

# WCS в Docker

- [Пример развертывания в Docker](#)
- [Настройка docker контейнера](#)
  - [Настройка сети](#)
    - [Особенности динамического определения IP адреса](#)
  - [Настройка docker томов](#)
  - [Дополнительная настройка](#)
  - [Порты контейнера](#)
- [Известные проблемы](#)

Начиная со сборки 5.2.458, WCS доступен в виде официального [Docker образа](#)

## Пример развертывания в Docker

Развернем WCS в Docker за несколько шагов:

1. Загрузите актуальную сборку с Docker Hub

```
sudo docker pull flashphoner/webcallserver
```

2. Настройте docker сеть

```
sudo docker network create \  
  --subnet 192.168.1.1/24 \  
  --gateway=192.168.1.1 \  
  --driver=bridge \  
  --opt com.docker.network.bridge.name=br-testnet testnet
```

3. Запустите docker контейнер, указав номер [ознакомительной](#) или [коммерческой](#) лицензии

```
docker run \  
  -e PASSWORD=password \  
  -e LICENSE=license_number \  
  -e LOCAL_IP=192.168.1.10 \  
  --net testnet --ip 192.168.1.10 \  
  --name wcs-docker-test --rm -d flashphoner/webcallserver:latest
```

4. Проверьте доступность контейнера по SSH

```
ssh root@192.168.1.10
```

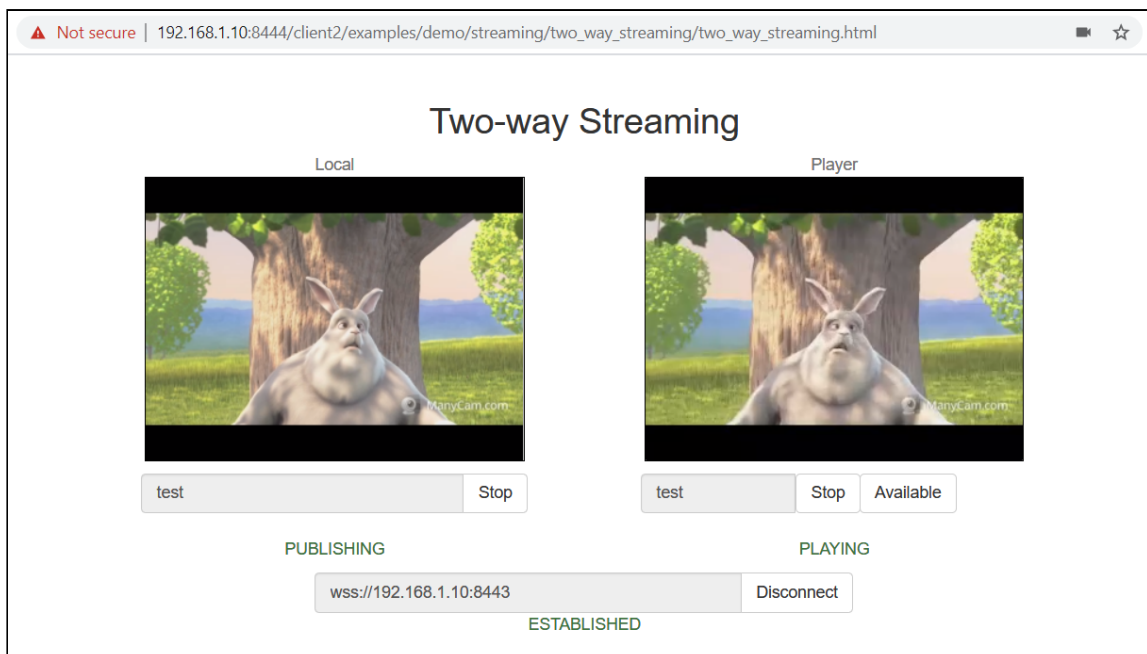
```
Warning: Permanently added '192.168.1.10' (ECDSA) to the list of known hosts.
root@192.168.1.10's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Fri Jan 10 04:19:52 2020 from 192.168.1.1
root@d24386fcea30:~#
```

5. Откройте страницу примера Two Way Streaming в локальном браузере и опубликуйте поток



6. Закройте страницу, остановите контейнер

```
sudo docker stop wcs-docker-test
```

При остановке контейнера лицензия будет деактивирована, чтобы избежать излишнего биллинга.

