

# AWS load balancer with auto scale quick setup

## Overview

WCS Amazon instances support AWS 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

- AMI on the basis of which new instances will be created for scaling out
- Load Balancer
- Launch Configuration
- Auto Scaling Group

## Launching AWS Auto Scaling group with classic load balancer from custom AMI

Load balancer with autoscaling deployment from custom AMI can be useful for long term projects (months and years). In this case, AWS Marketplace image will be more expensive due to hourly payment, therefore it is recommended to buy and activate WCS monthly subscription.

### Attention

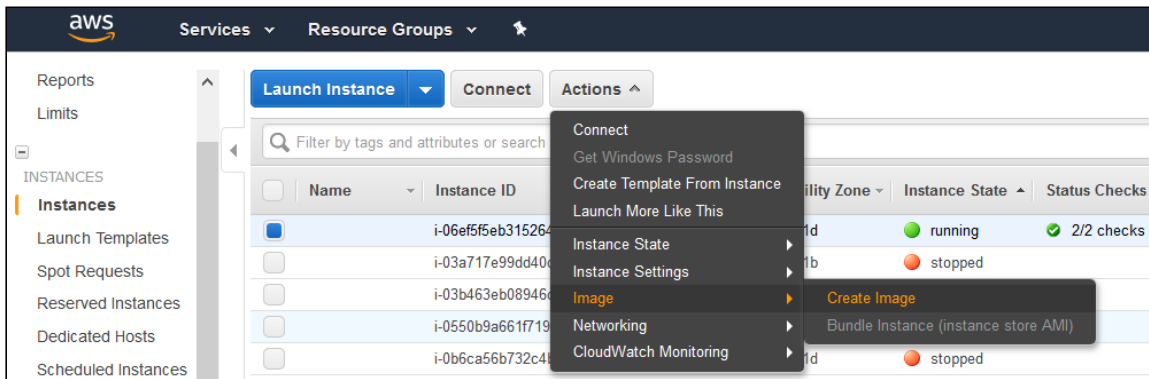
Amazon claims classic load balancer will work till August 2022.

## 1. Create new AMI

1.1. [Launch an instance](#) from a FlashphonerWebCallServer AMI and configure the WCS

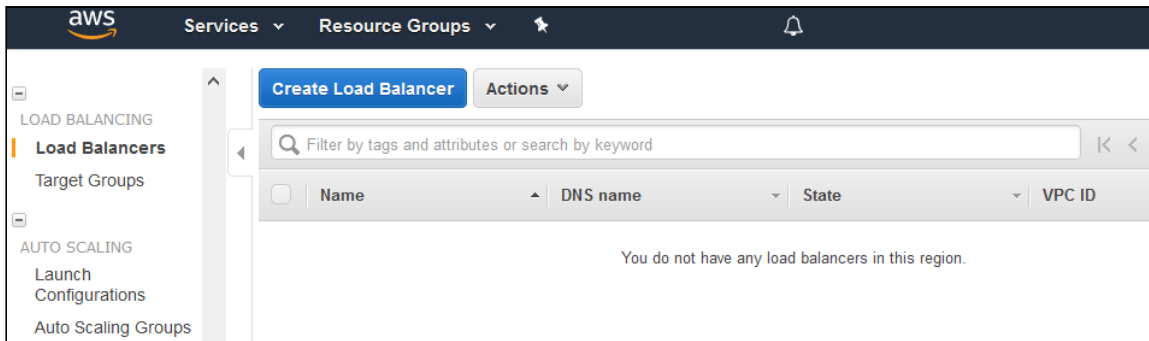
- activate license
- import certificates
- change configuration settings as required

