

# GCP load balancer with autoscale quick setup

## Overview

WCS Google Cloud Platform instances support TCP Network load balancer.

WebSocket connections will be distributed between active load balancer instances. In case a scaling policy is executed (when the policy target – e.g., CPU load on instance - is reached) and new instances are launched, they will be added to the load balancer.

The following components would be required

- Disk image to use in instance template
- Instance template to create new server instances while autoscaling
- Autoscale instance group
- Load balancer
- Server health checks

Let's try to deploy CDN for WebRTC streams including one Origin server and a group of Edge servers (from 1 to 3 instances) with CPU load autoscaling.

## Prepare server instances

1. Create one Origin and one Edge server as described [here](#). Reserve a static internal IP address to Origin server. Reserve external static IP address to use in load balancer
2. Configure CDN on Origin server side

```
cdn_enabled           = true
cdn_ip                = <origin_internal_ip>
cdn_role              = origin
cdn_nodes_resolve_ip = false
```

3. Configure CDN on Edge server side

```
cdn_enabled           = true
cdn_ip                = <edge_internal_ip>
cdn_point_of_entry    = <origin_internal_ip>
cdn_role              = edge
cdn_nodes_resolve_ip = false
```

4. Add the following parameter to Edge server settings

```
http_enable_root_redirect=false
```


5. Prepare and [import](#) SSL certificates to Origin and Edge servers. It is not recommended to use Let's Encrypt because it requires to update Edge disk image every 3 months.

## Create Edge disk image

1. Stop Edge server instance
2. In Google Cloud console, go to **Compute Engine** - **Images** section and click **Create image**. Choose Edge instance disk as disk image source and click **Create**

[←](#) Create an image

**Name** ?  
Name is permanent

test-edge-image-1 

**Source** ?  
Disk

**Source disk** ?  
test-edge-1

**Location** ?  
 Multi-regional  
 Regional  
eu (European Union) (default)

**Family** (Optional) ?

**Description** (Optional)

**Labels** ? (Optional)

+ Add label


**Encryption**  
Data is encrypted automatically. Select an encryption key management solution.

Google-managed key  
No configuration required

Customer-managed key  
Manage via Google Cloud Key Management Service

Customer-supplied key  
Manage outside of Google Cloud

---

Your free trial credit will be used for this image. [GCP Free Tier](#) 

**Create** **Cancel**

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

Do not delete source Edge instance after disk image creation, it will be necessary for Edge disk image updating.

# Create Edge instance template

1. Go to **Compute Engine** - **Instance templates** section and click **Create instance template**. Choose instance VM configuration

← **Create an instance template**

Describe a VM instance once and then use that template to create groups of identical instances [Learn more](#)

**Name** ?  
Name is permanent

test-edge-template

**Machine configuration**

**Machine family**

General-purpose Memory-optimized Compute-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


⌵ CPU platform and GPU

**Container** ?

Deploy a container image to this VM instance. [Learn more](#)

2. In **Boot disk** section click **Change**

**Boot disk** ?

 New 20 GB standard persistent disk  
Image  
test-edge-image-1

Change

On **Custom images** tab choose Edge disk image

### Boot disk

Select an image to create a boot disk. The image determines the operating system installed on the instance. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#).

**Public images** **Custom images**

Show images from  
Test GCP LB

Show deprecated images

Image  
test-edge-image-1

Created on Jun 25, 2020, 1:53:31 PM

Boot disk type <sup>?</sup> Size (GB) <sup>?</sup>

Standard persistent disk 20

3. On **Security** tab add the public SSH key if you do not have project SSH keys and 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](#)

## Create autoscaling group

1. Go to **Compute Engine** - **Instance groups** section and click **Create instance group**.  
Choose instance group region and zone, select Edge instance template

← Create an instance group

To create an instance group, select one of the options:

- New managed instance group**  
A group of VMs created from a template.  
Supports autohealing, autoscaling, auto updating, regional deployments, and load balancing.
- New managed instance group for stateful workloads**  
A group of VMs created from a template, with preserved disks and metadata individually for each VM.  
Supports autohealing, auto updating, regional deployments, and load balancing for stateful workloads.
- New unmanaged instance group**  
A group of existing VMs that you manage.  
Supports load balancing.

