

# WCS installation and activation

Installation archives of WCS are available for download [at this link](#).

You can download the installation archive directly to the server using `wget`, for example

```
wget
https://flashphoner.com/downloads/builds/WCS/5.2/FlashphonerWebCallServer-
5.2.XXXX.tar.gz
```

## Installation

To install WCS you need to unpack the archive and execute the installation script

`install.sh`:

```
tar -xvzf FlashphonerWebCallServer-5.2.XXXX.tar.gz
cd FlashphonerWebCallServer-5.2.XXXX
sudo ./install.sh
```

The installer will perform necessary checks and installs WCS to the following directory

`/usr/local/FlashphonerWebCallServer`

Installation includes the following steps:

- Reading and accepting of the license agreement
- Specifying IP addresses in case the script was unable to discover them automatically (see [IP address configuration](#))
- A request to add WCS to the Linux server autostart
- Java version checking and obsolete settings removing from `wcs-core.properties` file

As a result, a successful installation message will be shown. The following folders will be created in the `/usr/local/FlashphonerWebCallServer` directory:

- `bin` - WCS executables
- `client` - WCS JavaScript API 1.0, usage examples and stress-tests
- `client2` - WCS JavaScript API 2.0 and usage examples
- `conf` - configuration files
- `lib` - required libraries
- `tools` - tools to debug WCS

- `web` - WCS web interface static pages

## Silent mode installation

Sometimes, WCS can be installed in silent mode if automatic deployment system like Ansible is used to deploy a number of typical instances

```
./install.sh -silent
```

In this case:

- IP addresses detection does not work (`flashphoner.properties` file and other settings are supposed to be overwritten by automatic deployment system)
- `webcallserver.service` is configured to start from `root`

## Installation to a different Java versions

### JDK 12 or 14

The following settings should be removed from `wcs-core.properties` file

```
-XX:+UseConcMarkSweepGC  
-XX:+PrintGCDateStamps  
-XX:+PrintGCDetails
```

The following setting

```
-Xloggc:/usr/local/FlashphonerWebCallServer/logs/gc-core-
```

should be replaced by

```
-Xlog:gc*:/usr/local/FlashphonerWebCallServer/logs/gc-core-:time
```

### JDK 15

In addition to setting changes made for JDK 12 or 14, the following settings should be removed from `wcs-core.properties` file

```
-XX:+UseCMSInitiatingOccupancyOnly  
-XX:CMSInitiatingOccupancyFraction=70
```

### JDK 16

In addition to setting changes made for JDK 12, 14, 15, the following settings should be added to `wcs-core.properties` file

```
--illegal-access=permit
```

## JDK 17

In addition to setting changes made for JDK 12, 14, 15, 16, the following settings should be removed from `wcs-core.properties` file

```
--illegal-access=permit
```

and the following settings should be added

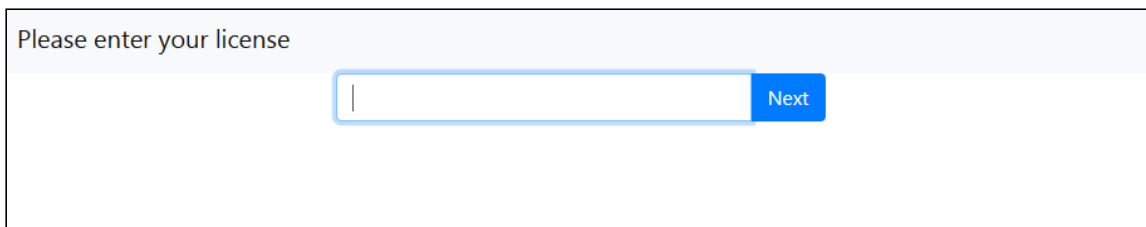
```
--add-exports java.base/sun.security.provider=ALL-UNNAMED  
--add-opens java.base/java.lang=ALL-UNNAMED
```

## Acquiring a license and activation

You can purchase a commercial license online here: <http://flashphoner.com/prices>. Also, a free 30-days trial version is available at this page: [http://flashphoner.com/get\\_trial/?product=wcs5](http://flashphoner.com/get_trial/?product=wcs5)

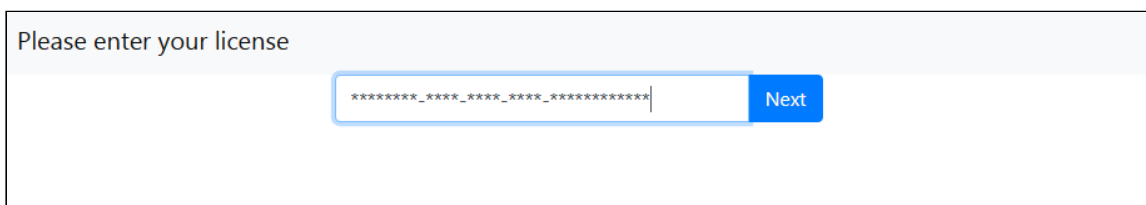
### Activation using web interface

On first logon to WCS web interface by address `https://yourserver:8444/admin/`, where `yourserver` is your server hostname with by default user `admin` (password `admin`), the request for license number is displayed:



