

# WCS on Google Cloud Platform

Since build [5.2.679](#), WCS can be deployed on Google Cloud Platform as standalone media server, CDN with low latency and CDN with load balancing between nodes.

Anyway, before deploying, the following should be prepared:

- an active GCP account and a project in the account
- a WCS [license](#) to activate on server/servers
- optionally, domain names to bind to servers static IPs and corresponding SSL certificates

## Server deployment

### Create and launch VM instance

1. In Google Cloud console go to **Compute Engine** - **VM instances** section and click **Create VM instance** to start VM creation. Choose the server name, datacenter region and zone, VM configuration

← Create an instance

To create a VM instance, select one of the options:

- New VM instance** > Create a single VM instance from scratch
- New VM instance from template > Create a single VM instance from an existing template
- New VM instance from machine image > Create a single VM instance from an existing machine image
- Marketplace > Deploy a ready-to-go solution onto a VM instance

**Name** ?  
Name is permanent  
test-origin-1

**Labels** ? (Optional)  
+ Add label

**Region** ?  
Region is permanent  
europe-west3 (Frankfurt)

**Zone** ?  
Zone is permanent  
europe-west3-c

**Machine configuration**

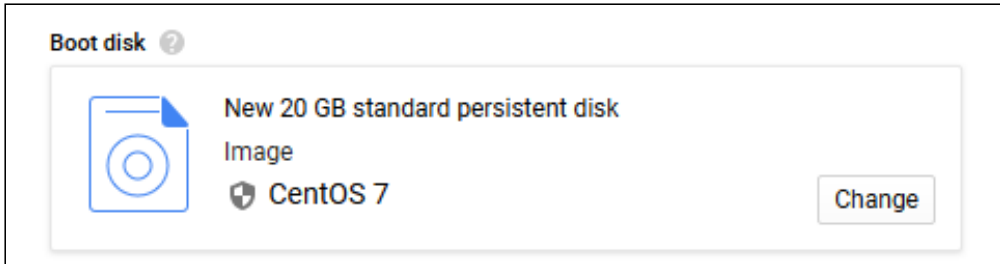
**Machine family**  
General-purpose | Memory-optimized  
Machine types for common workloads, optimized for cost and flexibility

**Series**  
N1  
Powered by Intel Skylake CPU platform or one of its predecessors

**Machine type**  
n1-standard-1 (1 vCPU, 3.75 GB memory)

	vCPU	Memory
	1	3.75 GB

2. In **Boot disk** section click **Change**. Choose CentOS 7.6 base image



**Boot disk**

Select an image or snapshot to create a boot disk, or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#).

[Public images](#) [Custom images](#) [Snapshots](#) [Existing disks](#)

Operating system  
CentOS

Version  
CentOS 7

x86\_64 built on 20200618, supports Shielded VM features

Boot disk type  Size (GB)

3. On **Security** tab add the public SSH key if you do not have project SSH keys

**SSH Keys**

Block project-wide SSH keys  
When checked, project-wide SSH keys cannot access this instance [Learn more](#)

You have 0 SSH keys

Enter public SSH key

✕

[+ Add item](#)

4. On **Network** tab choose external and internal IP addresses:

5. if the server supposed to be Origin in CDN, it is recommended to reserve a static internal IP address;

6. if there should be external entry points to the server (for example, to use for publishing/playing streams), it is recommended to reserve a static external IP address to

bind domain name to

### Network interface ⤴

**Network**  
default

**Subnetwork**  
default

**Internal IP**  
10.156.0.3

**Internal IP type**  
Ephemeral ⌵

**Alias IP ranges**

**Subnet range** Primary (10.156.0.0/20) ⌵ **Alias IP range** Example: 10.0.1.0/24 or /32 ✕

+ Add IP range

⤴ Hide alias IP ranges

**External IP** ?  
Ephemeral ⌵

**Network Service Tier** ?

Premium (Current project-level tier, [change](#)) ?

