Сервере БД (IP 192.168.49.10) в папке /tmp.

Восстановить бэкап можно только в существующую БД. Поэтому перед восстановлением запускаем утилиту psql и создаем БД:

```
192.168.49.10#bash
```

```
sudo -u postgres psql
```

В командной строке psql вводим команды для удаления базы данных и создания пустой базы данных "orbox":

```
192.168.49.10#psql
```

```
\1
CREATE DATABASE "orbox";
\1
\q
```

Запускаем утилиту psql с указанием имени базы данных и файла sql для восстановления:

```
192.168.49.10#bash
```

```
sudo -u postgres psql -d "orbox" -f /tmp/backup_current.sql
```

3.5.2. Настройка СК1

Меняем подключение с локальной БД на подключение к Серверу БД

Важно! На этом этапе можно отключить все источники или шаблоны, чтобы не было новых

заданий на обработку сразу после запуска Сервера Контроля.

На СК1 убеждаемся, что нет активных задач, и останавливаем СК1, для этого используем команды из раздела 1.6.:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --disable
```

Важно! После остановки СК1 веб-интерфейс будет недоступен!

На СК1 меняем подключение с локальной БД на подключение к Серверу БД, для этого меняем в конфигурационном файле Сервера Контроля в параметре DatabaseConnectionString адрес подключения с локального (localhost), смотри оригинальный файл 3.5., на IP удаленного Сервера БД (IP 192.168.49.10). Для изменения настроек Сервера Контроля выполняем команды, как указано в разделе 1.3.:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --edit-config server
```

```
192.168.49.17# cat ${HOME}/.config/orbox-server/volume/orbox-server/conf/appsettings.json
```

```
/* Порт вебсервера- при использовании HTTP порты( с номером ниже 1024  
требуют root прав на Linux) */  
"HttpPort": 8080,  
  
/* Использовать защищенное соединение (HTTPs) */  
"UseSSL": false,  
  
/* Строка подключения к базе данных. Если используется резервирование БД,  
Host и Port не указываются. */  
"DatabaseConnectionString": "Host=192.168.49.10;Port=5432;Database=orbox;  
SearchPath=public;User Id=postgres;Password=postgres;CommandTimeout=0;  
EntityAdminDatabase=postgres;Include Error Detail=true;Log Parameters=true",  
  
/* Резервирование */  
"Replication": {  
  
/* Резервирование Сервера Контроля включено (true) или отключено (false) */  
"ServerReplicationEnabled": false,  
},
```

Файл конфигурации appsettings.json на СК1 с подключением к Серверу БД (IP 192.168.49.10). Режим резервирования сервера контроля выключен.

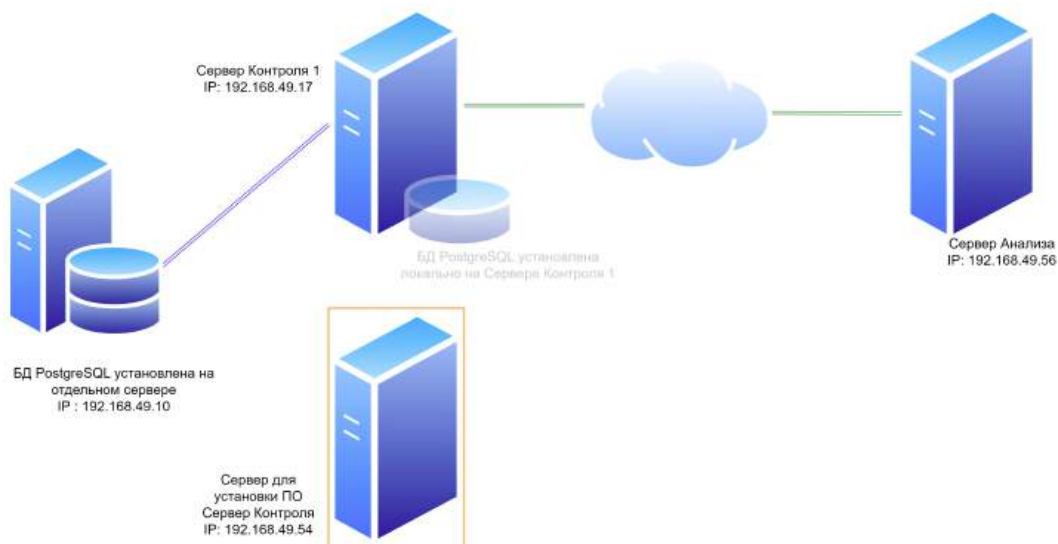


Рис. 6 – Схема подключения компонент системы ORBOX, работающей в стандартном режиме. СК1 подключен к Серверу БД, локальная БД не используется.

Включаем режим резервирования сервера контроля на СК1

На СК1 включаем режим резервирования сервера контроля с помощью параметра `ServerReplicationEnabled`. Для изменения настроек Сервера Контроля выполняем команды, как указано в разделе 1.3.:

```
cd /home/user/.config/orbox-server/
./setup-orbox-server-docker-x.x.x.x-lic.sh --edit-config server
```

Файл настроек на СК1 и СК2 должны быть одинаковыми (только параметр "Port" может быть уникальным для каждого из СК). Поэтому сразу указываем IP адреса обоих Серверов Контроля СК1 и СК2, СК1 указываем первым для большего приоритета при старте.

```
192.168.49.17# cat ${HOME}/.config/orbox-server/volume/orbox-server/conf/appsettings.json
```

```
/* Резервирование */
"Replication": {

  /* Резервирование Сервера Контроля включено (true) или отключено (false) */
  "ServerReplicationEnabled": true,

  /* Порт для синхронизации резервирования ( для текущего узла ) */
  "Port": "5003",

  /* Узлы резервирования Сервера Контроля в формате АДРЕС:ПОРТ, включая текущий узел.
  Порядок указания узлов соответствует приоритету при выборе основного узла
  первый( имеет высший приоритет). */
```

```
"ServerNodes": [  
    "192.168.49.17:5003",  
    "192.168.49.54:5003"  
],  
},
```

Файл конфигурации appsettings.json на СК1. Режим резервирования сервера контроля включен (добавили IP для 2-х серверов в список ServerNodes).

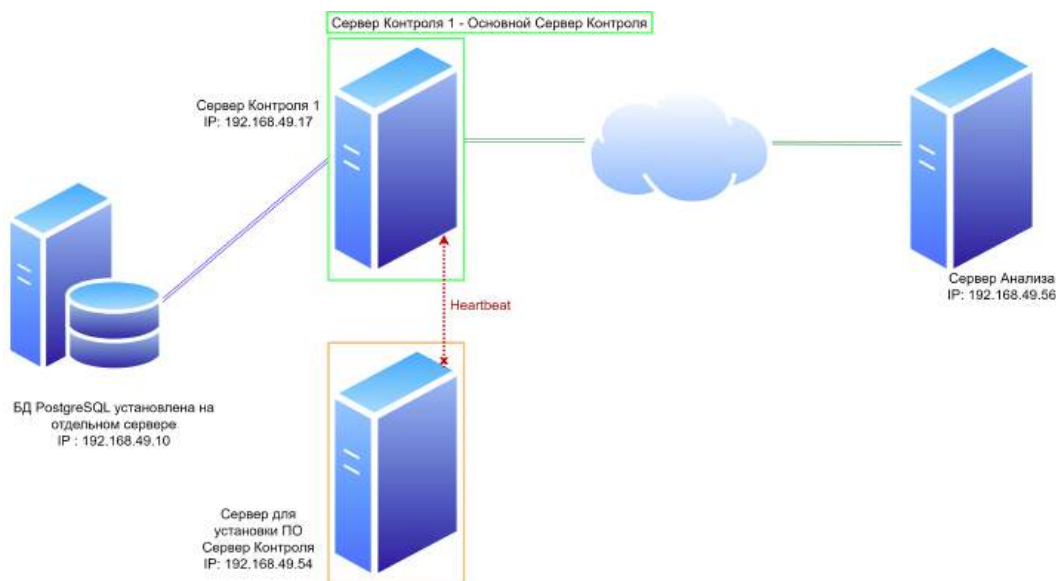


Рис. 7 – Схема подключения компонентов системы ORBOX. Режим резервирования сервера контроля включен на СК1. СК2 недоступен.

Проверяем работоспособность СК1

Запускаем СК1, для этого используем команды из раздела 1.6.:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --enable
```

Когда Сервер Контроля запускается в режиме резервирования сервера контроля, в веб-интерфейсе появляется блок мониторинга за состоянием режима резервирования на главной странице в разделе "Состояние системы".

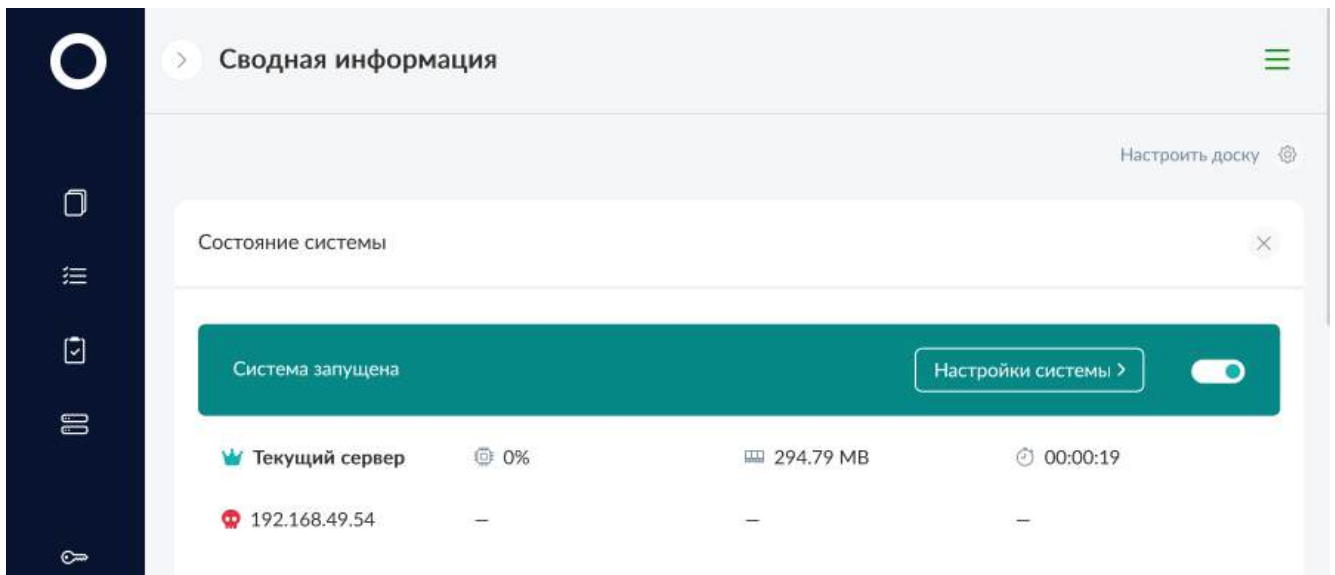





Рис. 8 – Главная страница веб-интерфейса системы ORBOX на СК1. Раздел Состояние системы. Режим резервирования сервера контроля на СК1 включен. СК2 недоступен. СК1 работает как основной Сервер Контроля.

Каждая строка отображает состояние настроенных для резервирования Серверов Контроля. Иконка в начале отображает статус сервера:

-  – Сервер работает в режиме основного.
-  – Сервер недоступен.
-  – Резервный сервер доступен для переключения в случае отказа основного сервера.

Проверяем, что СК1 подключается к удаленному серверу БД и восстановились все данные (Шаблоны, наборы Тестов, Источники, Назначения).

Проверяем, что все СА подключились к СК1, подробнее о странице Сервера Анализа смотри раздел ??.

3.5.3. Настройка СА

Поочередно на всех Серверах Анализа необходимо поменять настройки для работы с 2-мя СК. В примере меняем только на одном, если у вас несколько СА, то повторяем этот пункт 3.5.3. для всех Серверов Анализа.

Для изменения настроек нужно остановить СА, для этого используем команды из раздела 2.5.