1.2. In AWS console, select the instance and then **Actions** - **Image** - **Create Image** and create a new image:

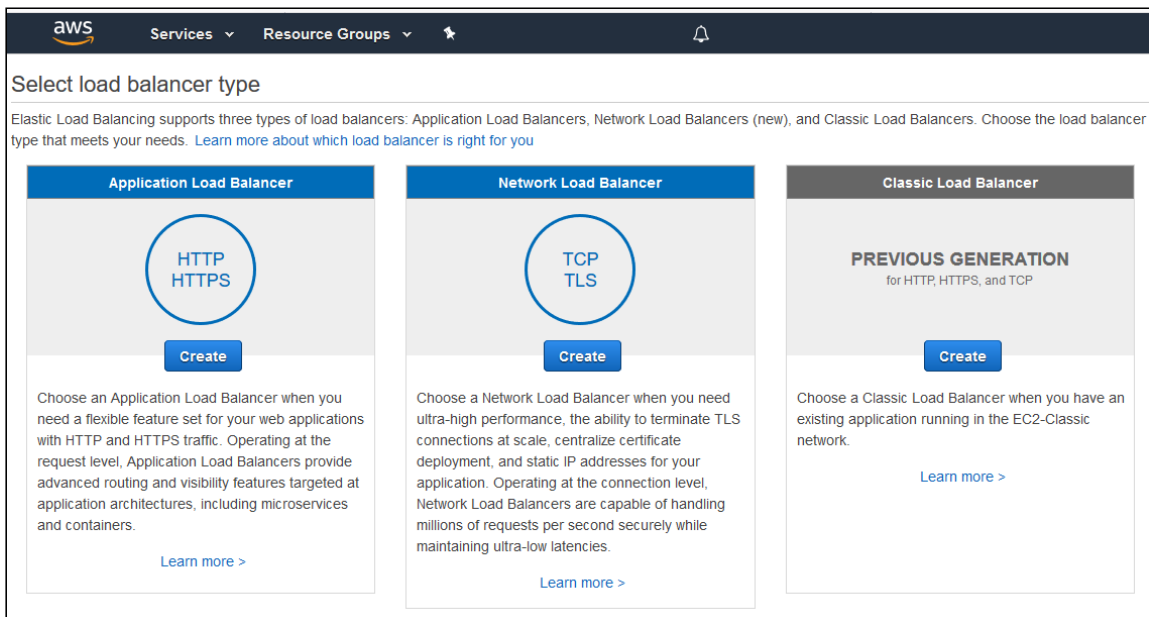


## 2. Create new Load Balancer

2.1. In AWS console, go to **EC2** - **Load Balancers** and click **Create Load Balancer**



2.2 Select **Classic Load Balancer** type (This type allows specifying port for health check)



2.3. When defining load balancer, add required protocols. For example TCP, port 8080 for WebSocket connections (`ws:host:8080`)

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

### Step 1: Define Load Balancer

#### Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

**Load Balancer name:**

**Create LB Inside:**

**Create an internal load balancer:**  (what's this?)

**Enable advanced VPC configuration:**

**Listener Configuration:**

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
<input type="text" value="HTTP"/>	<input type="text" value="8081"/>	<input type="text" value="HTTP"/>	<input type="text" value="8081"/>
<input type="text" value="TCP"/>	<input type="text" value="8080"/>	<input type="text" value="TCP"/>	<input type="text" value="8080"/>

2.4. Assign a security group.

2.5. Configure health check

The URL for health check is

- for HTTP: `http://WCS_ADDRESS:8081/?action=stat`
- for HTTPS: `https://WCS_ADDRESS:8444/?action=stat`

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

### Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

**Ping Protocol:**

**Ping Port:**

**Ping Path:**

**Advanced Details**

**Response Timeout:**  seconds

**Interval:**  seconds

**Unhealthy threshold:**

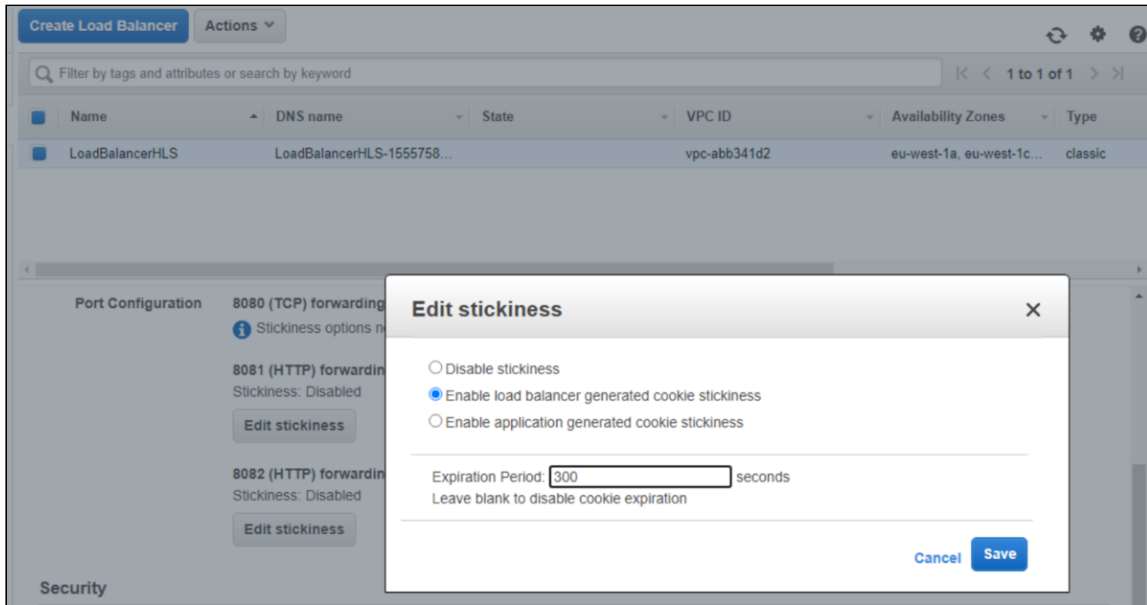
**Healthy threshold:**

2.6. Add existing EC2 instances as required

By default, cross-zone load balancing is enabled to distribute traffic between all available availability zones in your region.