Standard (europe-west3) ?

**IP forwarding**  
Off

**Public DNS PTR Record** ?

Enable

PTR domain name

Done Cancel

7. Click **Create**

Management **Security** Disks Networking Sole Tenancy

**Shielded VM** ?  
Turn on all settings for the most secure configuration.

- Turn on Secure Boot ?
- Turn on vTPM ?
- Turn on Integrity Monitoring ?

**SSH Keys**  
These keys allow access only to this instance, unlike [project-wide SSH keys](#) [Learn more](#)

- Block project-wide SSH keys  
When checked, project-wide SSH keys cannot access this instance [Learn more](#)

gcp

```
gTaJ8gvi6x9RQB6niVuTN80cK3H1A4xINxQ29GGxWJ  
wXe4kRKIkM4QnxUTsNNsC6yc/d57Ur773518Tevf3v  
4GcWQ9gCPvoIIHZqE79zB0xbRhggjj4ED1rRbC11ug0  
uGO+2kaChLkxHehJ+Xotz/NW0Az0cwkW1YSZGDditT  
vICrIDvRXFD0nuSuj8EpBU3Jjj54zChTI2k4dUDcPY  
kA/bAgy2tF5Ajc50ZCPIVcOu74R1/7RZ1YqgIJ1g+L  
aB gcp
```

+ Add item

^ Less

You can always create instance templates free of charge. Your free trial credit won't be used.

**Create** Cancel

Equivalent [REST](#) or [command line](#)

The server instance will be created and launched

VM instances [CREATE INSTANCE](#) [IMPORT VM](#) [REFRESH](#) [START](#) [STOP](#) [RESET](#) [DELETE](#)

Filter VM instances Columns

<input type="checkbox"/> Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/> test-origin-1	europe-west3-c			10.156.0.3 (nic0)	35.234.93.218	SSH

## Firewall rules setup

Firewall rules affect all the instances in the project, so they should be set up once

1. Go to **VPC network** - **Firewall** section and create **wcs-ports** rule

**VPC network** | **Create a firewall rule**

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

**Name \***  
wcs-ports

Lowercase letters, numbers, hyphens allowed

**Description**  
WCS specific ports rule

**Logs**  
Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. [Learn more](#)

On  
 Off

2. Allow incoming (ingress in terms of GCP) traffic from any node

**Network \***  
default

**Priority \***  
1000

Priority can be 0 - 65535 [Check priority of other firewall rules](#)

**Direction of traffic ?**

Ingress  
 Egress

**Action on match ?**

Allow  
 Deny

**Targets**  
All instances in the network

**Source filter**  
IP ranges

**Source IP ranges \***  
0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

**Second source filter**  
None

3. Specify WCS ports and click **Create**

**Protocols and ports** ?

Allow all

Specified protocols and ports

tcp : 554, 1935, 8080-8084, 8443-8445, 8888, 9091, 30000-33000

udp : 1935, 30000-33000

Other protocols

protocols, comma separated, e.g. ah, sctp

▼ **DISABLE RULE**

**CREATE**    CANCEL

Equivalent [REST](#) or [command line](#)