```
cd /home/user/.config/orbox-analyzer/
./setup-orbox-analyzer-docker-x.x.x.x.sh --disable
```

На СА добавим IP для подключения к СК2 (IP 192.168.49.54), для этого меняем в конфигурационном файле Сервера Анализа в параметре ControlServerAddress2 адрес подключения на IP СК2 (IP 192.168.49.54). Для изменения настроек Сервера Анализа выполняем команды, как указано в разделе [2.3.2.](#):

```
cd /home/user/.config/orbox-analyzer
./setup-orbox-analyzer-docker-x.x.x.x.sh --edit-config analyzer
```

```
192.168.49.56# cat ${HOME}/.config/orbox-analyzer/volume/orbox/conf/analyzer.config
```

```
# ip addresses, ports and ssl certificates of control servers
ControlServerAddress1 = 192.168.49.17
ControlServerPort1 = 8080

ControlServerAddress2 = 192.168.49.54
ControlServerPort2 = 8080

# use https for communication with control server
UseHTTPS = false
```

Файл конфигурации `analyzer.config` на СА, настроено подключение к обоим Серверам Контроля СК1 и СК2.

Запускаем СА, для этого используем команды из раздела [2.5.](#):

```
cd /home/user/.config/orbox-analyzer/
./setup-orbox-analyzer-docker-x.x.x.x.sh --enable
```

После перезапуска СА проверяем, что СА подключился к СК1 после смены настроек, подробнее о странице Сервера Анализа смотри раздел ??.

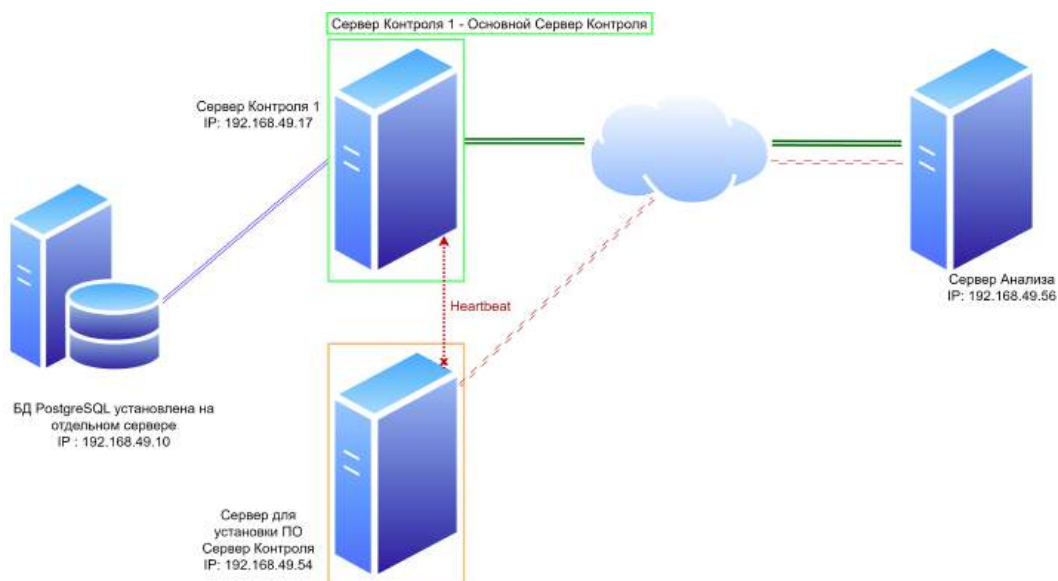


Рис. 9 – Схема подключения компонент системы ORBOX. Режим резервирования сервера контроля включен на СК1. СК2 недоступен. СА настроен на подключение к СК1 и СК2.

3.5.4. Настройка СК2

Устанавливаем ПО Сервер Контроля ORBOX на сервер СК2

Установка и настройка ПО Сервер Контроля описаны в разделе 1..

На сервер контроля СК2 (IP 192.168.49.54) нам нужно установить Docker, смотри подробнее в разделе 1.1.2. и ПО Сервер Контроля ORBOX, смотри подробнее в разделе 1.2..

Так как в режиме резервирования сервера контроля мы используем удаленный Сервер БД (IP 192.168.49.10), то шаги по установке БД PostgreSQL на сервере контроля пропускаем.

Настраиваем СК2

Для изменения настроек Сервера Контроля выполняем команды, как указано в разделе 1.3.:

```
cd /home/user/.config/orbox-server/
./setup-orbox-server-docker-x.x.x.x-lic.sh --edit-config server
```

Так как файлы настроек на СК1 и СК2 должны быть одинаковыми (только параметр "Port": "5003" может быть уникальным для каждого из СК), то все параметры настроек для Сервера Контроля СК2 (IP 192.168.49.54) берем из файла настроек Сервера Контроля СК1 (IP 192.168.49.17):

- Строка DatabaseConnectionString должна быть настроена на подключение к удаленному Сервера БД (IP 192.168.49.10);
- Параметр ServerReplicationEnabled должен быть включен;

- Список ServerNodes должен быть как на СК1 и содержать адреса обоих Серверов Контроля СК1 и СК2, СК1 должен быть указан первым для большего приоритета при старте.

Примечание: Не забываем, что остальные параметры должны быть также правильно настроены и совпадать с настройками указанными в настройках Сервера Анализа СА (IP 192.168.49.56).

192.168.49.54# cat \${HOME}/.config/orbox-server/volume/orbox-server/conf/appsettings.json

```
/* Порт вебсервера- при использовании HTTP порты( с номером ниже 1024 требуют
   root прав на Linux) */
"HttpPort": 8080,

/* Использовать защищенное соединение (HTTPS) */
"UseSSL": false,

/* Строка подключения к базе данных. Если используется резервирование БД, Host
   и Port не указываются. */
"DatabaseConnectionString": "Host=192.168.49.10;Port=5432;Database=orbox;
   SearchPath=public;User Id=postgres;Password=postgres;CommandTimeout=0;
   EntityAdminDatabase=postgres;Include Error Detail=true;Log Parameters=true",

/* Резервирование */
"Replication": {

   /* Резервирование Сервера Контроля включено (true) или отключено (false) */
   "ServerReplicationEnabled": true,

   /* Порт для синхронизации резервирования ( для текущего узла ) */
   "Port": "5003",

   /* Узлы резервирования Сервера Контроля в формате АДРЕС:ПОРТ, включая
   текущий узел.
   Порядок указания узлов соответствует приоритету при выборе основного узла (
   первый имеет высший приоритет ). */
   "ServerNodes": [
      "192.168.49.17:5003",
      "192.168.49.54:5003"
   ],
},
```

Файл конфигурации appsettings.json на СК2 с подключением к Серверу БД (IP 192.168.49.10). Режим резервирования сервера контроля включен.

Запускаем СК2, для этого используем команды из раздела 1.6.:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --enable
```

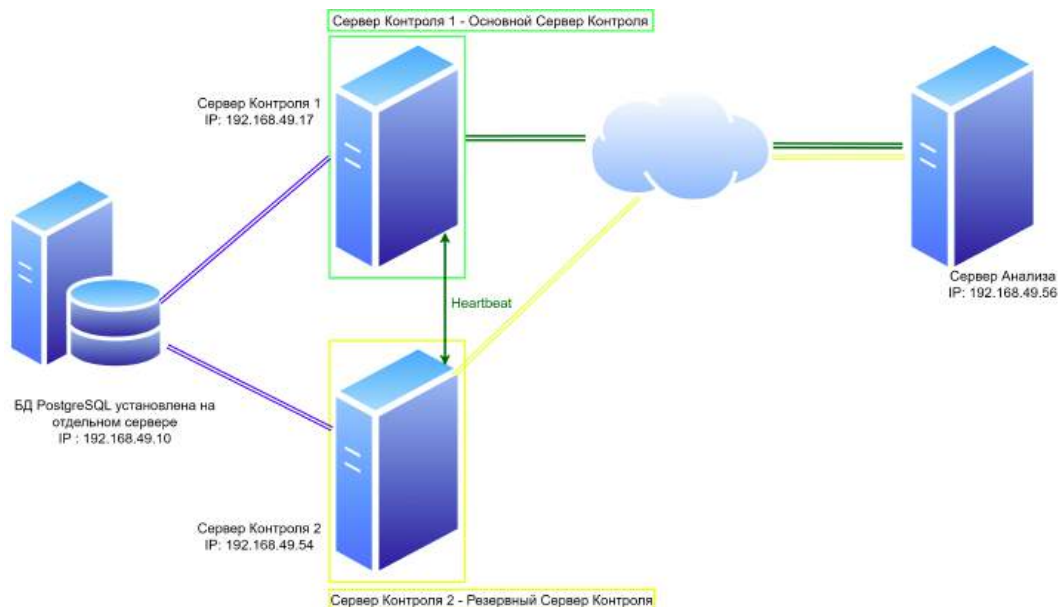


Рис. 10 – Схема подключения компонент системы ORBOX. Режим резервирования сервера контроля включен: СК1 основной, СК2 резервный. СА настроен на подключение к СК1 и СК2.