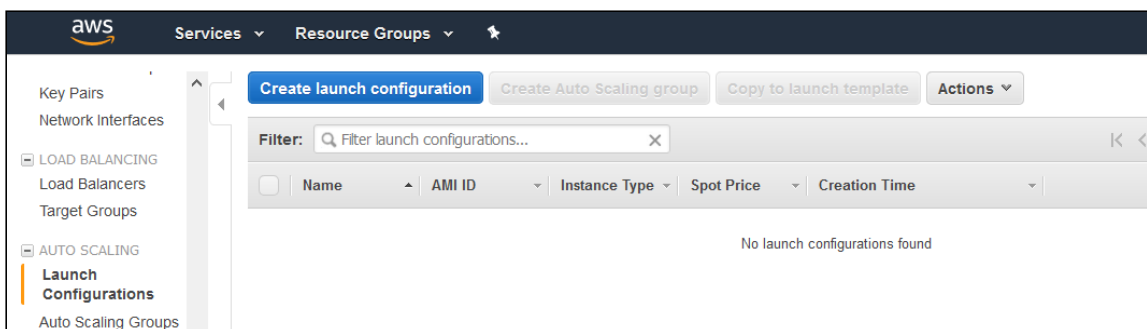
## 2.7. Complete the wizard to create the load balancer

## 2.8. Enable stickiness for HTTP/HTTPS LB ports

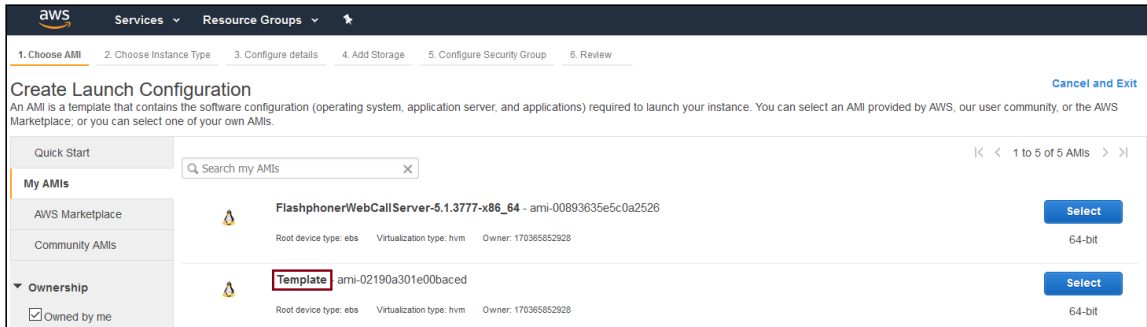


## 3. Create new Launch Configuration

3.1. In AWS console, go to **EC2** - **Launch Configurations** and click **Create launch configuration**

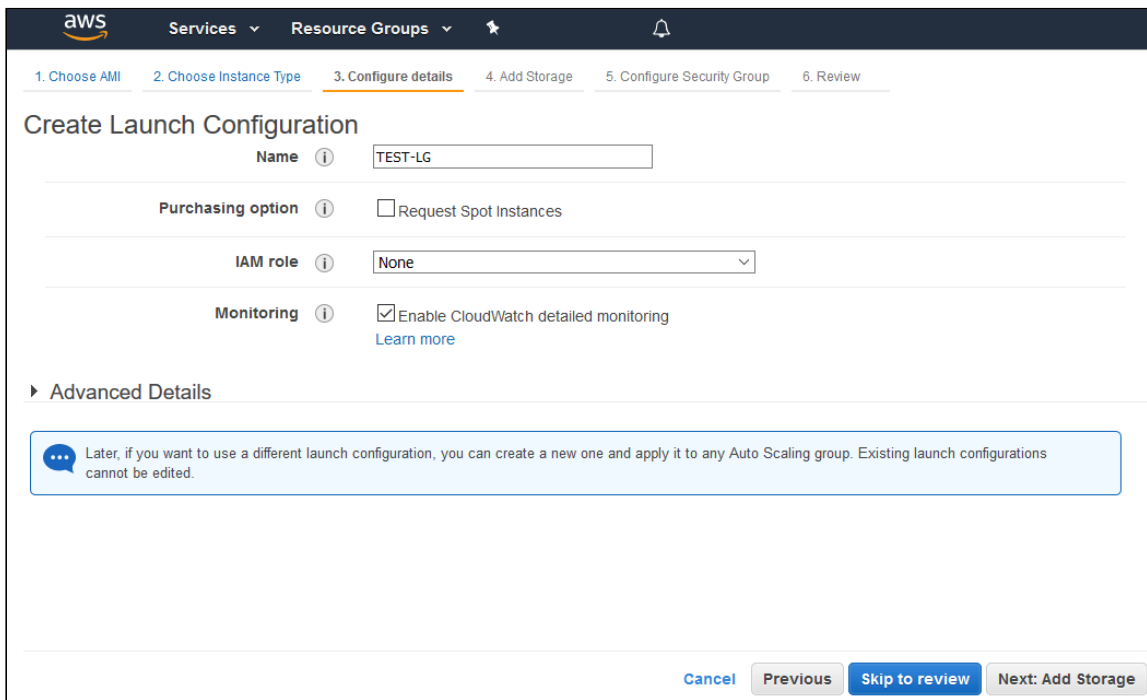


3.2. When choosing AMI, select the AMI previously created from an instance with required WCS configuration



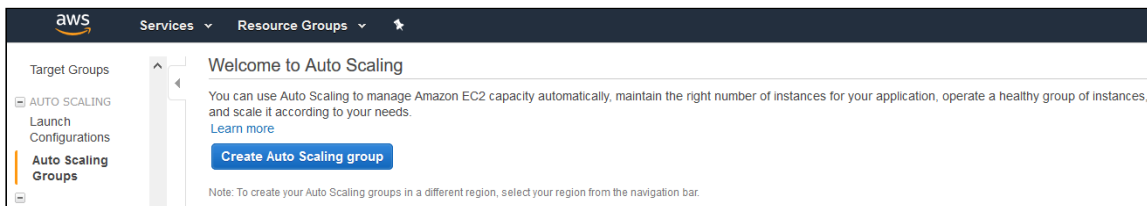
### 3.3. Complete the wizard to create the configuration