Organize VM instances in a group to manage them together. [Instance groups](#)

**Name** ⓘ  
Name is permanent  
test-edge-instance-group

**Description** (Optional)

**Location**  
To ensure higher availability, select a multiple zone location for an instance group. [Learn more](#)

Single zone  
 Multiple zones

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

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

Specify port name mapping (Optional)

**Instance template** ⓘ  
test-edge-template

**Number of instances**  
Based on autoscaling configuration

2. Choose **Autoscale** mode by **CPU utilization** metric, set metric target value to **80%** and set maximum number of instances to **3**

**Autoscaling**  
 Use autoscaling to allow automatic resizing of this instance group for periods of high and low load. [Autoscaling groups of instances](#) ↗

**Autoscaling mode**  
 Autoscale

**Autoscaling metrics**  
 Use metrics to determine when to autoscale the group.  
[Autoscaling policy and target utilization](#) ↗

**New metric** ^

**Metric type**  
 CPU utilization

**Target CPU utilization** ?  
 80 %

Done Cancel

+ Add new metric

**Cool down period** ?  
 Specify how long to wait for a new instance before taking its metrics into account.  
[Cool down period](#) ↗  
 60 seconds

**Minimum number of instances** ? 1      **Maximum number of instances** ? 3

**Scale In Controls** ?  
 Prevent a sudden drop in the number of running VM instances in the group by controlling the process of scaling in. [Learn more](#)

Enable Scale In Controls

Delete autoscaling configuration

3. Enable `Auto healing` and create a health check. Set TCP protocol, port `8081` and request `/health-check`



## ← Create a health check

Health checking mechanisms determine whether VM instances respond properly to traffic. You cannot create a legacy health check using this page. For more information, refer to the [Health Checks Concepts](#) documentation.

Name

wcs-health-check



Description

WCS health checking

Scope

Global

Regional

Protocol

TCP



Port

8081



Proxy protocol

NONE



Request

/health-check



Response



Logs

Turning on Health check logs can increase costs in Stackdriver.

On

Off

Configure health criteria and click **Create** to return to group setup

### Health criteria

Define how health is determined: how often to check, how long to wait for a response, and how many successful or failed attempts are decisive

Check interval  seconds [?](#)      Timeout  seconds [?](#)

Healthy threshold  consecutive successes [?](#)

Unhealthy threshold  consecutive failures [?](#)

---

You can create this health check free of charge

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

4. Expand **Advanced creation options** and enable **Do not retry machine creation**, then click **Create**

### Advanced creation options

Advanced configuration controlling how the instance group is created

**Do not retry machine creation.**  
If Compute Engine hits a usage limit or error during instance creation, then reduce the instance group size to create as many instances as possible.

[^ Hide advanced creation options](#)

---

Your free trial credit will be used for VM instances in this group. [GCP Free Tier](#) [↗](#)

[Create](#) [Cancel](#)

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

Autoscaling instance group will be created, and one instance will be launched

Instance groups									
CREATE INSTANCE GROUP REFRESH DELETE									
Instance groups are collections of VM instances that use load balancing and automated services, like autoscaling and autohealing. <a href="#">Learn more</a>									
Filter resources Columns									
Name	Zone	Instances	Template	Group type	Creation time	Recommendation	Autoscaling	In use by	
<input checked="" type="checkbox"/> test-edge-instance-group	europe-west3-c	1	test-edge-template	Managed	Jun 25, 2020, 2:09:10 PM		On: Target CPU utilization 80%		

## Create load balancer

1. Go to **Network** - **Load balancers** section and click **Create load balancer**. Choose **TCP Load Balancing**

The screenshot shows the 'Create a load balancer' page with three main options:

- HTTP(S) Load Balancing**: Layer 7 load balancing for HTTP and HTTPS applications. Includes options for internet-facing or internal and single or multi-region.
- TCP Load Balancing**: Layer 4 load balancing or proxy for applications that rely on TCP/SSL protocol. Includes options for internet-facing or internal and single or multi-region.
- UDP Load Balancing**: Layer 4 load balancing for applications that rely on UDP protocol. Includes options for internet-facing or internal and single-region.

2. Choose external load balancer **From internet to my VMs** and its region

The screenshot shows the 'Create a load balancer' page with a questionnaire:

Please answer a few questions to help us select the right load balancing type for your application

**Internet facing or internal only**

Do you want to load balance traffic from the Internet to your VMs or only between VMs in your network?