Please enter your license

Enter your license number and press



Please enter your license

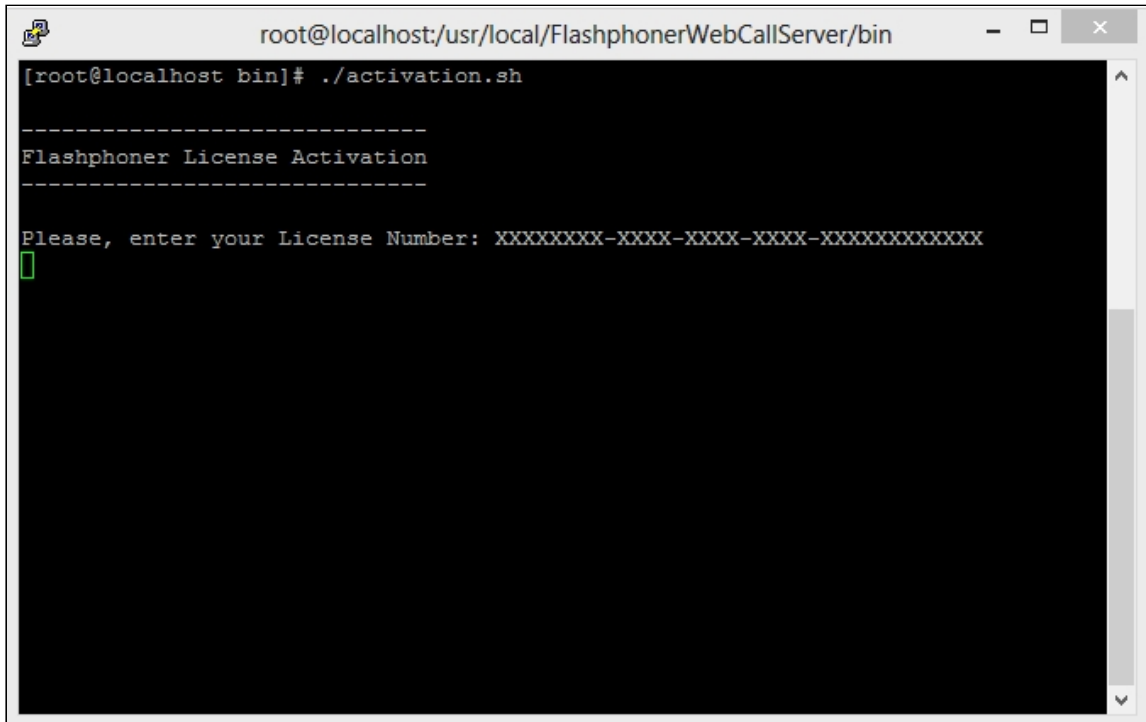
If activation was successful, log in page will be displayed.

### Activation using command line

Use this script to activate the license:

```
cd /usr/local/FlashphonerWebCallServer/bin
sudo ./activation.sh
```

The script asks you to enter the license number:

A terminal window titled 'root@localhost:usr/local/FlashphonerWebCallServer/bin'. The prompt is '[root@localhost bin]# ./activation.sh'. The output shows 'Flashphoner License Activation' followed by a separator line. Below that, it says 'Please, enter your License Number: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX' with a green cursor at the beginning of the line.

```
root@localhost:usr/local/FlashphonerWebCallServer/bin
[root@localhost bin]# ./activation.sh
-----
Flashphoner License Activation
-----
Please, enter your License Number: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
█
```

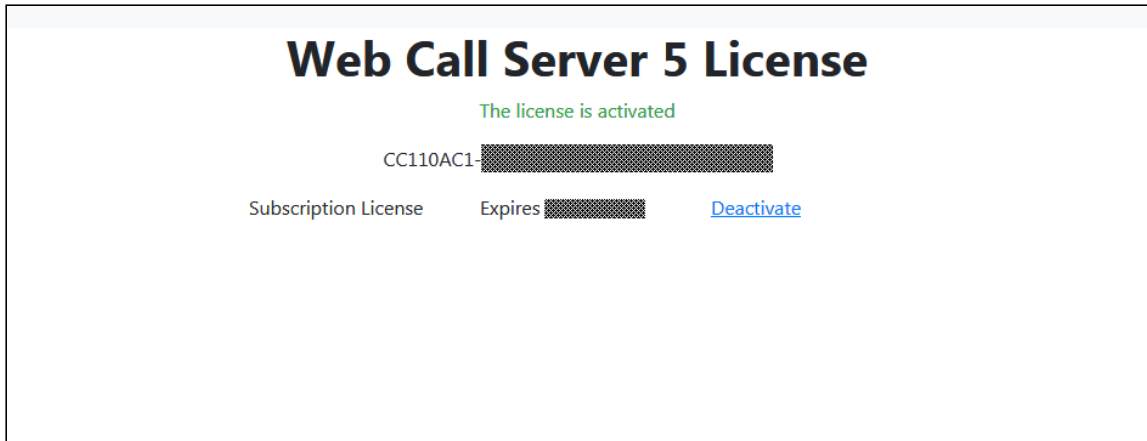
Enter the license number to finish the activation.