Detailed monitoring, where data is available in 1-minute periods, can be enabled when configuring details.



## 4. Create new Auto Scaling group

4.1. In AWS console, go to **EC2** - **Auto Scaling Groups** and click **Create Auto Scaling group**



4.2. Select the required launch configuration or template, or create a new one

**aws** Services ▾ Resource Groups ▾ 🔍

## Create Auto Scaling Group Cancel and Exit

Complete this wizard to create your Auto Scaling group. First, choose either a launch configuration or a launch template to specify the parameters that your Auto Scaling group uses to launch instances.

**Launch Configuration**

You can continue to use your launch configurations if they support the Amazon EC2 features you need. [Learn more](#)

**Launch Template** New

Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned. [Learn more](#)

[Create new launch template](#)

Create a new launch configuration

Use an existing launch configuration

Filter launch configurations... << 1 to 1 of 1 Launch Configurations >>

Name	AMI ID	Instance Type	Spot Price	Security Groups
TEST-LG	ami-0cf30f44be371890d	t2.small		sg-03b1bc951522cb94a

Cancel Next Step

### 4.3. Configure Auto Scaling group details

- add required subnets
- add required load balancer

**aws** Services ▾ Resource Groups ▾ 🔍

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

## Create Auto Scaling Group Cancel and Exit

Group name

Launch Configuration

Group size Start with  instances

Network  C Create new VPC

Subnet   C Create new subnet

Each instance in this Auto Scaling group will be assigned a public IP address.

▼ Advanced Details

Load Balancing  Receive traffic from one or more load balancers [Learn about Elastic Load Balancing](#)

Classic Load Balancers

Target Groups

Health Check Type  ELB  EC2

Health Check Grace Period  seconds

Monitoring  Enable CloudWatch detailed monitoring [Learn more](#)

Instance Protection

Service-Linked Role  C View Role in IAM

Cancel Next: Configure scaling policies

### 4.4. Configure scaling policies

aws Services Resource Groups

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

### Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. [Learn more](#) about scaling policies.

Keep this group at its initial size  
 Use scaling policies to adjust the capacity of this group

Scale between  and  instances. These will be the minimum and maximum size of your group.

**Scale Group Size** ✕

Name:

Metric type:

Target value:

Instances need:  seconds to warm up after scaling

Disable scale-in:

[Scale the Auto Scaling group using step or simple scaling policies](#) ⓘ

[Cancel](#) [Previous](#) [Review](#) [Next: Configure Notifications](#)

4.5. Complete the wizard to create the auto scaling group

## Launching Application Load Balancer using existing instances

Sometimes, a certain set of instances is already launched and configured (Origin servers group in CDN, for example), and load balancing between those servers should be set up. Use Application Load Balancer to do this.

### 1. Instances launching

Launch and configure server instances as needed by [this manual](#).

### 2. Application Load Balancer creation

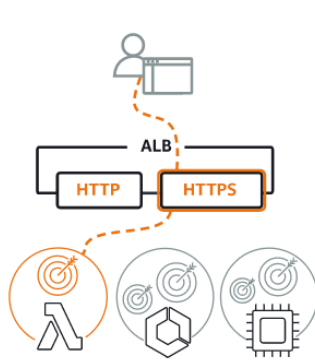
2.1. In EC2 Console menu, go to **Load balancers** - **Load balancers** section and click **Create load balancer**". Click **Create** for Application Load Balancer

## Select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

### Load balancer types

#### Application Load Balancer [Info](#)

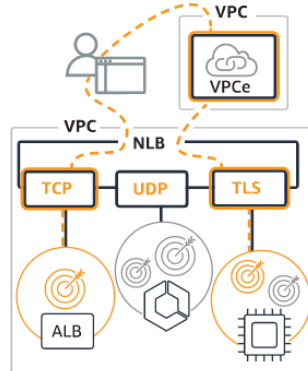


Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create



#### Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

#### Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

2.2. Enter the balancer name, choose **Internet-facing** type (suggested by default)

## Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

### ► How Application Load Balancers work

#### Basic configuration

##### Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

TEST-APP-LB

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

##### Scheme [Info](#)

Scheme cannot be changed after the load balancer is created.

Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

##### IP address type [Info](#)

Select the type of IP addresses that your subnets use.

IPv4

Recommended for internal load balancers.

Dualstack

Includes IPv4 and IPv6 addresses.