Проверяем состояние системы

Проверяем статус Серверов Контроля в веб-интерфейсе: СК1 (IP 192.168.49.17) должен остаться Основным. А иконка доступности Резервного Сервера СК2 (IP 192.168.49.54) контроля должна поменяться.

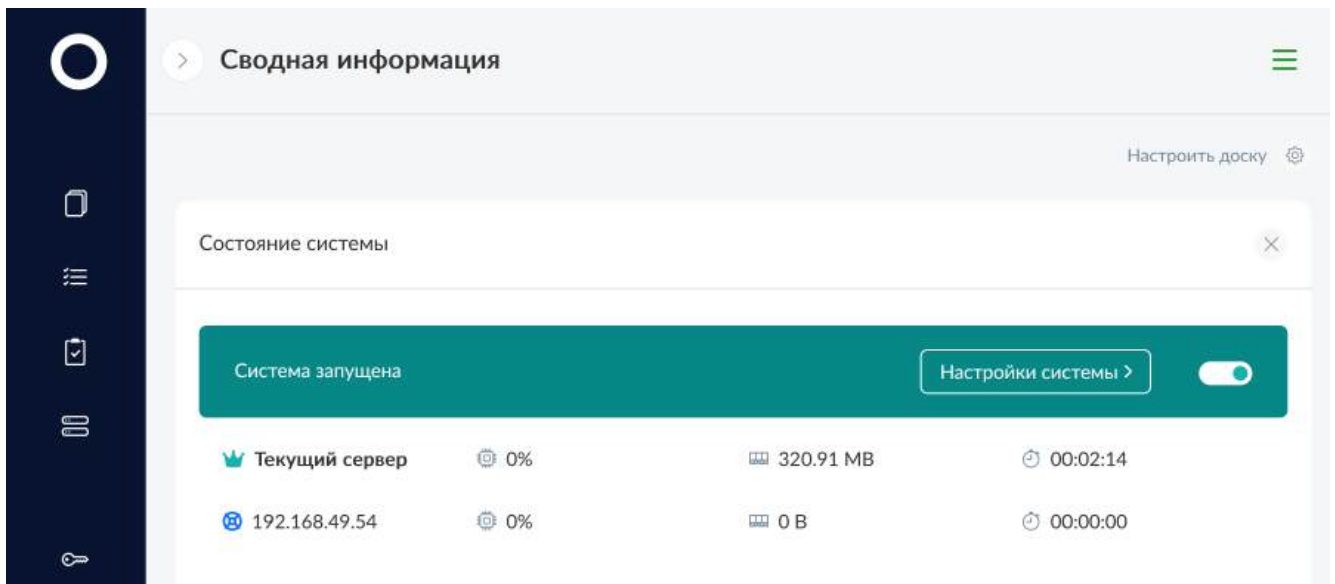


Рис. 11 – Главная страница веб-интерфейса системы ORBOX на СК1 - Состояние системы. Режим резервирования сервера контроля включен. СК1 работает как основной. СК2 подключен как резервный.

Если открыть веб-интерфейс Сервера Контроля СК2 (IP 192.168.49.54), там должна отображаться страница СЕРВЕР РАБОТАЕТ В РЕЖИМЕ РЕЗЕРВА:

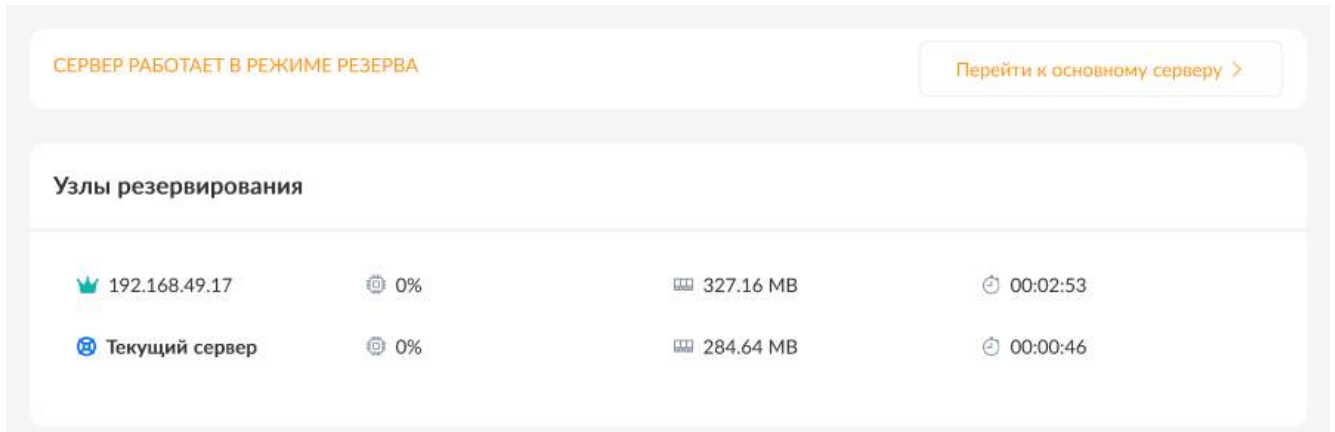


Рис. 12 – Главная страница веб-интерфейса системы ORBOX на СК2 в режиме резервирования. СК2 работает как резервный.

Важно! Если вы отключали источники или шаблоны в разделе 3.5.2., новые файлы будут взяты в работу и обработаны согласно настройкам приоритетов источников, подробнее в разделе ??.

На этом настройка примера 3.5. завершена, в следующем подразделе описан процесс проверки режима резервирования сервера контроля.

3.5.5. Тестирование работы режима резервирования сервера контроля

Для проверки режима резервирования сервера контроля остановим СК1 (IP 192.168.49.17). СК1 до остановки работал в режиме основного, а после остановки основного, резервный Сервер Контроля СК2 (IP 192.168.49.54) переключится из режима резервирования в основной режим. А все Сервера Анализа, в нашем случае СА (IP 192.168.49.56), подключатся к новому основному серверу СК2 (IP 192.168.49.54).

