



ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА

Программное обеспечение «Радиобокс»

Версия 1.0

Санкт-Петербург
2026

1. Общая информация

- 1.1. ПО "Радиобокс" представляет собой набор микросервисов, общей задачей которого является получение исходного аудиопотока, и преобразование этого потока в выходные потоки необходимых форматов для распространения на широкую аудиторию.
- 1.2. Компоненты ПО рассчитаны на запуск и работу внутри кластера kubernetes.
- 1.3. Экземпляр ПО "Радиобокс" предоставляется для проведения экспертной проверки в виде предварительно настроенной виртуальной машины.
 - 1.3.1. Формат предоставления - файл виртуальной машины VirtualBox с расширением .ova.
 - 1.3.2. Сборка осуществлялась с использованием VirtualBox 7.0 на ОС Debian 12.
 - 1.3.3. Предоставляемый образ содержит все необходимые для запуска и демонстрации экземпляра компоненты (см. раздел с описанием содержимого).
- 1.4. Для удобства проверки виртуальную машину было решено предоставить в двух вариантах:
 - 1.4.1. Образ для скачивания и установки на собственное оборудование.
 - 1.4.1.1. Доступна для скачивания по ссылке <http://78.29.55.9:17987/Radiobox.ova>
 - 1.4.1.2. Размер образа в формате .ova составляет примерно 10 Гб, скачивание может занять некоторое время.
 - 1.4.1.3. Машина для установки должна иметь достаточные ресурсы (см. раздел по установке образа).
 - 1.4.2. Запущенная виртуальная машина с доступом по сети, на основе того же самого образа, чтобы можно было пропустить процесс установки и сразу начать проверку.
 - 1.4.2.1. Доступна по ssh по ip-адресу 78.29.55.9:17022. (данные для доступа см. в разделе 4 Доступ к консоли VM).
 - 1.4.2.2. Не требует предварительной подготовки и установки, все порты, описанные в п. 3.8 уже проброшены.
- 1.5. Оба представленных варианта изнутри абсолютно одинаковы, их содержимое ничем не отличается.
- 1.6. В предоставленном варианте для удешевления демонстрации использован ряд архитектурных упрощений относительно производственной среды выполнения, которые в общем виде не влияют на работоспособность и функциональность ПО, но могут снизить его производительность. Как пример, базы данных в демо находятся в виде отдельных контейнеров в контуре кластера, а в производственной среде они размещаются на отдельных выделенных серверах. Такие упрощения нисколько не меняют сути работы, выполняемой ПО Радиобокс, и все его ключевые компоненты (контроллеры, кодировщики, чанкователи и серверы вещания) предоставляются в неизменном виде, в каком и выполняется их запуск и работа внутри производственного кластера.

2. Описание содержимого виртуальной машины

- 2.1. Виртуальная машина содержит внутри следующие компоненты:

- 2.1.1. Кластер `minikube` с развёрнутыми внутри него компонентами ПО и предзагруженными образами контейнеров отдельных компонентов. Он эмулирует производственную среду, в которой работает ПО.
- 2.1.2. Набор необходимых для тестирования системных утилит (`ffmpeg`, `curl`).
- 2.1.3. Набор манифестов, на основе которых выполнено конфигурирование кластера. Находится в папке `/home/user/radiobox`.
- 2.1.4. Скрипты для запуска и тестирования ПО:
 - 2.1.4.1. `/home/user/radiobox/up.sh` применяет все манифесты.
 - 2.1.4.2. `/home/user/radiobox/port-forward.sh` пробросы портов кластера на порты виртуальной машины.
 - 2.1.4.3. `/home/user/radiobox/example.sh` примеры команд для работы с ПО.
- 2.2. В демонстрационном экземпляре ПО уже настроено вещание двух входящих потоков, каждого в двух разных исходящих вариантах, один из них включен, а другой выключен.

3. Установка образа виртуальной машины

- 3.1. Хост-машина для установки должна иметь не менее 8Гб RAM и 8 ядер процессора, а также не менее 25 Гб свободного места.
- 3.2. На хост-машине должен быть установлен гипервизор Oracle VirtualBox.
- 3.3. Установка осуществляется путём импорта конфигурации виртуальной машины из предоставленного файла через интерфейс гипервизора.
- 3.4. После установки виртуальная машина запускается обычным способом.
- 3.5. Для работы требуется подключение к сети Интернет, чтобы ПО могло забирать входящие потоки, указанные в примерах. При необходимости можно заменить их на потоки, доступные в локальной сети хост-машины, и работать только внутри локальной сети. Кроме получения входящего звукового потока, каких-либо инициативных запросов во внешнюю сеть ПО не осуществляет.
- 3.6. Система внутри образа настроена на автоматическое получение сетевого адреса у маршрутизатора по DHCP, но `bridge`-интерфейс виртуальной машины по умолчанию выключен. Если предполагается выполнять проверку снаружи виртуальной машины (например, нужно послушать поток плеером), то необходимо поднять интерфейс и получить адрес, запустив команду `sudo ip link set enp0s8 up && sudo dhclient enp0s8` и затем посмотреть IP-адрес через `/usr/sbin/ifconfig`
Либо можно воспользоваться командой `/home/user/network-up.sh`, которая содержит те же самые команды (поднимет интерфейс и сразу покажет IP-адрес). После этого можно подключиться по SSH на выданный IP-адрес любым удобным терминалом.
- 3.7. После запуска кластер `minikube` внутри виртуальной машины остановлен. Чтобы запустить кластер нужно выполнить `minikube start`
После успешного старта ПО готово к использованию.
- 3.8. Для удобства использования можно сразу пробросить необходимые порты сервисов. Например, чтобы получить доступ к дашборду кластера на порту 19090, можно выполнить в консоли виртуальной машины

```
minikube kubectl -- port-forward -n kubernetes-dashboard svc/kubernetes-dashboard 19090:80 --address 0.0.0.0
```