2.3. In **Network mapping** section choose a subnets needed



**Network mapping** [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

**VPC** [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-  
vpc-e305fc9a  
IPv4: 172.31.0.0/16

**Mappings** [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection. Subnets cannot be removed after the load balancer is created, but additional subnets can be added.

**eu-west-1a**

Subnet  
subnet-003b4c5a

**IPv4 settings**

Assigned by AWS

**eu-west-1b**

**eu-west-1c**

## 2.4. Choose or create security groups as needed

**Security groups** [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups  
Select security groups

Create new security group [Info](#)

default sg-9c127cdf ✕  
VPC: vpc-e305fc9a

Note that a standard WCS ports set should be opened in load balancer security group

### Create new based on seller settings

A new security group will be generated by AWS Marketplace. It is based on recommended settings for Web Call Server 5 version 5.2.267.

#### Name your security Group

#### Description

Connection Method	Protocol	Port Range	Source (IP or Group)
SSH	tcp	22	Anywhere <input type="text" value="0.0.0.0/0"/>
	tcp	554	Anywhere <input type="text" value="0.0.0.0/0"/>
	udp	30000-33000	Anywhere <input type="text" value="0.0.0.0/0"/>
	tcp	8080-8084	Anywhere <input type="text" value="0.0.0.0/0"/>
	tcp	8443-8445	Anywhere <input type="text" value="0.0.0.0/0"/>
	tcp	8888	Anywhere <input type="text" value="0.0.0.0/0"/>
	tcp	9091	Anywhere <input type="text" value="0.0.0.0/0"/>
	tcp	1935	Anywhere <input type="text" value="0.0.0.0/0"/>
	udp	1935	Anywhere <input type="text" value="0.0.0.0/0"/>

Rules with source of 0.0.0.0/0 allows all IP addresses to access your instance. We recommend limiting access to only known IP addresses.

2.5. In **Listeners and routing** section add Websocket port listener (mandatory) and HTTP port listener (if needed)

**Listeners and routing** [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTP:8080 Remove

Protocol: HTTP : Port: 8080 (1-65535)

Default action: Forward to test-ws-app-group (Target type: Instance, IPv4) HTTP ↻

[Create target group](#)

▼ Listener HTTP:8081 Remove

Protocol: HTTP : Port: 8081 (1-65535)

Default action: Forward to test-http-app-group (Target type: Instance, IPv4) HTTP ↻

[Create target group](#)

[Add listener](#)

A target group must be created for every listener, see [below](#).

## 2.6. Click Create load balancer

▼ **Add-on services - optional**

Additional AWS services can be integrated with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the "Integrated Services" tab for the selected load balancer.

AWS Global Accelerator [Learn more](#)

Create an accelerator to get static IP addresses and improve the performance and availability of your applications. [Additional charges apply](#)

▶ **Tags - optional**

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

**Summary**

Review and confirm your configurations. [Estimate cost](#)

<p><b>Basic configuration</b> <a href="#">Edit</a></p> <p>TEST-APP-LB</p> <ul style="list-style-type: none"> <li>Internet-facing</li> <li>IPv4</li> </ul>	<p><b>Security groups</b> <a href="#">Edit</a></p> <ul style="list-style-type: none"> <li>default <a href="#">sg-9c127cdf</a></li> </ul>	<p><b>Network mapping</b> <a href="#">Edit</a></p> <p>VPC <a href="#">vpc-e305fc9a</a></p> <ul style="list-style-type: none"> <li>eu-west-1a <a href="#">subnet-003b4c5a</a></li> </ul>	<p><b>Listeners and routing</b> <a href="#">Edit</a></p> <ul style="list-style-type: none"> <li>HTTP:8080 defaults to <a href="#">test-ws-app-group</a></li> <li>HTTP:8081 defaults to <a href="#">test-http-app-group</a></li> </ul>
<p><b>Add-on services</b> <a href="#">Edit</a></p> <p>None</p>	<p><b>Tags</b> <a href="#">Edit</a></p> <p>None</p>		

**Attributes**

ⓘ Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Cancel Create load balancer

Load balancer is created

✔ Successfully created load balancer: **TEST-APP-LB**  
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers

**Suggested next steps**

- Review, customize, or enable attributes for your load balancer and listeners using the **Description** and **Listeners** tabs within **TEST-APP-LB**.
- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within **TEST-APP-LB**.

[View load balancer](#)

### 3. Websocket listener target group creation

#### 3.1. Choose target type **Instances** (supposed by default), set group name

Step 1  
**Specify group details**

Step 2  
Register targets

### Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

**Basic configuration**  
Settings in this section cannot be changed after the target group is created.

Choose a target type

- Instances**
  - Supports load balancing to instances within a specific VPC.
- IP addresses**
  - Supports load balancing to VPC and on-premises resources.
  - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
  - Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Lambda function**
  - Facilitates routing to a single Lambda function.
  - Accessible to Application Load Balancers only.
- Application Load Balancer**
  - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
  - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name  
  
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol:  Port:

#### 3.2. Set WCS instance Websocket port (8080), choose subnet and protocol version (HTTP1)

Protocol Port

HTTP :

VPC  
Select the VPC with the instances that you want to include in the target group.

