

Запуск и остановка

- Основные команды и контроль запуска
- Способы запуска
 - Запуск как сервиса
 - Автоматический перезапуск сервиса при сбоях
 - Запуск из командной строки
- Настройка переменных окружения
- Запуск с выводом в stdout
 - Запуск с выводом лога JVM в файл
- Запуск с различными правами пользователя
 - Запуск в сборках 5.2.864-5.2.972
 - Запуск в сборках 5.2.976 и новее
 - Изменение режима запуска
 - Настройка прав доступа к каталогам при запуске от пользователя flashphoner
- Параметры JVM
 - Автоматическое определение версии Java и корректировка параметров запуска
- Проверка работоспособности сервера при запуске

Основные команды и контроль запуска

После [активации лицензии](#) запустите WCS командой

```
sudo systemctl start webcallserver
```

Остановка сервера выполняется командой

```
sudo systemctl stop webcallserver
```

Есть несколько способов убедиться в том, что сервер стартовал и готов к работе:

1. Убедитесь, что серверный процесс работает.

```
pgrep -afn com.flashphoner.server.Server
```

В консоли должен отобразиться процесс WCS Core (PID 6880 на примере ниже):

```
[root@localhost ~]# pgrep -afn com.flashphoner.server.Server
6880 java -Xmx4g -Xms4g -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:
CMSInitiatingOccupancyFraction=70 -Djava.net.preferIPv4Stack=true -Dcom.sun.management.jmxremote=true -Dcom.sun.
management.jmxremote.local.only=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.
authenticate=false -Dcom.sun.management.jmxremote.port=50999 -Djava.rmi.server.hostname=p13.flashphoner.com -XX:
ErrorFile=/usr/local/FlashphonerWebCallServer/logs/error%p.log -XX:+PrintGCDateStamps -XX:+PrintGCDetails -
Xloggc:/usr/local/FlashphonerWebCallServer/logs/gc-core-2021-06-25_14-44.log -XX:+ExplicitGCInvokesConcurrent -
Dsun.rmi.dgc.client.gcInterval=36000000000 -Dsun.rmi.dgc.server.gcInterval=36000000000 -Dcom.flashphoner.fms.
AppHome=/usr/local/FlashphonerWebCallServer -Djava.library.path=/usr/local/FlashphonerWebCallServer/lib/so:/usr
/local/FlashphonerWebCallServer/lib -DWCS_NON_ROOT=true -DsessionDebugEnabled=false -Djdk.tls.client.protocols="
TLSv1,TLSv1.1,TLSv1.2" -cp /usr/local/FlashphonerWebCallServer/lib/* com.flashphoner.server.Server
[root@localhost ~]#
```

2. Убедитесь, что серверный процесс слушает основные порты.

```
netstat -nlp | grep java
```

```
[root@localhost ~]# netstat -nlp | grep java
tcp        0      0 0.0.0.0:1098          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:1935          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:8080          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 127.0.0.1:2001        0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:8081          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 127.0.0.1:2002        0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:8082          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 192.168.1.5:3478      0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:50999         0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:8888          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:8443          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:8444          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:8445          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:9091          0.0.0.0:*             LISTEN      6880/java
tcp        0      0 0.0.0.0:45731         0.0.0.0:*             LISTEN      6880/java
udp        0      0 0.0.0.0:1935          0.0.0.0:*             6880/java
```

Если вы использовали стандартный набор портов, в списке портов, должны появиться порты 8080, 8444 (Websockets), 1935 (RTMP) или другие порты, которые вы сконфигурировали для WCS сервера.

3. Убедитесь, что WCS сервер пишет основной серверный лог

```
tail -f /usr/local/FlashphonerWebCallServer/logs/server_logs/flashphoner.log
```

В логе должна появиться информация о настройках, с которыми стартовал сервер.