## Deactivating a license

To activate the same license on another server, for example, if you transfer WCS to another server, deactivate the license on the old server and activate it again on the new server.

### Deactivation using web interface

Log in to WCS web interface by address `https://yourserver:8444/admin/`, where `yourserver` is your server hostname. Press `License` button. The license key page will be displayed:



Click to **Deactivate** link. The license will be deactivated, and license key request page will be displayed:

Please enter your license

Next

## Deactivation using command line

Use this script for deactivation:

```
sudo ./deactivation.sh
```

## If activation fails

Check your firewall and make sure the WCS-сервер is able to establish an HTTPS / TCP connection to the activation server `my.flashphoner.com:443`. If the activation server is unavailable, configure HTTP proxy as described below.

## Configuring HTTP proxy if the activation server is not available

Flashphoner license servers are behind AWS ELB (Elastic Load Balancer) at `my.flashphoner.com`. In case the ELB (and, correspondingly, the license servers) become unavailable for some reason, you will need to configure HTTP Proxy to make subscriptions work and to activate licenses.

Proxy settings can be found in the [configuration file](#) named `/usr/local/FlashphonerWebCallServer/conf/wcs-core.properties`:

```
# Proxy configuration
-Dhttp.proxyHost=proxy1.flashphoner.com
-Dhttp.proxyPort=3128
-Dhttps.proxyHost=proxy1.flashphoner.com
-Dhttps.proxyPort=3128
-Dhttp.nonProxyHosts=localhost|127.0.0.1|10.*|169.254.169.254
```

You can specify the address of your proxy server or the address of one of two Flashphoner proxy servers:

```
proxy1.flashphoner.com
proxy2.flashphoner.com
```

### Warning

Proxy settings do not allow quote symbols! The following example will NOT work:

```
-Dhttp.proxyHost="proxy2.flashphoner.com"
-Dhttp.proxyPort="3128"
-Dhttps.proxyHost="proxy2.flashphoner.com"
-Dhttps.proxyPort="3128"
-Dhttp.nonProxyHosts="localhost|127.0.0.1|10.*|169.254.169.254"
```

## Configuring Firewall and NAT

For correct operation of the server, all ports listed in the table in the [Server core](#) section must be open for incoming traffic (UDP) or for establishing connection (TCP).

The table below lists example NAT rules for routing of incoming traffic in case the traffic comes to the external IP address and has to be routed to the server where WCS is installed. The table shows the minimum required number of ports necessary for operation of the server. Depending on the specific functions and modules (see the [Architecture](#) section), the set of rules may be extended in accordance with the table in the [Server core](#) section. The same ports must be opened in the firewall of the Linux server.

The table lists default ports. You can configure the WCS server to other other ports (see the [Settings](#) section).

Type	Protocol	Port Range	Source
Custom UDP Rule	UDP	30000-33000	0.0.0.0/0
Custom TCP rule	TCP	8081	0.0.0.0/0
Custom TCP Rule	TCP	8444	0.0.0.0/0
Custom TCP Rule	TCP	8443	0.0.0.0/0

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	30000-33000	0.0.0.0/0
Custom TCP Rule	TCP	1935	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0

SSH access via the standard port 22 is specified here only to administer the Linux server itself. Hence, the minimum required set of ports is:

- 1935 - RTMP
- 8081 - built-in WCS web server that works via HTTP
- 8443 - Secure Websocket
- 8444 - built-in WCS web server that works via HTTPS
- 30000-33000 - SIP, Media, RTSP

Description of SSL certificate configuration for Secure Websockets and HTTPS can be found in this section: [SSL certificates management](#).

## IP address configuration

Proper installation requires discovering of the external IP address of your server and the address of the network interface. The installation script attempts to detect the address automatically, and if this has failed for some reasons, the script asks you to enter addresses manually.