-  
vpc-e305fc9a  
IPv4: 172.31.0.0/16

Protocol version

HTTP1  
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2  
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC  
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

3.3. In **Health check** configure instance health check using HTTP port (8081) and statistics page query `/?action=stat`

### Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

Health check path  
Use the default path of "/" to ping the root, or specify a custom path if preferred.  
  
Up to 1024 characters allowed.

▼ **Advanced health check settings** Restore defaults

Port  
The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.

Traffic port

Override  
  
1-65535

Healthy threshold  
The number of consecutive health check successes required before considering an unhealthy target healthy.  
  
2-10

Unhealthy threshold  
The number of consecutive health check failures required before considering a target unhealthy.  
  
2-10

Timeout  
The amount of time, in seconds, during which no response means a failed health check.  
 seconds  
2-120

Then click **Next**

**Interval**  
The approximate amount of time between health checks of an individual target

seconds

5-300

**Success codes**  
The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

► **Tags - optional**  
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Cancel](#) [Next](#)

3.4. At **Register targets** page select instances as needed and click **Include as pending** below

**Register targets**  
Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. Once you are satisfied with your selections, click Register pending targets.

**Available instances (2/2)**

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	IPv4 address	Subnet ID
<input checked="" type="checkbox"/>	i-0dec078d94e7520ef		running	Web Call Server 5-5-2-944-systemd246-AutogenByAWSMP-	eu-west-1b	3.249.98.141	subnet-41072d27
<input checked="" type="checkbox"/>	i-0dbaf422e637b2d9a		running	Web Call Server 5-5-2-944-systemd246-AutogenByAWSMP-	eu-west-1b	34.240.11.186	subnet-41072d27

2 selected

Ports for the selected instances  
Ports for routing traffic to the selected instances.

1-65535 (separate multiple ports with commas)

Then click **Register pending targets**

**Review targets**

**Targets (2)**

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	IPv4 address	Subnet ID
X	Pending	i-0dbaf422e637b2d9a		8080	running	Web Call Server 5-5-2-944-systemd246-AutogenByAWSMP-	eu-west-1b	34.240.11.186	subnet-41072d27
X	Pending	i-0dec078d94e7520ef		8080	running	Web Call Server 5-5-2-944-systemd246-AutogenByAWSMP-	eu-west-1b	3.249.98.141	subnet-41072d27

2 pending

[Cancel](#) [Register pending targets](#)

Target group is created

Successfully created target group: test-ws-app-group

EC2 > Target groups

**Target groups (3)** Info

<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input type="checkbox"/>	test-ws-app-group	arn:aws:elasticloadbalancin...	8080	HTTP	Instance	-	vpc-e305f9a

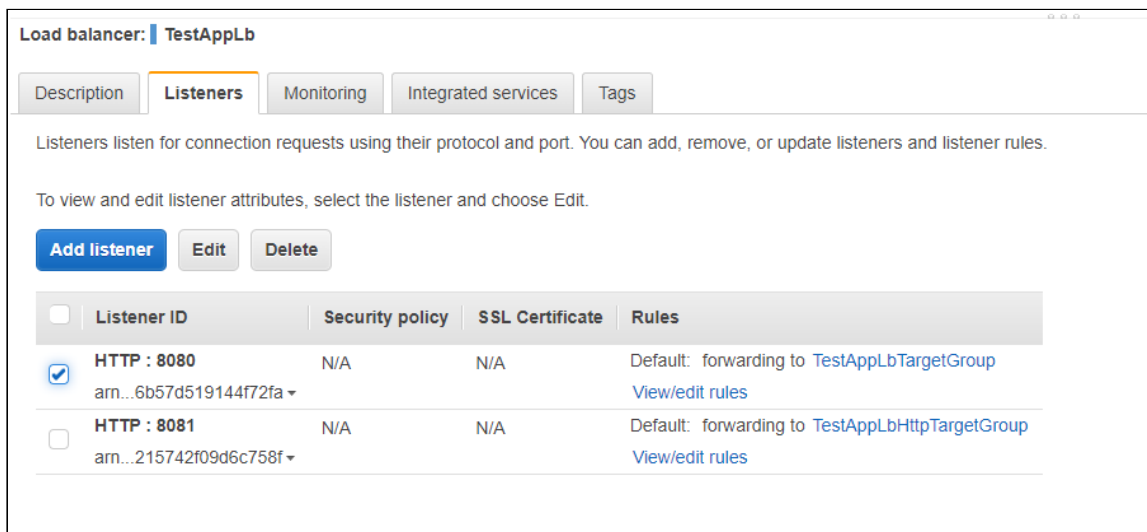
Select a target group above.

Application load balancer using this group will forward requests to it after at least one of the group instances passes health check.

## 4. Listener parameters configuration

If Application Load Balancer is created to use in Autoscaling group (see below), HTTPS listener cannot be configured on creation, only HTTP. In this case, listener parameters should be changed.

4.1. In EC2 Console section **Load balancers** - **Load balancers** choose **Listeners** tab for load balancer to configure. Choose Websocket listener and click Edit



Load balancer: **TestAppLb**

Description **Listeners** Monitoring Integrated services Tags

Listeners listen for connection requests using their protocol and port. You can add, remove, or update listeners and listener rules.