Пример:

```
18:29:51,945 INFO      SettingsLoader - main Override setting allow_outside_codecs: from true to false
18:29:51,974 INFO      SettingsLoader - main Override setting codecs: from null to opus,alaw,ulaw,g729,
speex16,g722,mpg4-generic,telephone-event,h264,vp8,flv,mpv
18:29:51,975 INFO      SettingsLoader - main Override setting media_port_from: from 31001 to 31001
18:29:51,978 INFO      SettingsLoader - main Override setting keep_alive.enabled: from websocket,rtmp,rtmfp
to websocket,rtmfp
18:29:51,978 INFO      SettingsLoader - main Override setting webrtc_cc_min_bitrate: from 30000 to 3000000
18:29:51,979 INFO      SettingsLoader - main Override setting codecs_exclude_sip: from null to mpg4-generic,
flv,mpv,opus,ulaw,h264,g722,g729
18:29:51,979 INFO      SettingsLoader - main Override setting wss.port: from 8443 to 8443
18:29:51,980 INFO      SettingsLoader - main Override setting codecs_exclude_sip_rtmp: from null to opus,g729,
g722,mpg4-generic,vp8,mpv
18:29:51,980 INFO      SettingsLoader - main Override setting codecs_exclude_streaming: from null to
telephone-event
18:29:51,980 INFO      SettingsLoader - main Override setting webrtc_cc_max_bitrate: from 10000000 to 7000000
18:29:51,980 INFO      SettingsLoader - main Override setting ip: from 0.0.0.0 to 192.168.1.5
18:29:51,980 INFO      SettingsLoader - main Override setting client_log_level: from INFO to DEBUG
18:29:51,980 INFO      SettingsLoader - main Override setting ip_local: from 0.0.0.0 to 192.168.1.5
18:29:51,980 INFO      SettingsLoader - main Override setting media_port_to: from 32000 to 32000
18:29:51,981 INFO      SettingsLoader - main Override setting ws.port: from 8080 to 8080
```

Логи должны реагировать на соединения web-клиентов. Если [при тестировании](#) этого не происходит, убедитесь что серверный процесс работает и web-клиент корректно настроен на соединение именно с этим сервером. См. раздел [Возможные неисправности и способы их устранения](#) для получения дополнительной информации.

Если серверный процесс запущен и в логах нет ошибок, это означает что WCS сервер готов к работе и можно начинать тестирование.

Способы запуска

Запуск как сервиса

Запуск WCS как сервиса осуществляется командой:

```
sudo systemctl start webcallserver
```

Это предпочтительный способ запуска. При этом сервис стартует от пользователя `root`, а основной процесс от пользователя `flashphoner` или от `root`, в зависимости от [настройки режима запуска](#).

В сборке [5.2.1537](#) тип сервиса изменен с `simple` на `forking`. Кроме того, права доступа к файлу, содержащему PID основного процесса WCS, приведены в соответствие с требованиями `systemd`. Сервис помечается как `active` (`running`), и `systemd`, при необходимости, может послать сигнал для остановки основного процесса непосредственно ему

```
● webcallserver.service - Flashphoner WebCallServer
   Loaded: loaded (/etc/systemd/system/webcallserver.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-01-20 12:17:35 +07; 2 days ago
     Process: 19202 ExecStart=/bin/bash webcallserver start (code=exited, status=0/SUCCESS)
    Main PID: 19832 (java)
   CGroup: /system.slice/webcallserver.service
           └─19817 sudo -u flashphoner env LD_LIBRARY_PATH=/usr/local/FlashphonerWebCallServer/lib/so java -Xmx1024M -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=70 -Djava.net.preferIPv4Stack=true -Dcom.flashphoner.fms.AppHome=/usr/local/FlashphonerWebCallServer -cp /usr/local/FlashphonerWebCallServer-5.2.1536/bin : flashphoner
           └─19832 java -Xmx1024M -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=70 -Djava.net.preferIPv4Stack=true -Dcom.flashphoner.fms.AppHome=/usr/local/FlashphonerWebCallServer -cp /usr/local/FlashphonerWebCallServer-5.2.1536/bin : flashphoner

Jan 20 12:17:31 p13.flashphoner.com sudo[19758]: root : (command continued) -Dcom.flashphoner.fms.AppHome=/usr/local/FlashphonerWebCallServer -cp /usr/local/FlashphonerWebCallServer-5.2.1536/bin : flashphoner
Jan 20 12:17:32 p13.flashphoner.com sudo[19815]: root : TTY=unknown ; PWD=/usr/local/FlashphonerWebCallServer-5.2.1536/bin ; USER=flashphoner ; COMMAND=/bin/echo
Jan 20 12:17:32 p13.flashphoner.com sudo[19817]: root : TTY=unknown ; PWD=/usr/local/FlashphonerWebCallServer-5.2.1536/bin ; USER=flashphoner ; COMMAND=/bin/env LD_LIBRARY_PATH=/usr/local/FlashphonerWebCallServer/lib/so java -Xmx1024M -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=70 -Djava.net.preferIPv4Stack=true -Dcom.flashphoner.fms.AppHome=/usr/local/FlashphonerWebCallServer -cp /usr/local/FlashphonerWebCallServer-5.2.1536/bin : flashphoner
Jan 20 12:17:32 p13.flashphoner.com sudo[19817]: root : (command continued) -Dcom.flashphoner.fms.AppHome=/usr/local/FlashphonerWebCallServer -Djava.library.path=/usr/local/FlashphonerWebCallServer-5.2.1536/bin : flashphoner
Jan 20 12:17:35 p13.flashphoner.com sudo[19893]: root : TTY=unknown ; PWD=/usr/local/FlashphonerWebCallServer-5.2.1536/bin ; USER=flashphoner ; COMMAND=/bin/touch /usr/local/FlashphonerWebCallServer-5.2.1536/bin/.wcs.pid
Jan 20 12:17:35 p13.flashphoner.com sudo[19912]: root : TTY=unknown ; PWD=/usr/local/FlashphonerWebCallServer-5.2.1536/bin ; USER=flashphoner ; COMMAND=/bin/echo
Jan 20 12:17:35 p13.flashphoner.com sudo[19939]: root : TTY=unknown ; PWD=/usr/local/FlashphonerWebCallServer-5.2.1536/bin ; USER=flashphoner ; COMMAND=/bin/echo
Jan 20 12:17:35 p13.flashphoner.com sudo[19939]: root : [ OK ]
Jan 20 12:17:35 p13.flashphoner.com systemd[1]: webcallserver.service: Supervising process 19832 which is not our child. We'll most likely not notice when it exits.
Jan 20 12:17:35 p13.flashphoner.com systemd[1]: Started Flashphoner WebCallServer.
```