## Настройка docker контейнера

## Настройка сети

По умолчанию, IP адрес назначается контейнеру динамически. Если планируется использовать статический IP адрес, нужно указать его в параметрах команды `docker run` и передать в контейнер при помощи переменной `LOCAL_IP`:

```
docker run -e LICENSE=license_number -e PASSWORD=secret \
-e LOCAL_IP=static_ip \
--net your-docker-network \
--ip static_ip \
--name webcallserver-instance -d webcallserver:latest
```

Если docker узел находится в локально сети за NAT, и сервер должен быть доступен снаружи, необходимо указать внешний IP адрес в переменной `EXTERNAL_IP`:

```
docker run -e LICENSE=license_number -e PASSWORD=secret \
-e LOCAL_IP=static_ip \
-e EXTERNAL_IP=external_ip \
--net your-docker-network \
--ip static_ip \
--name webcallserver-instance -d webcallserver:latest
```

## Особенности динамического определения IP адреса

Если IP адрес назначается контейнеру динамически, и при этом контейнер имеет несколько сетевых интерфейсов (такое может быть, например, в случае использования интерфейсов docker хоста), на этапе запуска контейнера необходимо вручную выбрать IP адрес, к которому привяжется WCS сервер. В этом случае контейнер должен быть запущен в интерактивном режиме

```
docker run \
-e PASSWORD=password \
-e LICENSE=license_number \
--net host \
--name wcs-docker-test --rm -it flashphoner/webcallserver:latest
```

Скрипт запуска контейнера определит IP адреса всех интерфейсов и предложит ввести тот, который будет прослушивать WCS сервер

```
192.168.0.195 192.168.1.1 172.17.0.1 172.16.0.158
We have found 4 IPs, what should be used for WCS: 192.168.0.195 192.168.1.1 172.17.0.1 172.16.0.158 ? 172.17.0.1
```

В параметр `ip_local` будет записан выбранный IP адрес, например

```
ip_local=172.17.0.1
```

Настройка параметров контейнера

При запуске в контейнер можно передавать следующие параметры:

- **PASSWORD** - пароль на доступ внутрь контейнера по SSH. Если эта переменная не определена, контейнер не будет доступен по паролю
- **LICENSE** - номер лицензии WCS. Если эта переменная не определена, лицензия может быть активирована через веб-интерфейс
- **LOCAL\_IP** - локальный IP адрес контейнера, который будет записан в параметр **ip\_local** в файле настроек **flashphoner.properties**. Если эта переменная не определена, IP адрес контейнера будет определен автоматически
- **EXTERNAL\_IP** - локальный IP адрес контейнера, который будет записан в параметр **ip** в файле настроек **flashphoner.properties**. Если эта переменная не определена и **LOCAL\_IP** не определена, внешний IP адрес контейнера будет определен автоматически
- **CDN\_ROLE** - роль контейнера в **CDN**. Если эта переменная не определена, контейнер будет работать как отдельно стоящий сервер
- **CDN\_ENTRY\_POINT** - адрес сервера-точки входа в CDN для данного контейнера. Если эта переменная не определена, точка входа не задается
- **CDN\_GROUPS** - **группы CDN**, в которые должен входить контейнер. Если эта переменная не определена, контейнер не принадлежит ни к одной группе

## Настройка docker томов

При запуске в контейнер могут быть смонтированы следующие тома:

- **/conf** - каталог дополнительных файлов конфигурации, путь отсчитывается от корня контейнера. Этот каталог может быть смонтирован только для чтения
- **/usr/local/FlashphonerWebCallServer/hls** - каталог для сегментов HLS. Этот каталог должен быть смонтирован для записи
- **/usr/local/FlashphonerWebCallServer/logs** - каталог для логирования. Этот каталог должен быть смонтирован для записи
- **/usr/local/FlashphonerWebCallServer/media** - каталог с файлами для воспроизведения VOD. Этот каталог может быть смонтирован только для чтения. Этот каталог может быть общим с другими экземплярами WCS
- **/usr/local/FlashphonerWebCallServer/records** - каталог для файлов записей потоков. Этот каталог должен быть смонтирован для записи. Этот каталог может быть общим с другими экземплярами WCS

Пример запуска контейнера со всеми смонтированными томами:

```
docker run -e LICENSE=license_number -e CDN_ROLE=origin \
-v /opt/wcs/conf:/conf \
-v /opt/wcs/logs:/usr/local/FlashphonerWebCallServer/logs \
```

```
-v /opt/wcs/hls:/usr/local/FlashphonerWebCallServer/hls \
-v /opt/wcs/media:/usr/local/FlashphonerWebCallServer/media \
-v /opt/wcs/records:/usr/local/FlashphonerWebCallServer/records \
--name webcallserver-instance -d webcallserver:latest
```

Не допускается монтирование всего каталога  
/usr/local/FlashphonerWebCallServer/conf как тома, или отдельных файлов из этого каталога!