Для остановки СК1 (IP 192.168.49.17) используем команды из раздела 1.6.:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --disable
```

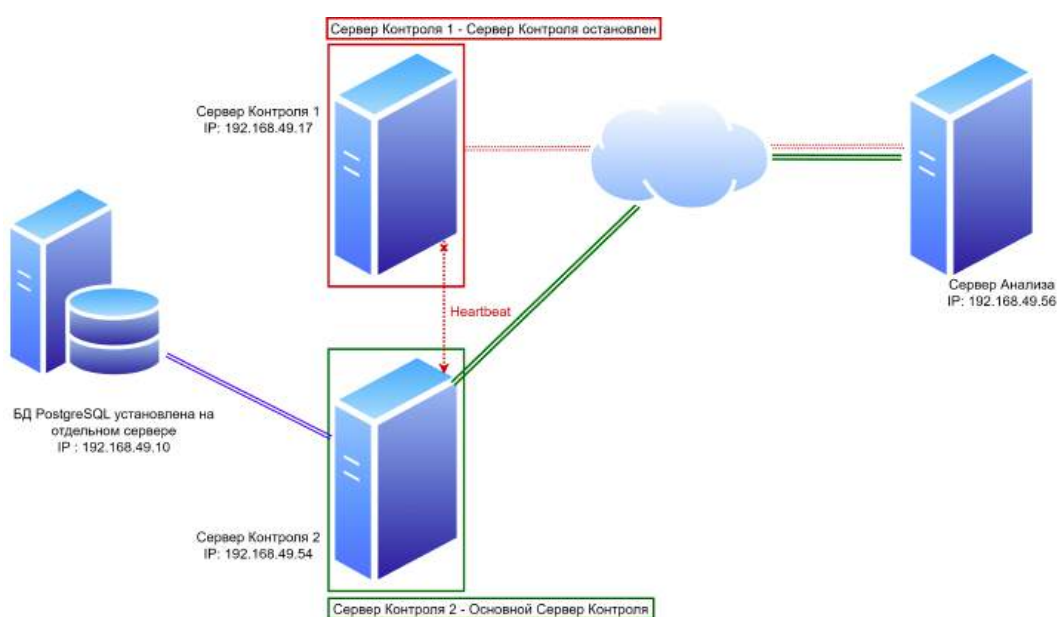
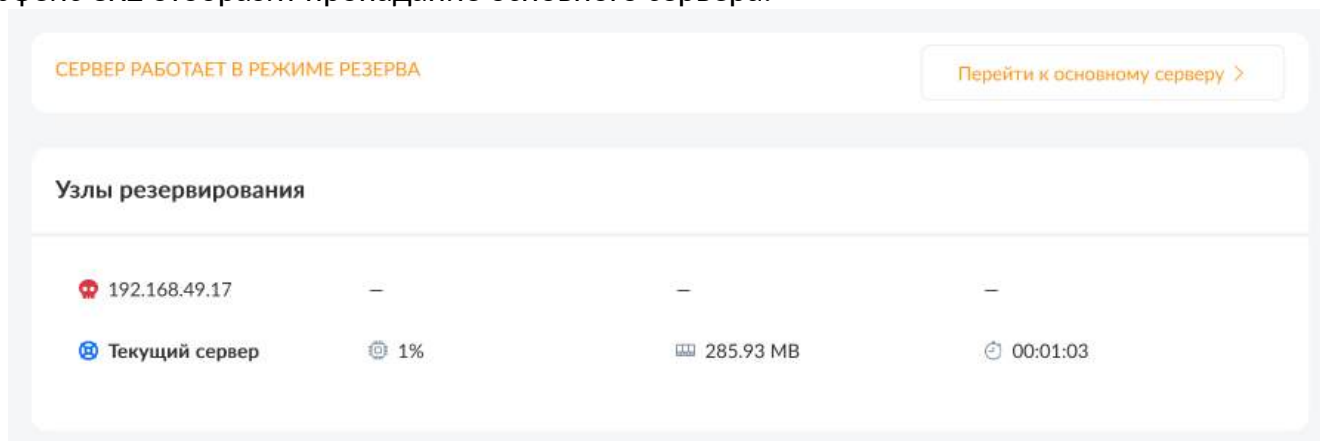
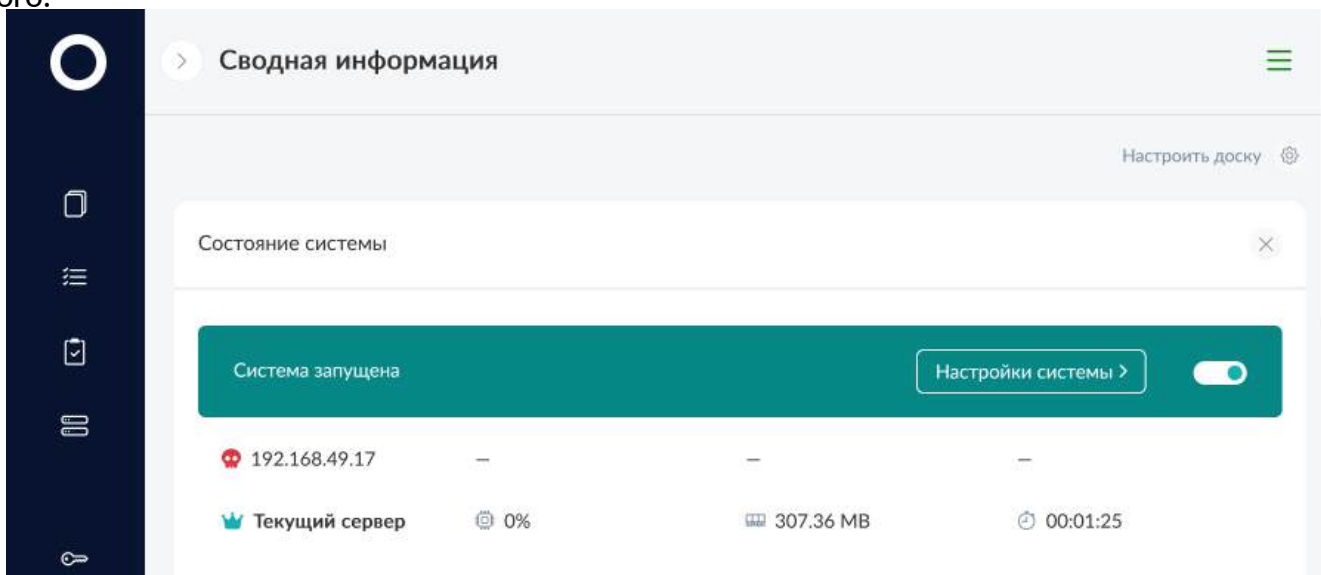


Рис. 13 – Схема подключения компонент системы ORBOX. Режим резервирования сервера контроля включен: СК2 основной, СК1 недоступен. СА настроен на подключение к СК1 и СК2.

Если открыть веб-интерфейс Сервера Контроля СК2 (IP 192.168.49.54), то сначала веб-интерфейс СК2 отобразит пропадание основного сервера:



Затем СК2 обновит страницу веб-интерфейса. Теперь на СК2, после ввода пароля, будет доступна полная версия веб-интерфейса Сервера Контроля, работающего в режиме основного.

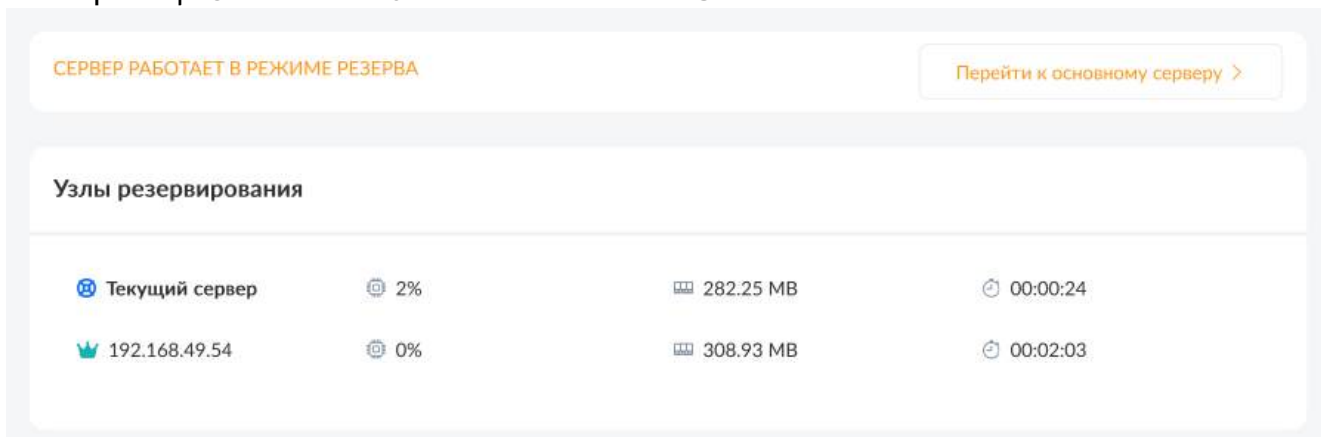


Проверим, что все СА доступны на странице Серверы Анализа в режиме, когда СК2 основной. Для проверки откроем в веб-интерфейсе страницу Сервера Анализа, подробнее о странице Сервера Анализа смотри раздел ??.

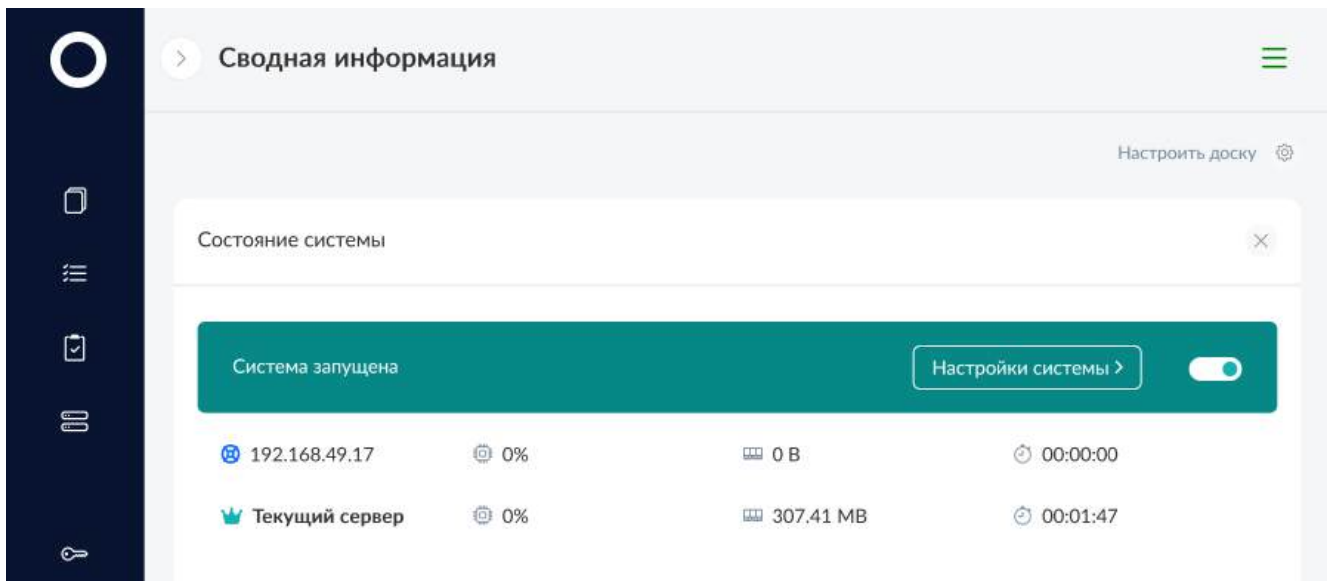
Запустим СК1 (IP 192.168.49.17) снова, для этого используем команды из раздела 1.6.:

```
cd /home/user/.config/orbox-server/  
./setup-orbox-server-docker-x.x.x.x-lic.sh --enable
```

Если открыть веб-интерфейс Сервера Контроля СК1 (IP 192.168.49.17), то будет отображаться страница СЕРВЕР РАБОТАЕТ В РЕЖИМЕ РЕЗЕРВА:



Если открыть веб-интерфейс Сервера Контроля СК2 (IP 192.168.49.54), то в веб-интерфейсе СК2 обновится информация о подключенном СК1:



Сервер Контроля СК2 будет работать в качестве основного до остановки или выхода из строя, после чего резервный Сервер Контроля СК1 переключится в режим основного.

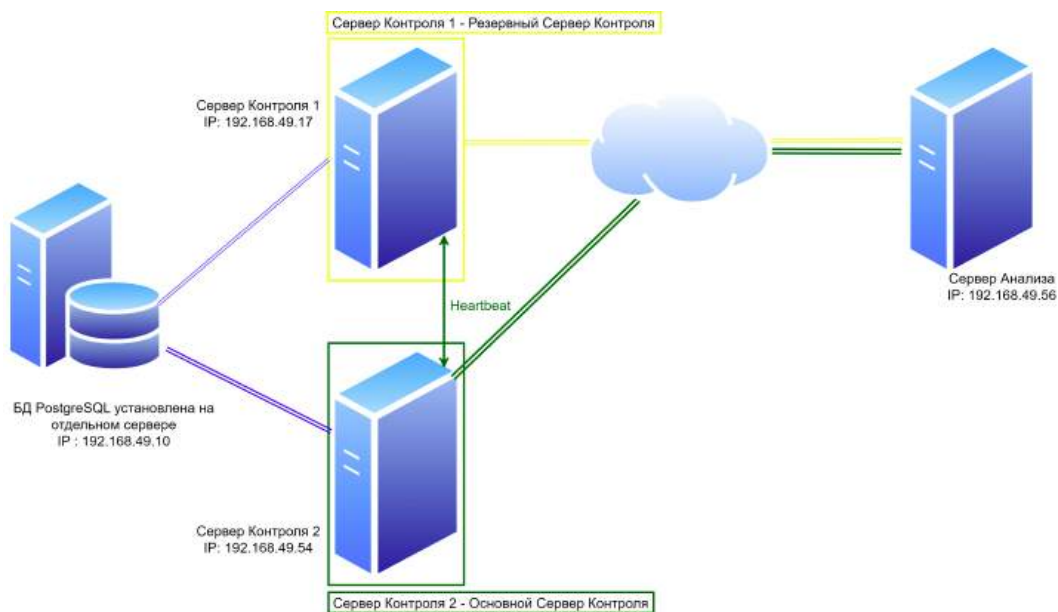


Рис. 14 – Схема подключения компонент системы ORBOX. Режим резервирования сервера контроля включен: СК2 основной, СК1 резервный. СА настроен на подключение к СК1 и СК2.

4. Архив брака

4.1. Общая информация

Архив брака позволяет сохранять участки видеофайлов, на которых был обнаружен брак, в процессе выполнения тестов.

4.2. Установка архива брака

Установка архива брака производится следующим образом:

1. Если на компьютере установлена предыдущая версия (не docker), то её требуется удалить.
2. Установить docker версии не ниже 20.10.24. Информацию по Docker можно посмотреть на официальном сайте docs.docker.com.

Важно! Нужно установить docker именно с официального сайта, потому что в репозитории вашего дистрибутива может быть старая версия.

После установки нужно добавить своего пользователя в группу docker и перезагрузить компьютер:

```
sudo usermod -aG docker user
```

где user - имя вашего пользователя.

3. Скопировать на машину, где планируется установить архив брака 2 файла:
 - setup-defects-archive-docker-x.x.x.x.sh
 - defects-archive-x.x.x.x.tar.gz
4. Установить для скрипта setup-defects-archive-docker-x.x.x.x.sh права на выполнение:

```
chmod +x ./setup-defects-archive-docker-x.x.x.x.sh
```

5. Сменить текущего пользователя на обычного. В данном руководстве предполагается, что этот пользователь имеет имя user. Установка из-под пользователя root или с помощью sudo не поддерживается. Для установки сервера анализа из папки, где находятся файлы, выполнить команду:

```
./setup-defects-archive-docker-x.x.x.x.sh --install
```

После успешной установки файл `setup-defects-archive-docker-x.x.x.x.sh` будет скопирован в папку `/home/user/.config/orbox-defects-archive/`.

6. Обновить конфигурационные файлы командой:

```
./setup-defects-archive-docker-x.x.x.x.sh --update-config all
```

Установка завершена.

4.3. Настройка

Настройки архива брака находятся в файле `orbox-defects-archive.conf`, который находится в папке `/home/user/.config/orbox-defects-archive/conf/`.

Для быстрого открытия этого файла на редактирование можно воспользоваться командой:

```
./setup-defects-archive-docker-x.x.x.x.sh --edit-config defects
```

В конфигурационном файле нужно указать `ip` адрес и порт сервера контроля.

Настройка архива брака на стороне сервера контроля происходит через веб-интерфейс системы ORBOX, в разделе «Общие настройки» (Рисунок 15).

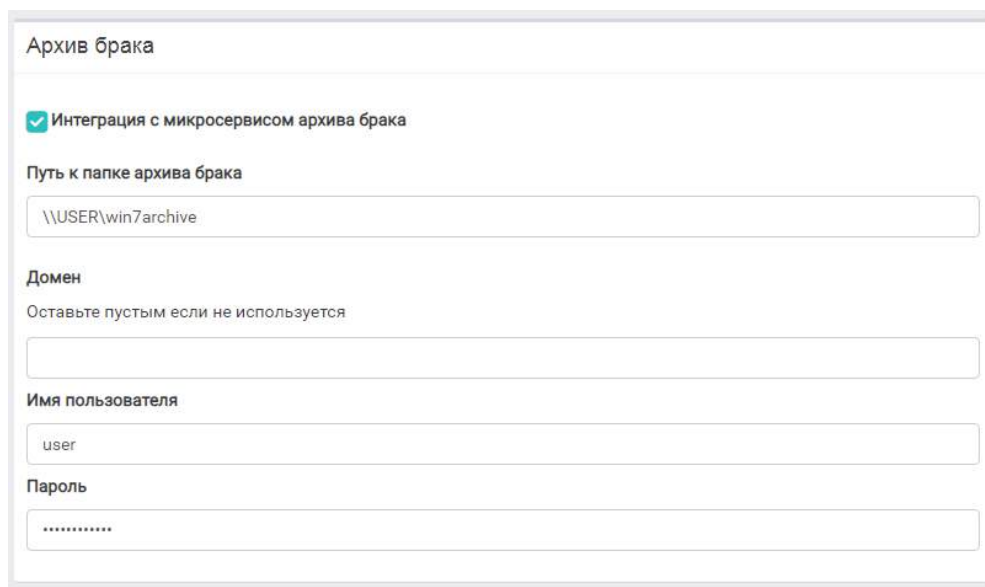


Рис. 15 – Общие настройки Архива брака

Чтобы выполнить настройку архива брака, необходимо задать следующие параметры:

- Интеграция с микросервисом архива брака - включить использование Архива брака.
- Домен – сетевой домен.

- Имя пользователя – имя пользователя для доступа к удаленной папке, где будут сохранены участки видеофайлов, на которых был обнаружен брак.
- Пароль – пароль пользователя для доступа к удаленной папке, где будут сохранены участки видеофайлов, на которых был обнаружен брак.

4.4. Установка https сертификата

- Разместить сертификат по пути без пробелов, например /home/user/cert.pem. Выполнить команду:

```
cd /home/user/.config/orbox-defects-archive/  
./setup-defects-archive-docker-x.x.x.x.sh --install-cert /home/user/cert.  
pem
```

Эта команда скопирует сертификат в папку /home/user/.config/orbox-defects-archive/volume/orbox-defects-archive/ под именем cert и пропишет его в конфигурационном файле архива брака.

- Открыть конфигурационный файл архива брака командой:

```
cd /home/user/.config/orbox-defects-archive/  
./setup-defects-archive-docker-x.x.x.x.sh --edit-config defects
```

Параметр useHttps установить в значение True.

Важно! Архив брака не поддерживает сертификаты на основе эллиптических кривых.

4.5. Запуск и выключение архива брака

Запуск архива брака в режиме консольного приложения производится следующим образом:

Перейти в папку /home/user/.config/orbox-defects-archive/ и запустить скрипт с ключом --start:

```
cd /home/user/.config/orbox-defects-archive/  
./setup-defects-archive-docker-x.x.x.x.sh --start
```

В этом режиме архив брака работает пока открыта консоль. Чтобы завершить работу нужно нажать сочетание клавиш Ctrl+C.

Запуск архива брака в фоновом режиме производится следующим образом:

Перейти в папку /home/user/.config/orbox-defects-archive/ и запустить скрипт с ключом --enable:

```
cd /home/user/.config/orbox-defects-archive/  
./setup-defects-archive-docker-x.x.x.x.sh --enable
```

При этом так же будет включен автозапуск при включении компьютера.
Чтобы выключить архив брака нужно выполнить команду:

```
cd /home/user/.config/orbox-defects-archive/  
./setup-defects-archive-docker-x.x.x.x.sh --disable
```

Это также отключит автозапуск.

5. ORBOX плеер

5.1. Общие требования

5.2. Первичная установка

Для первичной установки плеера необходимо запустить инсталляционную программу `player-X.X.X.XX.exe` из директории с файлами управляющего сервера. Также данную программу можно скачать из веб-клиента по кнопке «Скачать плеер» на вкладке результатов анализа видео файла.

После запуска инсталляционного файла появится окно с предложением выбора языка установки программы. Для выбора доступны русский и английский языки (Рисунок 16).

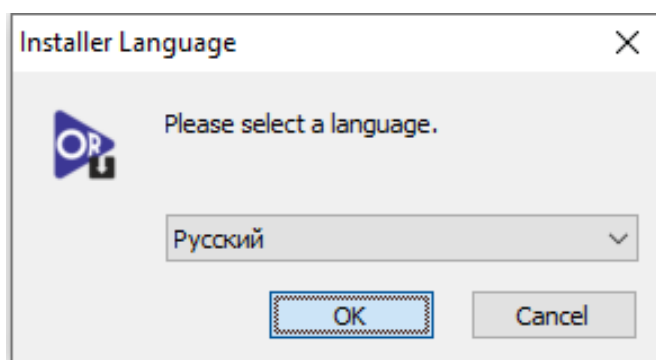


Рис. 16 – Окно выбора языка

После выбора языка нажать кнопку «ОК» для продолжения установки. После этого появится окно мастера установки программы (Рисунок 17).

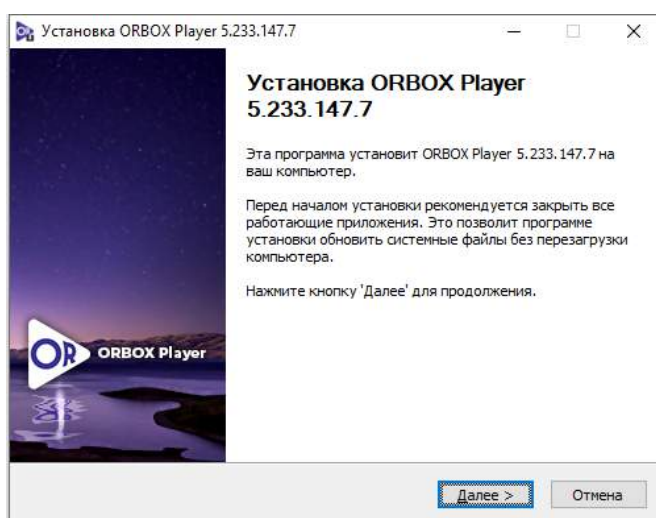


Рис. 17 – Мастер установки программы

Для продолжения нажать кнопку «Далее». После этого появится окно с предложением

выбрать компоненты приложения (Рисунок 18).

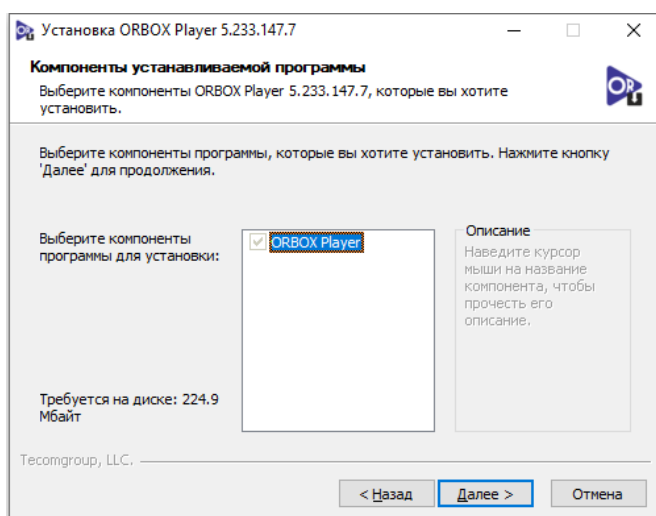


Рис. 18 – Окно выбора компонент программы

Выбрать компоненты для установки. После выбора необходимых компонентов нажать кнопку «Далее». После этого появится окно с выбором директории для установки приложения (Рисунок 19).

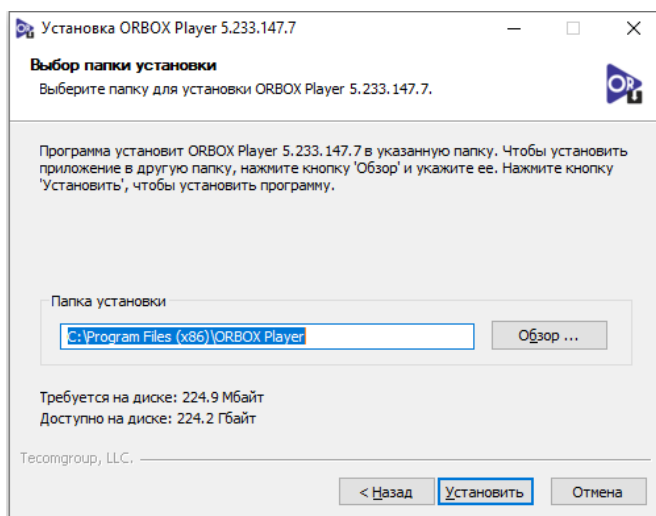


Рис. 19 – Окно выбора папки для установки

Нажать кнопку «Обзор», после нажатия на которую появится диалоговое окно «Обзор папок» (Рисунок 20). Директорию установки приложения также можно отредактировать вручную.

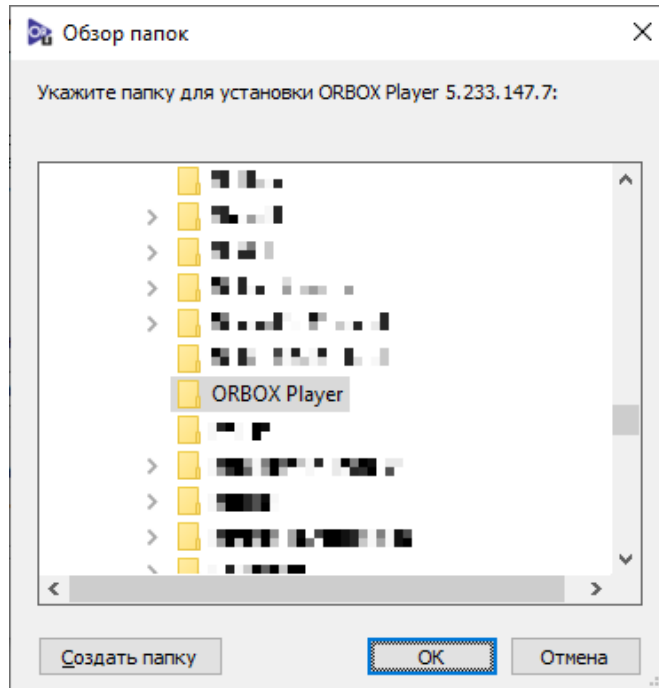


Рис. 20 – Окно «Обзор папок»

После выбора директории установки приложения нажать кнопку «Установить». После чего начнется процесс установки программы (Рисунок 21).

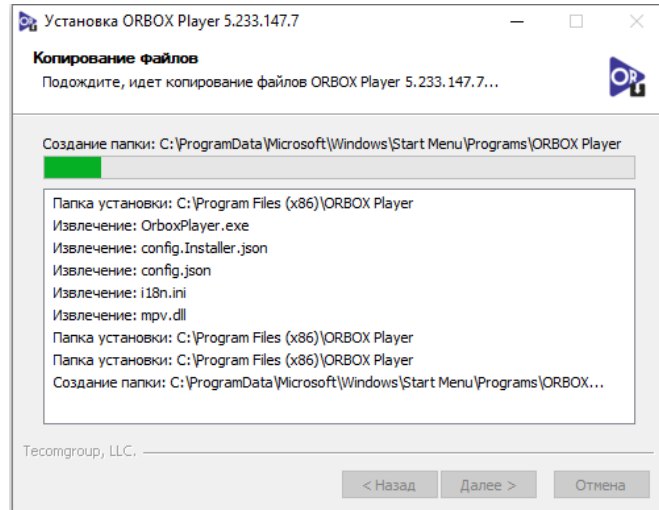


Рис. 21 – Процесс установки программ

После завершения процесса установки откроется окно с информацией о завершении установки программы (Рисунок 22).

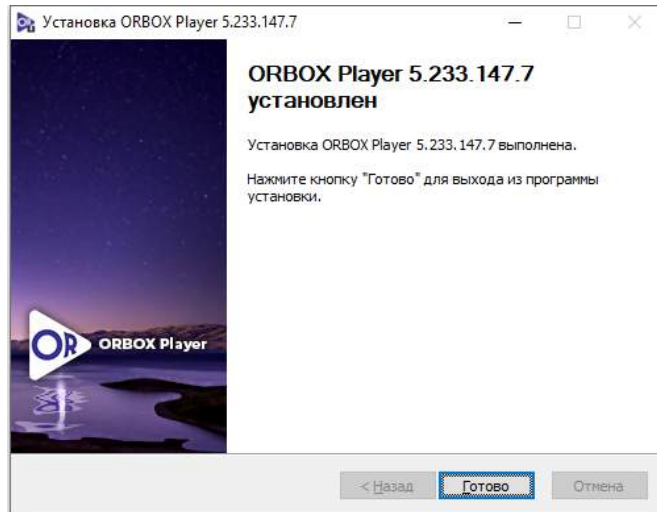


Рис. 22 – Завершение установки программы

Для завершения процесса установки необходимо нажать кнопку «Готово».

5.3. Повторная установка

5.3.1. Полная установка

Если программа уже была установлена ранее, необходимо выполнить следующие шаги для повторной установки.

Для установки плеера под ОС Windows необходимо запустить инсталляционную программу player-X.X.X.XX.exe из директории с файлами управляющего сервера. Также данную программу можно скачать из веб-клиента по кнопке «Скачать плеер» с вкладки результатов анализа видео файла. После запуска инсталляционного файла появится окно мастера установки программы (Рисунок 17).

Для продолжения нажать кнопку «Далее». После этого появится окно с выбором варианта установки (Рисунок 23).

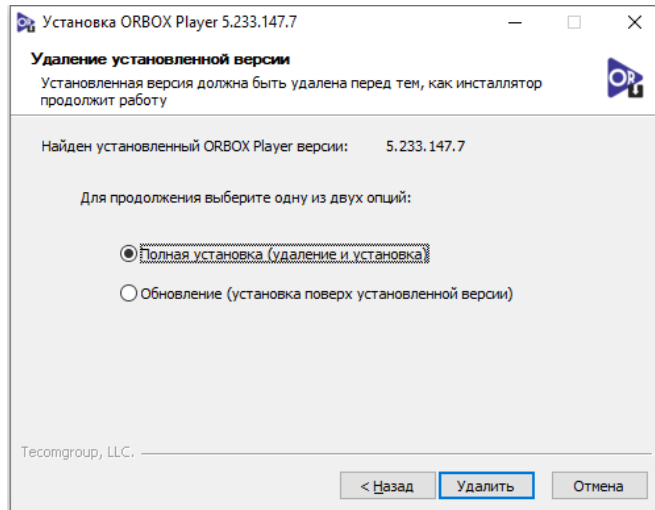


Рис. 23 – Окно выбора варианта установки

Для полной установки (удаление предыдущей версии и установка новой) необходимо выбрать пункт «Полная установка (удаление и установка)», после чего нажать кнопку «Удалить». После этого появится диалоговое окно с подтверждением удаления программы (Рисунок 24).

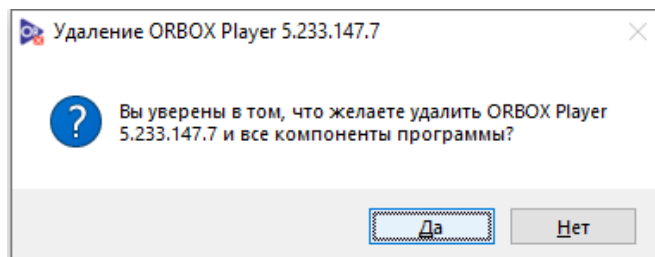


Рис. 24 – Подтверждение удаления установленной версии программы

Для подтверждения удаления нажать кнопку «Да». Начнется процесс удаления программы. После завершения процесса удаления откроется окно с информацией о завершении удаления программы (Рисунок 25).

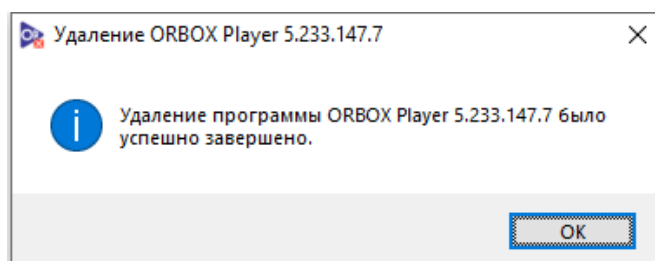


Рис. 25 – Завершение удаления программы

Для завершения процесса удаления старой версии программы и начала установки новой нажать кнопку «ОК». Дальнейший процесс установки аналогичен процессу первичной

установки, описанному в главе «Первичная установка».

5.3.2. Обновление программы

При обновлении ранее установленной программы, необходимо выполнить следующие шаги. Для установки плеера из директории с файлами управляющего сервера необходимо запустить инсталляционную программу `player-X.X.X.XX.exe`. Также данную программу можно скачать из веб-клиента по кнопке «Скачать плеер» с вкладки результатов анализа видеофайла.

После запуска инсталляционного файла появится окно мастера установки программы (Рисунок 17).



Рис. 26 – Мастер установки программ

Для продолжения необходимо нажать кнопку «Далее». После этого появится окно с выбором варианта установки (Рисунок 27).

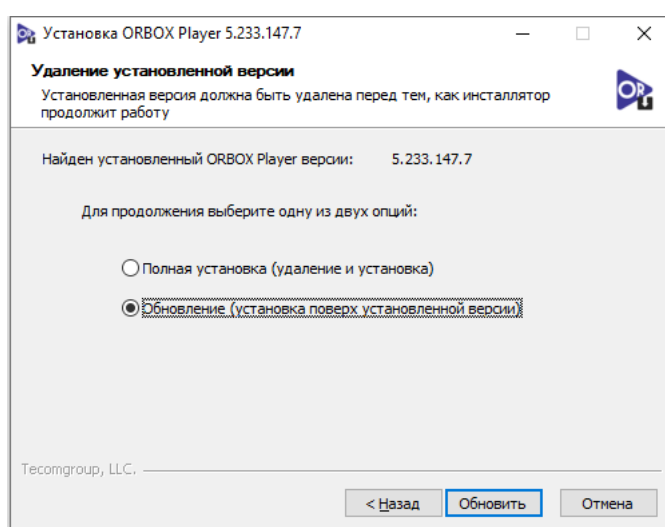


Рис. 27 – Завершение удаления программы

Для обновления текущей версии программы необходимо выбрать опцию «Обновление (установка поверх установленной версии)» и нажать кнопку «Обновить». В данном случае текущая версия приложения будет обновлена. Дальнейший процесс установки аналогичен процессу первичной установки, описанному в разделе «Первичная установка».

6. Adobe Premier Plugin

6.1. Установка и настройка ORBOX Connector

Установка и настройка ORBOX Connector производится следующим образом:

1. Распаковать архив WindowsInstaller-x.x.x.x.zip в любое место на ПК (Рисунок 28):

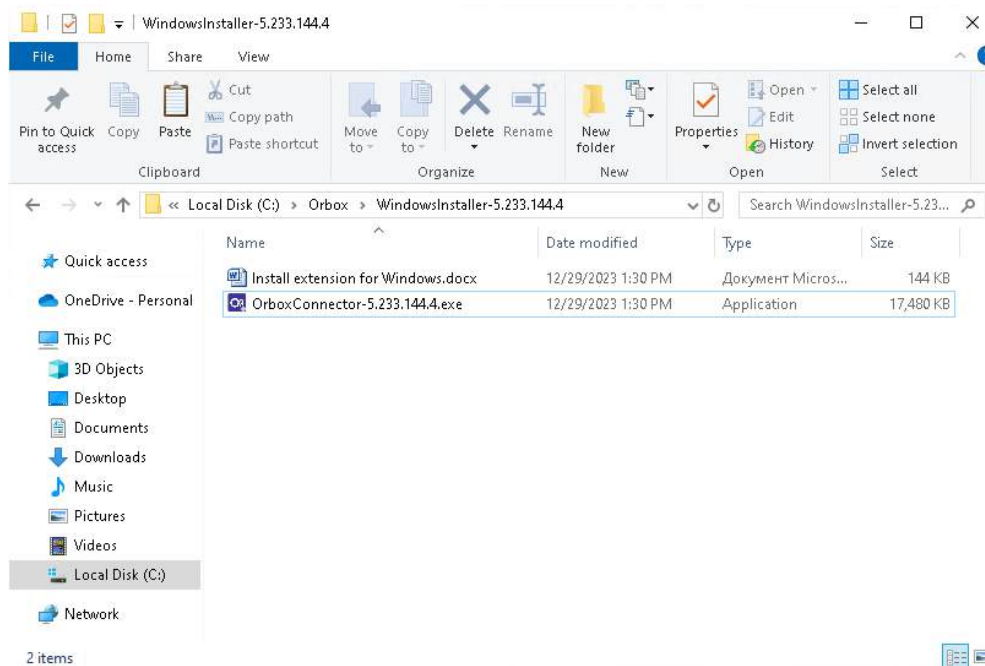


Рис. 28 – Содержимое архива WindowsInstaller-x.x.x.x.zip

2. Запустить установщик, и пройти все предложенные шаги (Рисунок 29, Рисунок 30):



Рис. 29 – Начало установки ORBOX connector

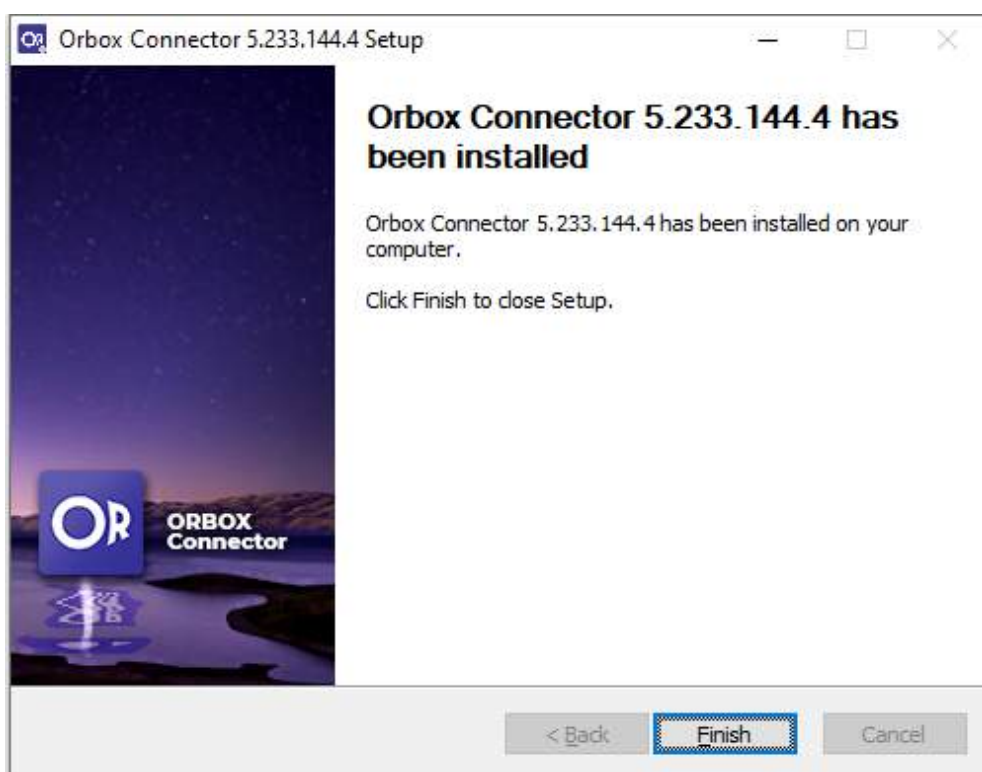


Рис. 30 – Окончание установки ORBOX connector

3. Для получения результатов анализа плагин связывается с Сервером Контроля. В зависимости от настроек Сервера Контроля взаимодействие может происходить по

протоколу HTTP или HTTPS. Необходимые настройки необходимо вносить в конфигурационный файл плагина config.json. Файл располагается в директории C:\ProgramData\OrboxConnector.

- (a) Для соединения по HTTP, необходимо указать IP адрес и порт Сервера Контроля в поле controlServerHost конфигурационного файла плагина config.json.

Например:

```
{
  "version": 1,
  "controlServerHost": "http://192.168.0.1:8080"
  "caChainCert": [],
  "logLevel": "debug",
  "maxLogSize": 52428800,
  "maxLogFiles": 3,
  "consoleLogger": true,
  "language": "en",
  "audioTestsNames": [
    "LoudnessTest",
    "SilenceTest",
    "AudioPhaseTest",
    "ClippingTest",
    "AudioTestSignalTest",
    "NoAudioSignalTest",
    "MonoStereoTest"
  ]
}
```

- (b) Для соединения по HTTPS, также, как и при соединении по HTTP, необходимо указать IP адрес и порт Сервера Контроля. Помимо этого, если используются сертификаты, сгенерированные вручную, а не полученные в центрах сертификации, то будет необходимо указать путь к файлу с цепочкой доверенных сертификатов в поле caChainCert. Цепочка доверенных сертификатов представляет собой файл, в котором содержится информация по корневому и промежуточным сертификатам, которые участвовали в подписании конечного сертификата Сервера Контроля. Способ генерации сертификатов средствами openssl описан в разделе ?? FAQ

Например:

```
{
  "version": 1,
  "controlServerHost": "https://192.168.0.1:8081",
  "caChainCert": [
    "C:\\certs\\ca-chain.cert.pem"
  ],
  "logLevel": "debug",
  "maxLogSize": 52428800,
  "maxLogFiles": 3,
  "consoleLogger": true,
  "language": "en",
}
```

```
"audioTestsNames": [  
  "LoudnessTest",  
  "SilenceTest",  
  "AudioPhaseTest",  
  "ClippingTest",  
  "AudioTestSignalTest",  
  "NoAudioSignalTest",  
  "MonoStereoTest"  
]  
}
```

Также, можно указать пути к промежуточному и корневому сертификатам по отдельности, для этого в поле `caChainCert` указать пути к файлам через запятую:

```
{  
  "caChainCert": [  
    "C:\\Users\\voidMain\\Desktop\\intermediate.cert.pem",  
    "C:\\Users\\voidMain\\Desktop\\ca.cert.pem"  
  ]  
}
```

7. Высокодоступный кластер PostgreSQL

7.1. Общие данные

В данном руководстве приведён пример развёртывания высокодоступного кластера PostgreSQL, состоящего из трёх узлов на операционной системе Debian 11. Данная конфигурация позволяет пережить выход из строя любого одного узла. Если требуется дополнительно повысить устойчивость, то нужно увеличить количество узлов в кластере.

7.2. Составные части кластера

PostgreSQL – это мощная и расширяемая система управления базами данных (СУБД), которая предоставляет надёжное хранение и обработку структурированных данных.

Patroni – это инструмент высокой доступности (High Availability, HA) для системы управления базами данных PostgreSQL. Он предоставляет механизмы автоматического переключения и отказоустойчивости для кластера PostgreSQL.

EtcD – это распределенное хранилище ключ-значение, которое используется для хранения и управления конфигурационными данными и метаданными в распределённых системах.

HAProxy (High Availability Proxy) – это программное обеспечение, которое предоставляет балансировку нагрузки и проксирование для протоколов TCP и HTTP. Оно используется для распределения входящих запросов между несколькими серверами, чтобы обеспечить высокую доступность и улучшить производительность веб-приложений.

7.3. Настройка доменных имён

Для универсальности во всех конфигурационных файлах используются доменные имена вместо IP адресов. Для узлов, на которых установлены экземпляры PostgreSQL, используются имена `orbox-psql1`, `orbox-psql2` и `orbox-psql3`. Для узлов EtcD используются имена `orbox-etcD1`, `orbox-etcD2` и `orbox-etcD2`.

Чтобы операционная система могла разрешать эти доменные имена нужно отредактировать файл `/etc/hosts` и добавить записи сопоставления между IP адресами и доменными именами.

Предположим, что на каждом узле установлено по одному экземпляру `etcD`, `postgresql`, `patroni` и `haproxy`, тогда в файл `/etc/hosts` нужно добавить следующие записи (IP адреса даны для примера и при развёртывании их нужно заменить на реальные адреса узлов):