После этого можно смотреть дашборд браузером по адресу *http://<IP-адрес ВМ>:19090*. Посмотреть текущие порты или запустить все необходимые пробросы сразу можно с помощью файла *./home/user/radiobox/port-forward.sh*

После запуска команды проброса портов будут доступны следующие порты:

- *19090* - k8s dashboard, просмотр содержимого кластера
- *13306* - база данных mariadb
- *18000* - выходной icecast
- *10080* - выходной hls
- *13000* - API сервиса

Важно: для того, чтобы пробросы из файла заработали, нужно сначала выполнить хотя бы один раз *minikube kubectl -- port-forward* чтобы пробросить какой-то один порт отдельно, а потом уже запускать файл. Если запустить файл сразу, то minikube уйдёт в загрузку зависимостей, и пробросы не запустит.

4. Доступ к консоли виртуальной машины

4.1. Адрес для доступа зависит от выбранного способа доступа:

4.1.1. Если используется доступ к запущенной виртуальной машине, то необходимо использовать ip *78.29.55.9* и порт *17022 ssh -p 17022 user@78.29.55.9*

4.1.2. Если был скачан и развёрнут представленный образ виртуальной машины, то необходимо определить сетевой адрес виртуальной машины в локальной сети (см. п. 3.6) и использовать его.

```
ssh user@<IP-адрес ВМ>
```

4.2. Доступ осуществляется с помощью имени пользователя и пароля.

4.3. Имя пользователя - *user*, пароль - *12345687*.

4.4. Пользователь имеет привилегии *sudo*.

4.5. Для доступа под root напрямую в консоль можно использовать команду *sudo su*.

5. Доступ к веб-интерфейсу кластера

5.1. Внутри кластера запущен процесс *kubernetes-dashboard*, позволяющий удобно просматривать содержимое.

5.2. При использовании доступа к запущенной виртуальной машине можно смотреть веб-интерфейс по адресу *http://78.29.55.9:19090*, логин и пароль не требуются

5.3. При развёртывании образа виртуальной машины необходимо выполнить проброс порта, как описано в пункте 3.8, после чего интерфейс будет доступен по адресу *http://<IP-адрес ВМ>:19090*.

5.4. Данный веб-интерфейс не является частью предоставленного ПО, и используется лишь для удобства демонстрации. В производственном режиме такой интерфейс не используется.

6. Доступ к кластеру из консоли

6.1. Для просмотра и изменения параметров кластера можно использовать консольную утилиту *minikube*. Информацию по её использованию можно получить при помощи команды *minikube --help*.

6.2. Для работы с контейнерами внутри кластера можно использовать стандартную утилиту *kubectl* внутри *minikube*. Информацию по её использованию можно получить при помощи команды *minikube kubectl -- --help*.

6.3. Пароли отдельных сервисов (например БД) можно посмотреть в соответствующих манифестах в папке */home/user/radiobox*.

7. Схема тестирования

7.1. В общем виде тестирование заключается в отправке команд в API для настройки потоков и последующей проверке наличия ожидаемых выходных потоков на соответствующих точках монтирования.

7.2. Система содержит два преднастроенных аккаунта, в каждом из которых добавлено по одному входящему потоку и по два исходящих потока.

7.2.1. Аккаунт 1

Пользователь *acme-user*

Токен доступа *acmetoken000000000000000000000002*

В аккаунте подключен один входящий поток, из него формируются два выходных, все потоки активны.

7.2.2. Аккаунт 2

Пользователь *beta-user*

Токен доступа *betatoken000000000000000000000003*

В аккаунте подключен один входящий поток, из него формируются два выходных. Входящий поток отключен, соответственно выходящие неактивны.

7.3. В рамках проведения теста можно включать, выключать или удалять имеющиеся потоки, и создавать новые, которые потом также можно включать, выключать или удалять.

7.4. Для того, чтобы потоки были доступны снаружи виртуальной машины необходимо выполнить пробросы портов от сервисов *icecast-service* (порт *8000*) и *hls-streaming-server* (порт *80*). Примеры выполнения таких пробросов можно найти в файле */home/user/radiobox/port-forward.sh* (см. п 3.8).

7.5. Для удобства тестирования в файле */home/user/radiobox/example.sh* были собраны типовые команды, которые могут быть использованы для проверки работы сервиса.

8. Контакты

8.1. По всем вопросам, связанным с настройкой тестового экземпляра можно писать ответственному специалисту на почту *mi@micro-it.ru* (Марат Исмагилов).