Например, так делать нельзя:

```
docker run -e LICENSE=license_number -e PASSWORD=secret \
-v /opt/wcs/conf:/usr/local/FlashphonerWebCallServer/conf \
--name webcallserver-instance -d webcallserver:latest
```

и так тоже:

```
docker run -e LICENSE=license_number -e PASSWORD=secret \
-v
/opt/wcs/conf/flashphoner.properties:/usr/local/FlashphonerWebCallServer/conf/fl
--name webcallserver-instance -d webcallserver:latest
```

## Дополнительная настройка

В каталог, смонтированный как том `/conf`, могут быть помещены следующие файлы настроек

- `id_rsa.pub` - публичный SSH ключ для доступа к контейнеру. Если этот файл найден при запуске контейнера, будет работать SSH доступ в контейнер по соответствующему приватному ключу. В противном случае, если переменная `PASSWORD` определена, контейнер будет доступен по заданному паролю. Если не определены ни публичный ключ, ни пароль, SSH сервис не будет запущен
- `flashphoner.properties` - основной файл настроек сервера. Может быть использован для настройки экземпляра WCS
- `wcs-core.properties` - файл настроек Java. Может использоваться для [настройки памяти](#) и [подключения из JMC](#)
- `log4j.properties` - файл настроек [логирования](#)
- `wss.jks` - файл хранилища ключей SSL сертификатов. Файл может быть подготовлен при помощи [keytool](#)
- `cdn_profiles.yml` - файл настройки [профилей](#) транскодирования в CDN
- `database.yml` - файл настройки [пользователей](#) и [приложений REST hook](#)
- `*.sdp` - файлы [настроек SDP](#)

Все эти файлы могут быть использованы, например, для восстановления настроек экземпляра WCS из сохраненной копии и для автоматического развертывания.

Файлы копируются в контейнер 'как есть', за исключением `flashphoner.properties`:

- параметры `ip` и `ip_local` будут изменены (см описание переменной `IP_LOCAL`)
- если переменная `CDN_ROLE` определена, но файл не содержит никаких настроек CDN, будут добавлены [настройки по умолчанию](#)

## Порты контейнера

По умолчанию доступны следующие порты:

- `22/tcp` - SSH (этот порт активен, если настроен)
- `554/tcp` - RTSP порт
- `1935/tcp` - RTMP порт
- `1935/udp` - RTMFP порт
- `2001/tcp` - CLI порт
- `7777/tcp` - порт мониторинга сервера
- `8080-8084/tcp` - WS, HTTP, CDN порты
- `8443-8445/tcp` - WSS, HTTPS порты
- `8888/tcp` - HTTPS порт (для совместимости со старыми версиями)
- `9091/tcp` - HTTP порт (для совместимости со старыми версиями)
- `30000-35000/tcp` - TCP медиа порты
- `30000-35000/udp` - UDP медиа порты
- `50999/tcp` - JMX порт

Состав портов при необходимости может быть изменен при помощи параметров запуска `docker run --expose`. В этом случае значения портов должны быть указаны в файлах настроек `flashphoner.properties` и `wcs-core.properties`, см выше

## Известные проблемы

1. При запуске контейнера нельзя изменить ограничение на количество открытых файлов изнутри контейнера.

Симптомы: при запуске контейнера в интерактивном режиме выводится сообщение

```
ulimit: open files: cannot modify limit: Operation not permitted
```

Решение: установить ограничение на количество открытых файлов в параметрах запуска контейнера

```
docker run --ulimit nofile=100000:100000 \  
-e LICENSE=license_number -e PASSWORD=secret \  
--name webcallserver-instance --rm webcallserver:latest
```

2. Не рекомендуется использовать Docker на Windows из-за проблем с пробросом портов

Симптомы: при развертывании образа в Docker под Windows контейнер запущен, но недоступен по SSH, по HTTP/HTTPS, не устанавливается WebRTC соединение

Решение: использовать Docker на Linux, либо [запускать WCS в WSL 2](#)

## Attachments:

- [image2020-1-10\\_11-30-30.png](#) (image/png)
- [image2020-1-10\\_11-24-5.png](#) (image/png)
- [image2020-3-17\\_14-18-3.png](#) (image/png)
- [image2020-3-17\\_14-29-24.png](#) (image/png)