From Internet to my VMs  
 Only between my VMs

**Multiple regions or single region**

Do you want to place the backends for your load balancer in a single region or across multiple regions?

Multiple regions (or not sure yet)  
 Single region only

**Continue**

3. In **Backend configuration** section, on **Select existing instance groups** tab select Edge instance group and set session affinity to client IP and protocol

**New TCP load balancer**

**Backend configuration**

Name <sup>?</sup>  
Name is permanent  
test-lb

**Backend configuration**  
Your backend is configured →

**Frontend configuration**  
You have not configured your frontend yet

**Review and finalize**  
Optional

Name <sup>?</sup>  
test-lb

Region <sup>?</sup>  
europe-west3

Backends <sup>?</sup>

test-edge-instance-group ×  
No more instance groups available in this region

Backup pool <sup>?</sup> (Optional)  
None

Failover ratio <sup>?</sup>  
10 %

Health check <sup>?</sup>  
wcs-lb-health-check (HTTP)  
port: 8081, timeout: 5s, check interval: 10s, unhealthy threshold: 3 attempts

Session affinity <sup>?</sup>  
Client IP and protocol

4. Choose **Create health check**. Create server health check, set port **8081** and request **/**

**Create a health check**

Autohealing instance groups and load balancing use health checks to detect when an instance is unresponsive [Learn more](#)

Name <sup>?</sup>  
Name is permanent  
wcs-lb-health-check

Description (Optional)  
WCS health check for load balancer

Protocol  
HTTP

Port <sup>?</sup>  
8081

Request path <sup>?</sup>  
/

**More**

Health criteria

Define how health is determined: how often to check, how long to wait for a response, and how many successful or failed attempts are decisive

Check interval ?

10 seconds

Timeout ?

5 seconds

Healthy threshold ?

2 consecutive successes

Unhealthy threshold ?

3 consecutive failures

Save and continue

Cancel

5. In `Frontend configuration` section create TCP port configurations for ports `8081`, `8080`, `8443`, `8444` for HTTP(S) and WS(S). Set external static IP address to load balancer

**New TCP load balancer**

**Name** ⓘ  
Name is permanent

**Backend configuration**  
Your backend is configured

**Frontend configuration**  
Your frontend is configured →

ⓘ **Review and finalize**  
Optional

**Frontend configuration**

Specify an IP address, port and protocol. This IP address is the frontend IP for your clients requests.

**New Frontend IP and port** 🗑️ ⬆️

**Name (Optional)** ⓘ  
Name is permanent

**Add a description**

**Protocol**  
TCP

**Network Service Tier** ⓘ  
 Premium (Current project-level tier, [change](#)) ⓘ  
 Standard ⓘ

**IP**  
test-lb-entry-point (34.107.5.128) ▾

**Port**

[+ Add Frontend IP and port](#)

**New TCP load balancer**

**Name** ⓘ  
Name is permanent

**Backend configuration**  
Your backend is configured

**Frontend configuration**  
Your frontend is configured →

ⓘ **Review and finalize**  
Optional

**Frontend configuration**

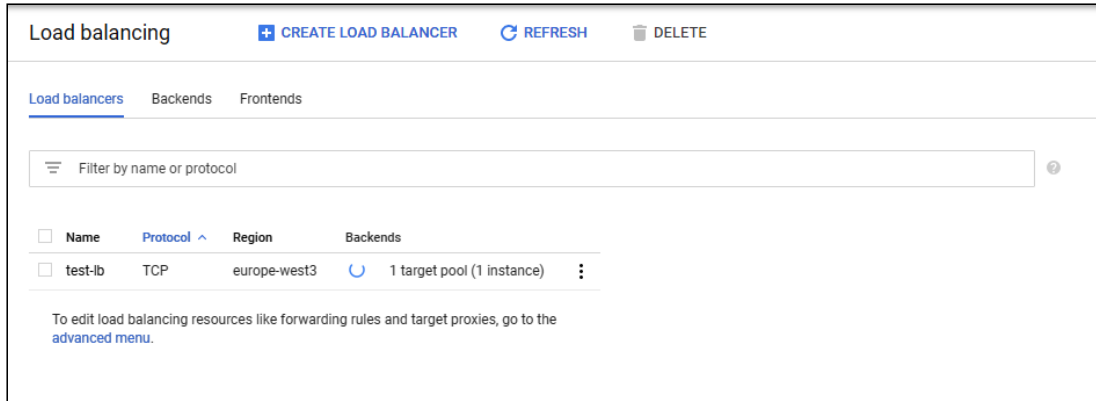
Specify an IP address, port and protocol. This IP address is the frontend IP for your clients requests.