## WCS installation and configuration

1. Install JDK. It is recommended to use at least JDK 12 or 14 if high load is planning

```
#!/bin/bash
sudo rm -rf jdk*
curl -s
https://download.java.net/java/GA/jdk12.0.2/e482c34c86bd4bf8b56c0b35558996b9
12.0.2_linux-x64_bin.tar.gz | tar -zx
[ ! -d jdk-12.0.2/bin ] && exit 1
sudo mkdir -p /usr/java
[ -d /usr/java/jdk-12.0.2 ] && sudo rm -rf /usr/java/jdk-12.0.2
sudo mv -f jdk-12.0.2 /usr/java
[ ! -d /usr/java/jdk-12.0.2/bin ] && exit 1
sudo rm -f /usr/java/default
sudo ln -sf /usr/java/jdk-12.0.2 /usr/java/default
sudo update-alternatives --install "/usr/bin/java" "java" "/usr/java/jdk-
12.0.2/bin/java" 1
sudo update-alternatives --install "/usr/bin/jstack" "jstack"
"/usr/java/jdk-12.0.2/bin/jstack" 1
sudo update-alternatives --install "/usr/bin/jcmd" "jcmd" "/usr/java/jdk-
12.0.2/bin/jcmd" 1
sudo update-alternatives --install "/usr/bin/jmap" "jmap" "/usr/java/jdk-
12.0.2/bin/jmap" 1
sudo update-alternatives --set "java" "/usr/java/jdk-12.0.2/bin/java"
sudo update-alternatives --set "jstack" "/usr/java/jdk-12.0.2/bin/jstack"
sudo update-alternatives --set "jcmd" "/usr/java/jdk-12.0.2/bin/jcmd"
sudo update-alternatives --set "jmap" "/usr/java/jdk-12.0.2/bin/jmap"
```

## 2. Install accessory tools and libraries

```
sudo yum install -y tcpdump mc iperf3 fontconfig
```

## 3. Stop firewalld (it is not necessary to block any ports on VM level because firewall rules were set up on project level)

```
sudo systemctl stop firewalld  
sudo systemctl disable firewalld
```

## 4. Disable SELinux

```
sudo setenforce 0
```

## 5. Install WCS

```
curl -OL  
https://flashphoner.com/downloads/builds/WCS/5.2/FlashphonerWebCallServer-  
5.2.xxx.tar.gz  
tar -xzf FlashphonerWebCallServer-5.2.xxx.tar.gz  
cd FlashphonerWebCallServer-5.2.xxx  
sudo ./install.sh
```

Where `xxx` is WCS actual build number

## 6. Activate your license

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

## 7. Configure WCS (below the example of Origin server settings to publish WebRTC and RTMP streams)

```
flashphoner.properties [----] 29 L:[ 1+23 24/ 40] *(680 / 981b) 0010 0x00A
# Config flashphoner.properties
# To get more settings:
# ssh -p 2001 admin@localhost
# default password: admin
# show node-settings
# show node-settings | grep port

#server ip
ip                =34.107.12.11
ip_local          =10.156.0.3

#webrtc ports range
media_port_from   =31001
media_port_to     =32000

#codecs
codecs            =opus,alaw,ulaw,g729,speex16,g722,mpeg4-generic,telephone-event,h264,vp8,flv,mpv
codecs_exclude_sip    =mpeg4-generic,flv,mpv
codecs_exclude_streaming =flv,telephone-event
codecs_exclude_sip_rtmp =opus,g729,g722,mpeg4-generic,vp8,mpv

#websocket ports
ws.port           =8080
wss.port          =8443

cdn_enabled=true
cdn_ip=10.156.0.3
cdn_role=origin
cdn_nodes_resolve_ip=false

# Request keyframes from WebRTC publishers every 5 seconds
periodic_fir_request=true

# Disable RTMP keepalives to publish from OBS
keep_alive.enabled=websocket,rtmp

client_mode=false

rtc_ice_add_local_component=true
```

Do not change IP addresses in `ip`, `ip_local` и `cdn_ip` settings, they will be configured automatically on WCS startup.