To view and edit listener attributes, select the listener and choose Edit.

**Add listener** Edit Delete

<input type="checkbox"/>	Listener ID	Security policy	SSL Certificate	Rules
<input checked="" type="checkbox"/>	<b>HTTP : 8080</b> arn...6b57d519144f72fa ▾	N/A	N/A	Default: forwarding to <a href="#">TestAppLbTargetGroup</a> <a href="#">View/edit rules</a>
<input type="checkbox"/>	<b>HTTP : 8081</b> arn...215742f09d6c758f ▾	N/A	N/A	Default: forwarding to <a href="#">TestAppLbHttpTargetGroup</a> <a href="#">View/edit rules</a>

4.2. Choose HTTPS protocol and set Secure Websocket port (8443 for example)

## Edit listener

arn:aws:elasticloadbalancing:eu-north-1:170365852928:listener/app/TestAppLb/8e650022f70c3d2f/6b57d519144f72fa

### Listener details

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

Protocol : Port  
HTTPS : 8443  
1-65535

### Default actions [Info](#)

Specify the default actions for traffic on this listener. Default actions apply to traffic that does not meet the conditions of rules on your listener. Rules can be configured after the listener is created.

▼ 1. Forward to [Info](#) Remove

Target group		Weight (0-999)	
TestAppLbTargetGroup Target type: Instance, IPv4	HTTP	1	×
Traffic distribution: 100%			
Select a target group		0	×

[Create target group](#)

Enable group-level stickiness [Info](#)  
If you enable stickiness for your target group, requests routed to it remain in the same group for the duration you specify.

Add action ▼

4.3. In **Secure listener settings** section choose SSL certificate to use with domain assigned to load balancer entry point or create a new one. Then click **Save changes**

### Secure listener settings [Info](#)

**Security policy**  
Your load balancer uses a Secure Socket Layer (SSL) negotiation configuration, known as a security policy, to negotiate SSL connections with clients.

ELBSecurityPolicy-2016-08

[Compare security policies](#)

**Default SSL certificate**  
The certificate used if a client connects without SNI protocol, or if there are no matching certificates. You can add more certificates after you create the load balancer.

From ACM : \*.flashphoner.com  
d62e2c7d-23a5-4ef2-8244-6b7dc92d9246

[Request new ACM certificate](#)

Cancel Save changes



Load balancer listener parameters are changed and will be applied immediately

## Edit listener

**Suggested next steps**

- Review or customize your listener. [Edit listener](#)

[View listeners](#)

## Launching AWS Auto Scaling group with load balancing from AWS Marketplace AMI

Load balancer with autoscaling deployment from AWS Marketplace AMI may be useful for periodic servers group launching, for example, during the event (lasting for hours, days, weeks). In this case, WCS monthly subscription may be more expensive than AWS hourly payment, therefore it is recommended to use AWS Marketplace AMI.

### 1. Create launch template

1.1. In EC2 Console go to **Instances** - **Launch Templates** section and click **Create launch template**. Launch template creation wizard will open. Enter template name and description

EC2 > Launch templates > Create launch template

## Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

### Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance [Info](#)  
Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ **Template tags**

▶ **Source template**

## 1.2. Choose latest FlashphonerWebCallServer image

### Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

#### Amazon machine image (AMI) [Info](#)

AMI

FlashphonerWebCallServer-5.2.629-x86\_64-hourly-01e37234-6170-4b8d-98b...  
ami-035ddfe555ad2e6f8

Catalog: AWS Marketplace    architecture: 64-bit (x86)    virtualization: hvm

## 1.3. Choose instance type, key pair for SSH access to an instance, security group

### Instance type [Info](#)

Instance type

t2.micro    Free tier eligible    [Instance types](#)

Family: General purpose    1 vCPU    1 GiB Memory  
On-Demand Linux pricing: 0.0126 USD per Hour  
On-Demand Windows pricing: 0.0172 USD per Hour

### Key pair (login) [Info](#)

Key pair name

test\_userdata    [Create new key pair](#)

### Network settings

Networking platform [Info](#)

Virtual Private Cloud (VPC)  
Launch into a virtual network in your own logically isolated area within the AWS cloud

EC2-Classic  
Launch into a single flat network that you share with other customers

Security groups [Info](#)

Select security groups

WCS sg-0ec50e70028ff86d7 ✕  
VPC: vpc-e305fc9a

## 1.4. Set disk size and parameters for instances

### Storage (volumes) [Info](#)

▼ Volume 1 (AMI Root)  
AMI Volumes are not included in the template unless modified

Volume type <a href="#">Info</a> EBS	Device name - required <a href="#">Info</a> /dev/sda1	Snapshot <a href="#">Info</a> snap-0fee0446fee3252a5
Size (GiB) <a href="#">Info</a> <input type="text" value="10"/>	Volume type <a href="#">Info</a> <input type="text" value="General purpose SSD (gp2)"/>	IOPS <a href="#">Info</a> <input type="text" value="2000"/>
Delete on termination <a href="#">Info</a> <input type="text" value="Yes"/>	Encrypted <a href="#">Info</a> <input type="text" value="No"/>	Key <a href="#">Info</a> <input type="text" value="MyKey"/>