```
172.17.0.2 orbox-etcD1
172.17.0.3 orbox-etcD2
172.17.0.4 orbox-etcD3
```

```
172.17.0.2 orbox-psql1
172.17.0.3 orbox-psql2
172.17.0.4 orbox-psql3
```

orbox-etcd1 и **orbox-psql1** имеют один и тот же адрес, потому что Etcd установлен на том же узле, что и PostgreSQL. Если экземпляры Etcd вынесены на отдельные узлы, то у них будут другие IP адреса, не совпадающие с адресами PostgreSQL.

7.4. Установка и настройка Etcd

На каждом из узлов **orbox-etcd1**, **orbox-etcd2** и **orbox-etcd3** выполнить установку Etcd, а также создать папку для хранения данных:

```
apt update
apt install -y --no-install-recommends etcd

mkdir -p /var/lib/etcd/orbox-psql
chown -R etcd:etcd /var/lib/etcd/orbox-psql
chmod 0700 /var/lib/etcd/orbox-psql
```

На узле **orbox-etcd1** создать конфигурационный файл **/etc/default/etcd** со следующим содержимым:

```
ETCD_NAME=orbox-etcd1
ETCD_INITIAL_CLUSTER="orbox-etcd1=http://orbox-etcd1:2380,orbox-etcd2=http://
  orbox-etcd2:2380,orbox-etcd3=http://orbox-etcd3:2380"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="orbox-psql-ha-cluster"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://orbox-etcd1:2380"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ADVERTISE_CLIENT_URLS="http://orbox-etcd1:2379"
ETCD_DATA_DIR="/var/lib/etcd/orbox-psql"
```

На узлах **orbox-etcd2** и **orbox-etcd3** также создать подобные конфигурационные файлы, но заменить в параметрах **ETCD_NAME**, **ETCD_INITIAL_ADVERTISE_PEER_URLS**, **ETCD_ADVERTISE_CLIENT_URLS** строку **orbox-etcd1** на **orbox-etcd2** и **orbox-etcd3** для узлов **orbox-etcd2** и **orbox-etcd3** соответственно.

Запуск Etcd (выполнить на всех узлах с Etcd):

```
systemctl enable etcd
systemctl restart etcd
```

Для проверки состояния кластера Etcd выполнить на любом узле с Etcd команду:

```
ETCDCTL_API=3 etcdctl endpoint status --cluster -w table
```

Потребуется некоторое время на сборку кластера, поэтому вначале команда может завершаться ошибкой. После успешного запуска вывод команды должен быть примерно следующим:

```
+-----+-----+-----+-----+-----+-----+
|          ENDPOINT          | VERSION | DB SIZE | IS LEADER | RAFT TERM | RAFT
| INDEX |
+-----+-----+-----+-----+-----+-----+
| http://orbox-etcd1:2379 | 3.3.25 | 20 kB | true | 2 |
| 9 |
| http://orbox-etcd2:2379 | 3.3.25 | 20 kB | false | 2 |
| 9 |
| http://orbox-etcd3:2379 | 3.3.25 | 20 kB | false | 2 |
| 9 |
+-----+-----+-----+-----+-----+-----+
```

Далее нужно отредактировать на всех узлах файл `/etc/default/etcd` и заменить строку `ETCD_INITIAL_CLUSTER_STATE="new"` на `ETCD_INITIAL_CLUSTER_STATE="existing"`.

7.5. Установка и настройка PostgreSQL

На каждом из узлов `orbox-psql1`, `orbox-psql2` и `orbox-psql3` выполнить установку PostgreSQL 15:

```
apt install -y --no-install-recommends ca-certificates lsb-release gnupg2 wget
curl
sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-
pgdg main" > /etc/apt/sources.list.d/pgdg.list'
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | apt-key
add -
apt update
apt install -y --no-install-recommends postgresql-15
```

Поскольку Patroni сам управляет запуском PostgreSQL, то нужно выключить его автозапуск:

```
systemctl stop postgresql
systemctl disable postgresql
```

7.6. Установка и настройка Patroni

Patroni устанавливается на тех же самых узлах, что и PostgreSQL orbox-psql1, orbox-psql2 и orbox-psql3:

```
apt install -y --no-install-recommends python3-pip python3-psycopg2
pip install patroni[etcd]
```

На всех узлах создать папку, в которой будут находиться конфигурационный файл Patroni и файлы базы данных PostgreSQL:

```
mkdir -p /etc/patroni/
mkdir -p /var/lib/postgresql/psql-ha/
chown -R postgres:postgres /var/lib/postgresql/
chmod 0700 /var/lib/postgresql/psql-ha/
```

На узле orbox-psql1 создать конфигурационный файл `/etc/patroni/patroni.yml` со следующим содержимым:

```
scope: orbox-ha-psql
name: orbox-psql1

restapi:
  listen: 0.0.0.0:8008
  connect_address: orbox-psql1:8008

etcd3:
  hosts:
    - orbox-etcd1:2379
    - orbox-etcd2:2379
    - orbox-etcd3:2379

bootstrap:
  dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 1048576
    postgresql:
      use_pg_rewind: true
      pg_hba:
        - host replication replicator 127.0.0.1/32 scram-sha-256
        - host replication replicator 0.0.0.0/0 scram-sha-256
        - host all all 0.0.0.0/0 scram-sha-256
```

```

        parameters:

initdb:
  - encoding: UTF8
  - data-checksums

users:
  orbox_user:
    password: orbox_user_password
    options:
      - createrole
      - createdb

postgresql:
  listen: 0.0.0.0:5432
  connect_address: orbox-psql1:5432

  data_dir: /var/lib/postgresql/psql-ha/
  bin_dir: /usr/lib/postgresql/15/bin/

  pgpass: /tmp/pgpass0
  authentication:
    replication:
      username: replicator
      password: replicator_password
    superuser:
      username: postgres
      password: postgres_password
  parameters:
    unix_socket_directories: '/var/run/postgresql/'

tags:
  nofailover: false
  noloadbalance: false
  clonefrom: false
  nosync: false

```

На узлах **orbox-psql2** и **orbox-psql3** также создать подобные конфигурационные файлы, но заменить все строки **orbox-psql1** на **orbox-psql2** и **orbox-psql3** для узлов **orbox-psql2** и **orbox-psql3** соответственно.

Также для повышения безопасности во всех трёх конфигурационных файлах нужно

внести следующие изменения:

- Изменить строки с паролями: `orbox_user_password`, `replicator_password`, `postgres_password`
- В строке **host replication replicator 0.0.0.0/0 scram-sha-256** заменить `0.0.0.0/0` на реальную подсеть, в которой находятся узлы кластера

`orbox_user` и `orbox_user_password` являются логином и паролем, с помощью которых Сервер Контроля будет подключаться к базе данных.

На всех узлах создать systemd unit `/etc/systemd/system/patroni.service` со следующим содержимым:

```
[Unit]
Description=Runners to orchestrate a high-availability PostgreSQL
After=syslog.target network.target

[Service]
Type=simple

User=postgres
Group=postgres

ExecStart=/usr/local/bin/patroni /etc/patroni/patroni.yml
ExecReload=/bin/kill -s HUP $MAINPID
KillMode=process
TimeoutSec=30
Restart=no

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

Запуск Patroni (выполнить на всех узлах с Patroni):

```
systemctl daemon-reload
systemctl enable patroni
systemctl start patroni
```

Для проверки состояния кластера Patroni выполнить на любом узле с Patroni команду:

```
patronictl -c /etc/patroni/patroni.yml list
```

Вывод команды должен быть примерно следующим:

```
+ Cluster: orbox-ha-psql ---+-----+-----+-----+
| Member      | Host          | Role    | State    | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| orbox-psql1 | orbox-psql1  | Leader  | running  | 1  |           |
```

orbox-psql2	orbox-psql2	Replica	streaming	1		0	
orbox-psql3	orbox-psql3	Replica	streaming	1		0	
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

7.7. Установка и настройка HAProxy

HAProxy устанавливается на тех же самых узлах, что и PostgreSQL orbox-psql1, orbox-psql2 и orbox-psql3:

```
apt install -y --no-install-recommends haproxy
```

На узлах orbox-psql1, orbox-psql2 и orbox-psql3 создать конфигурационный файл **/etc/haproxy/haproxy.cfg** со следующим содержимым:

```
global
    maxconn 100
defaults
    log global
    mode tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
    timeout check 5s
listen stats
    mode http
    bind *:7000
    stats enable
    stats uri /
listen postgres
    bind *:5000
    option httpchk
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server orbox-psql1 orbox-psql1:5432 maxconn 100 check port 8008
    server orbox-psql2 orbox-psql2:5432 maxconn 100 check port 8008
    server orbox-psql3 orbox-psql3:5432 maxconn 100 check port 8008
```

Конфигурационный файл на всех узлах одинаковый. Такая конфигурация будет принимать соединения к базе данных на порту 5000 и проксировать их на узел PostgreSQL, который в настоящий момент является лидером.

Запуск HAProxy (выполнить на всех узлах с HAProxy):

```
systemctl enable haproxy
systemctl restart haproxy
```

Проверить состояние HAProxy можно через браузер на страницах <http://orbox-psql1:7000>, <http://orbox-psql2:7000> или <http://orbox-psql3:7000>.

7.8. Настройка сервера контроля ORBOX

В конфигурационном файле `appsettings.json` указать логин, пароль (берутся из конфигурации Patroni), в качестве порта базы данных указать порт прокси сервера HAProxy, а также включить репликацию базы данных и перечислить все узлы HAProxy:

```
"DatabaseConnectionString": "Host=orbox-psql1;Port=5000;Database=orbox;SearchPath=public;User Id=orbox_user;Password=orbox_user_password;CommandTimeout=0;EntityAdminDatabase=postgres;Include Error Detail=true;Log Parameters=true",

"Replication": {
  "DatabaseReplicationEnabled": true,
  "DatabaseNodes": [
    "orbox-psql1:5000",
    "orbox-psql2:5000",
    "orbox-psql3:5000"
  ]
}
```