Protocol:TCP, IP:34.107.5.128, Port:8081	<i>Not saved</i> <input type="button" value="✎"/>
Protocol:TCP, IP:34.107.5.128, Port:8080	<i>Not saved</i> <input type="button" value="✎"/>
Protocol:TCP, IP:34.107.5.128, Port:8444	<i>Not saved</i> <input type="button" value="✎"/>
Protocol:TCP, IP:34.107.5.128, Port:8443	<i>Not saved</i> <input type="button" value="✎"/>

[+ Add Frontend IP and port](#)

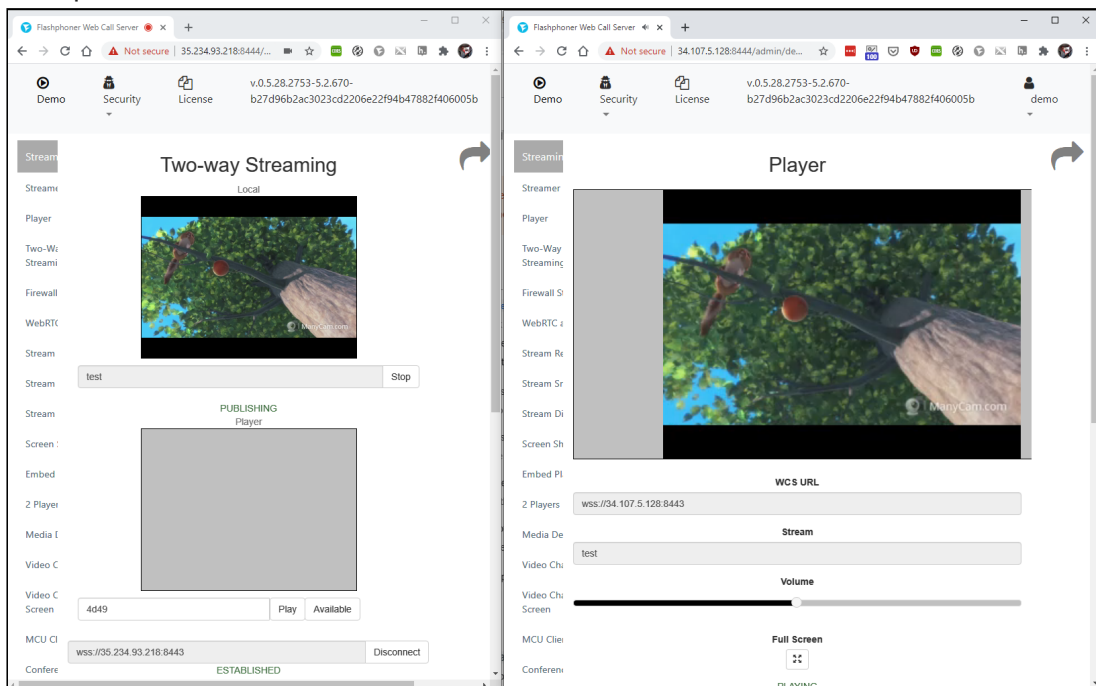
You can add another port configurations (**1935** for RTMP subscribers, **8082**, **8445** for HLS etc depending on Edge use case)

6. Click **Create**. Load balancer will start



## Load balancer testing

1. Go to Origin web interface and publish **test** stream in Two Way Streaming example
2. Go to Edge web interface using load balancer IP address. Play the **test** stream in Player example



## Updating Edge servers settings

To update Edge servers settings, for example, to update SSL certificates, Edge disk image must be updated as follows:

1. Disable autoscaling and delete all Edge instances in Edge instance group
2. Launch source Edge server instance

3. Update the settings as needed (for example, update SSL certificates)
4. Stop source Edge instance
5. Delete Edge disk image
6. Create new Edge disk image with the same name (for example, `test-edge-image-1`)
7. Enable autoscaling in Edge instance group (autoscaling settings will be preserved)