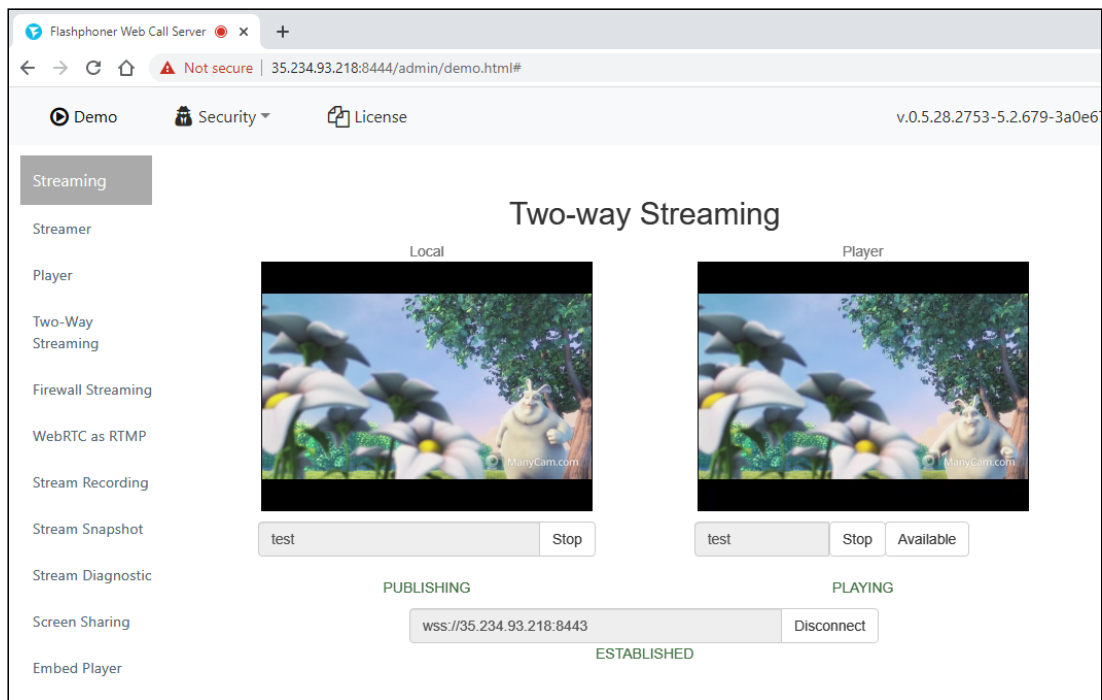
## WCS starting and testing

### 1. Start WCS

```
sudo systemctl start webrtcserver
```

2. Enter to WCS web interface, open Two Way Streaming example, publish and play `test` stream





## CDN deployment

1. [Create](#) and [configure](#) Origin server as described above
2. Create and configure Edge server (below the example of Edge server setting to play WebRTC streams)

```
flashphoner.properties [-M--] 0 L:[ 1+36 37/ 37] *(874 / 874b) <EOF>
# Config flashphoner.properties
# To get more settings:
# ssh -p 2001 admin@localhost
# default password: admin
# show node-settings
# show node-settings | grep port

#server ip
ip                =34.107.12.11
ip_local          =10.156.0.5

#webrtc ports range
media_port_from   =31001
media_port_to     =32000

#codecs
codecs            =opus,alaw,ulaw,g729,speex16,g722,mpeg4-generic,telephone-event,h264,vp8,flv,mpv
codecs_exclude_sip    =mpeg4-generic,flv,mpv
codecs_exclude_streaming =flv,telephone-event
codecs_exclude_sip_rtmp =opus,g729,g722,mpeg4-generic,vp8,mpv

#websocket ports
ws.port          =8080
wss.port         =8443

cdn_enabled=true
cdn_ip=10.156.0.5
cdn_role=edge
cdn_point_of_entry=10.156.0.3
cdn_nodes_resolve_ip=false

client_mode=false

rtc_ice_add_local_component=true

http_enable_root_redirect=false
```

Do not change IP addresses in `ip`, `ip_local` и `cdn_ip` settings, they will be configured automatically on WCS startup. Set the `cdn_point_of_entry` parameter to Origin server static internal IP address

## CDN testing

1. Start WCS on Origin and Edge VM instances.
2. Go to Origin web interface and publish `test` stream in Two Way Streaming example

3. Go to Edge web interface and play **test** stream in Player example