Автоматический перезапуск сервиса при сбоях

В сборке [5.2.1562](#) добавлен автоматический перезапуск сервиса `webcallserver.service` в тех случаях, когда сервис по каким-либо причинам перешел в статус `failed`. Предпринимается до 5 попыток перезапуска в случае, если между попытками проходит не более 2 минут.

Сервис по-прежнему может быть остановлен или запущен вручную. Если сервис остановлен вручную, попытки перезапуска предприниматься не будут.

Автоматический перезапуск может быть отключен командой

```
cd /usr/local/FlashphonerWebCallServer/bin
sudo ./webcallserver set-service-restart disable
```

Запуск из командной строки

При необходимости, запуск WCS может быть осуществлен командой:

```
cd /usr/local/FlashphonerWebCallServer/bin
sudo ./webcallserver start
```

Этот способ запуска полезен при отладке и тестировании.

Настройка переменных окружения

Подготовка запуска и переменных окружения происходит в скрипте `setenv.sh`. В этом скрипте могут быть заданы дополнительные параметры для WCS Core. Кроме того, здесь же задается параметр, предотвращающий утечки памяти на многопроцессорных системах:

```
MALLOC_ARENA_MAX=4
```

Запуск с выводом в stdout

В некоторых случаях, например когда сервер не стартует и не выдает никаких ошибок, требуется запуск с прямым выводом в консоль `stdout`. Прямой вывод в `stdout` не может использоваться в промышленной эксплуатации, т.к. при закрытии консоли или потере SSH-соединения сервер будет остановлен. Поэтому рекомендуется использовать прямой вывод в `stdout` только в целях отладки.

Для того чтобы запустить сервер в данном режиме, необходимо выполнить команду:

```
cd /usr/local/FlashphonerWebCallServer/bin
sudo ./webcallserver start standalone
```

Запуск с выводом лога JVM в файл

В сборке [5.2.1562](#) добавлена возможность запустить WCS с выводом лога JVM в файл, аналогично прямому выводу в stdout

```
cd /usr/local/FlashphonerWebCallServer/bin
sudo ./webcallserver start --java-log
```

Весь вывод Java машины в этом случае перенаправляется в файл `/usr/local/FlashphonerWebCallServer/logs/java.log`. Данную возможность следует использовать только в целях отладки, но не в промышленной эксплуатации, т.к. объем вывода в этот файл будет очень большим.