If you followed recommendation of the [Requirements and prerequisites](#) section and began with a server with a static and public IP address that matches the one of the network interface, the IP address of your server can be discovered with a simple command

```
ifconfig
```

or

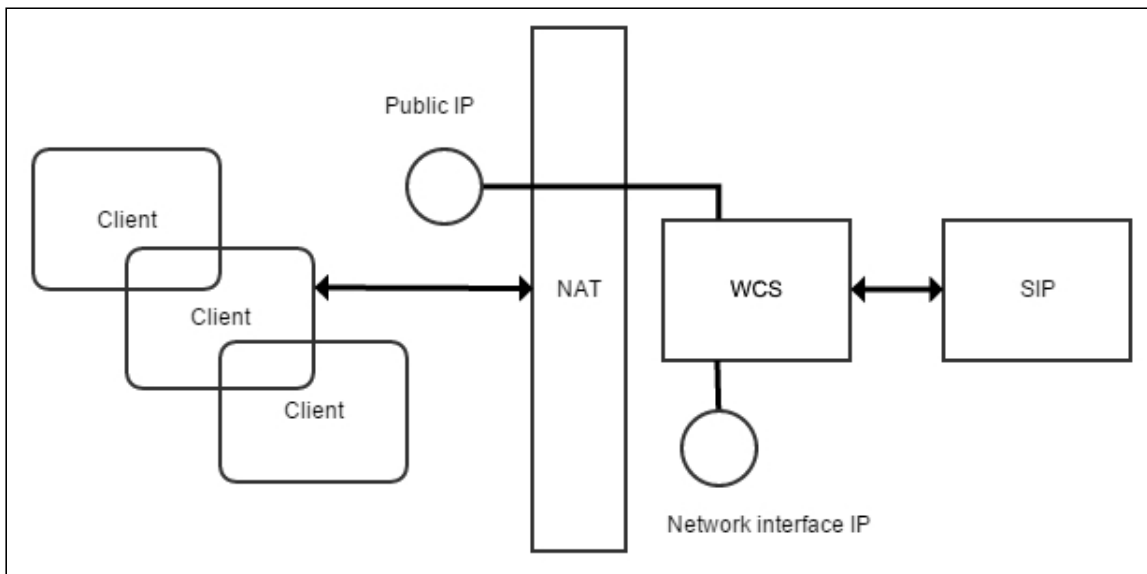
```
ip a
```

```
root@CentOS-64-64-minimal:~  
[root@CentOS-64-64-minimal ~]# ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:24:21:9C:37:ED  
          inet addr:188.40.69.75  Bcast:188.40.69.75  Mask:255.255.255.255  
          inet6 addr: 2a01:4f8:100:43a7::2/64  Scope:Global  
          inet6 addr: fe80::224:21ff:fe9c:37ed/64  Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:993166521  errors:0  dropped:0  overruns:0  frame:0  
          TX packets:914983127  errors:0  dropped:0  overruns:0  carrier:0  
          collisions:0  txqueuelen:1000  
          RX bytes:1172968732177 (1.0 TiB)  TX bytes:897517351983 (835.8 GiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128  Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:16397063  errors:0  dropped:0  overruns:0  frame:0  
          TX packets:16397063  errors:0  dropped:0  overruns:0  carrier:0  
          collisions:0  txqueuelen:0  
          RX bytes:2488566815 (2.3 GiB)  TX bytes:2488566815 (2.3 GiB)  
  
[root@CentOS-64-64-minimal ~]#
```

In this example, the IP address of the server is **188.40.69.75**. Write the address down or copy it. You will need it later on installing.

### Discovering IP addresses in case of NAT

Often the WCS server has to be placed behind a corporate NAT or on the rented server in a data center that does not provide dedicated static IP addresses. A good example of this case is Amazon EC2, where servers are behind NAT.



When the WCS server is to be deployed behind NAT, you need to discover two IP addresses.



1. The external IP address or Public IP not necessarily belongs to the WCS server, but traffic that goes to this external IP address should be obligatory redirected to WCS
2. Network interface IP address or Private IP - this is a real address assigned to the network interface. Here WCS will open sockets to interact with the network

For example, a test server on Amazon EC2 has the external IP address of: `54.189.212.111`, and the network interface address is: `172.30.21.25`. If your server is behind NAT in regard to browser clients, write down or copy these IP addresses. You will need them later, when you will configure the server.

## Automatic and semi-automatic IP addresses detection

Since build [5.2.921](#), WCS tries to automatically detect server IP addresses on startup if they are not set, or server is starting in cloud environment supporting AWS API or Google Cloud API endpoints. If IP addresses cannot be detected on startup, it may be done manually using the following command

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

This command:

- tries to detect IP addresses automatically
- if this cannot be done, asks user for additional data
- writes results to `flashphoner.properties` file

It is recommended to use this command if there are some problems with IP addresses detection behind NAT.