Запуск с различными правами пользователя

Запуск в сборках 5.2.864-5.2.972

Начиная со сборки [5.2.864](#), права, с которыми запускается WCS, подчиняются следующей логике:

1. Команда

```
sudo systemctl start webcallserver
```

запускает WCS всегда от пользователя flashphoner, если он есть в системе

2. Команда

```
./webcallserver start
```

запускает WCS от пользователя root, если она выполнена с правами root, например.

```
sudo ./webcallserver start
```

или от пользователя flashphoner, если запущена другим пользователем

Это касается и запуска с выводом на консоль

```
./webcallserver start standalone
```

Запуск в сборках 5.2.976 и новее

Начиная со сборки [5.2.976](#), права, с которыми запускается WCS, определяются только настройкой в файле `/usr/local/FlashphonerWebCallServer/bin/setenv.sh`:

При значении по умолчанию

```
WCS_NON_ROOT=true
```

WCS запускается от пользователя flashphoner

При

```
WCS_NON_ROOT=false
```

WCS запускается от пользователя root.

При этом сервис может стартовать от пользователя root, смена пользователя для запуска Java производится автоматически.

Изменение режима запуска

В сборке [5.2.1255](#) добавлена команда для изменения режимов запуска:

- переключение в режим запуска от пользователя root

```
sudo ./webcallserver set-root-mode enable
```

- переключение в режим запуска от пользователя flashphoner

```
sudo ./webcallserver set-root-mode disable
```

При этом WCS будет остановлен перед внесением изменений в настройки и автоматически запущен после внесения изменений

Настройка прав доступа к каталогам при запуске от пользователя flashphoner

В сборке [5.2.976](#) добавлена проверка возможности записи в каталоги сервера, а также в собственные каталоги, при запуске WCS от пользователя flashphoner. Если прав на запись не хватает, WCS не стартует с выводом сообщения на консоль и в лог `/usr/local/FlashphonerWebCallServer/logs/startup.log`

```
FlashphonerWebCallServer cannot be started from user flashphoner, please fix the permissions to the folders or run 'webcallserver set-permissions'!
```

В этом случае необходимо выполнить команду

```
sudo ./webcallserver set-permissions
```

Параметры JVM

Параметры задаются в файле [wcs-core.properties](#).

Дополнительные опции запуска могут быть также добавлены в файле `bin/setenv.sh` в следующие переменные:

`WCS_JAVA_OPTS`- список опций для WCS Core

При запуске параметры JVM проверяются на совместимость с текущей установленной версией Java. Если JVM не может стартовать с заданными параметрами, информация об этом выводится в файл `/usr/local/FlashphonerWebCallServer/logs/startup.log`, в соответствии с диагнозом, возвращенным Java при попытке запуска.

Автоматическое определение версии Java и корректировка параметров запуска

В сборке [5.2.972](#) добавлена возможность автоматического определения версии Java и корректировки [параметров запуска](#) при старте WCS, например, после обновления JDK. Параметры запуска JVM также могут быть скорректированы командой

```
cd /usr/local/FlashphonerWebCallServer/bin
sudo ./webcallserver set-java-opts
```

При этом меняются настройки в файле [wcs-core.properties](#), предыдущие настройки копируются в файл с расширением `.backup` и порядковым номером, например

```
[root@localhost ~]# ls -l /usr/local/FlashphonerWebCallServer/conf/wcs-core.properties.backup.*
-rw-r--r--. 1 flashphoner flashphoner 1614 Jun 23 10:15 /usr/local/FlashphonerWebCallServer/conf/wcs-core.properties.backup.0
-rw-r--r--. 1 flashphoner flashphoner 1543 Jun 23 10:17 /usr/local/FlashphonerWebCallServer/conf/wcs-core.properties.backup.1
```

Отметим, что выбранный сборщик мусора (GC) при этом не меняется, но могут быть изменены его настройки (например, наименование ключей запуска).

Проверка работоспособности сервера при запуске

После запуска сервера, скрипт webcallserver проверяет его работоспособность, ожидая ответа 200 OK на [предназначенный для этого запрос](#)

```
GET http://localhost:8081/health-check HTTP/1.1
```

В сборке [5.2.1084](#) добавлен ключ запуска, позволяющий указать максимальное количество попыток проверки работоспособности

```
sudo ./webcallserver start --health-timeout 10
```

По умолчанию, предпринимается 10 попыток с паузой в 1 секунду между ними. При каждой попытке скрипт ожидает ответа в течение секунды. Таким образом, максимальное время ожидания готовности сервера может составить по умолчанию до 20 секунд ($10 * (1+1)$).

Если процесс WCS не ответил на указанное число запросов, или ответ отличается от 200 OK, в лог запуска и на консоль будет выведено сообщение

```
FlashphonerWebCallServer started, but is not healthy, please try to restart
```

При необходимости, проверку можно отключить, указав нулевое количество попыток

```
sudo ./webcallserver start --health-timeout 0
```