1.5. Expand **Advanced details** section. Insert custom update and setup script to **User data** text box

User data [Info](#)

```
#!/bin/bash

# Stop WCS before reconfiguring
PID=$(pgrep -f 'com.flashphoner.server.Server' | grep -v bash)
if [ -n "$PID" ]; then
    service webcallserver stop
fi

# Update WCS to the latest build (optionally, set to false if you don't)
UPDATE=true
if $UPDATE; then
    cd /tmp
```

User data has already been base64 encoded

The setup script example to update WCS to latest build and to configure CDN Edge server for WebRTC playback

**Edge setup script** >

1.6. Click **Create launch template**

### Resource tags [Info](#)

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

**Add tag**

50 remaining (Up to 50 tags maximum)

### Network interfaces [Info](#)

No network interfaces are currently included in this template. Add a network interface to include it in the launch template.

**Add network interface**

▶ **Advanced details** [Info](#)

Cancel **Create launch template**

Launch template will be created

EC2 > Launch templates

Launch templates (1) [Info](#) Refresh Actions Create launch template

Filter by tags or properties or search by keyword

Launch template ID	Launch template name	Default version	Latest version	Create time
lt-04d44d426947bae18	TestTemplate	1	1	2020-07-14T06:40:07.000Z

## 2. Create Auto scaling group

2.1. In EC2 Console go to **Instances** - **Auto Scaling Groups** section and click **Create an Auto Scaling Group**. Autoscaling group creation wizard will open. Enter group name

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
**Choose launch template or configuration**

Step 2  
Configure settings

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
Review

## Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

**Name**

**Auto Scaling group name**  
Enter a name to identify the group.

TestAutoscalingGroup

Must be unique to this account in the current Region and no more than 255 characters.

**Launch template** Info [Switch to launch configuration](#)

**Launch template**  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

TestTemplate

[Create a launch template](#)

**Version**

Default (1)

[Create a launch template version](#)

<b>Description</b> Test autoscaling launch template	<b>Launch template</b> <a href="#">TestTemplate</a> lt-04d44d426947bae18	<b>Instance type</b> -
<b>AMI ID</b> ami-035ddfe555ad2e6f8	<b>Security groups</b> -	<b>Security group IDs</b> -
<b>Key pair name</b>		

## 2.2 Choose launch template, set **Latest** version

**Launch template** Info
[Switch to launch configuration](#)

---

**Launch template**  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

TestTemplate ▼ ↻

[Create a launch template](#) ↗

**Version**

Latest (2) ▼ ↻

[Create a launch template version](#) ↗

<p><b>Description</b></p> <p>-</p>	<p><b>Launch template</b></p> <p><a href="#">TestTemplate</a> <span style="font-size: 0.8em;">↗</span></p> <p>lt-04d44d426947bae18</p>	<p><b>Instance type</b></p> <p>t2.micro</p>
<p><b>AMI ID</b></p> <p>ami-035ddfe555ad2e6f8</p>	<p><b>Security groups</b></p> <p>-</p>	<p><b>Security group IDs</b></p> <p><a href="#">sg-0ec50e70028ff86d7</a> <span style="font-size: 0.8em;">↗</span></p>
<p><b>Key pair name</b></p> <p>test_userdata</p>		

---

**Additional details**

<p><b>Storage (volumes)</b></p> <p>/dev/sda1</p>	<p><b>Date created</b></p> <p>Tue Jul 14 2020 14:02:03 GMT+0700 (Novosibirsk Standard Time)</p>	
--	---	--

2.3. Set instances distribution percentage (on demand/spot). By default **70 %** on demand will be set, it is recommended to raise this value to **100 %**

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template or configuration

Step 2  
**Configure settings**

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
Review

## Configure settings [Info](#)

Configure the settings below. Depending on whether you chose a launch template, these settings may include options to help you make optimal use of EC2 resources.

### Purchase options and instance types [Info](#)

Adhere to launch template  
The launch template determines the purchase option (On-Demand or Spot) and instance type.

**Combine purchase options and instance types**  
Specify how much On-Demand and Spot capacity to launch and multiple instance types (optional). This choice is most helpful for optimizing the scale and cost for a fleet of instances.

### Instances distribution

**Optional On-Demand base**  
Specify how much On-Demand capacity the Auto Scaling group should have for its base portion. The maximum group size will be increased (but not decreased) to this value.

On-Demand Instances

**On-Demand percentage above base**  
Define the percentage split of On-Demand Instances and Spot Instances for your additional capacity beyond the base portion.

% On-Demand

% Spot

**Spot allocation strategy per Availability Zone**

Capacity optimized - Launch Spot Instances optimally based on the available Spot capacity (recommended)

Lowest price - Launch Spot Instances from the lowest priced instance pools

Number of lowest priced Spot Instance pools to diversify across

Value must be between 1 and 20

## 2.4 Choose instance types

### Instance types [Info](#)

Choose the instance types that best suit the needs of your application.

Primary instance type Weight [Info](#)

1.  1vCPU 1 Gib Memory

**i** Your launch template does not specify an instance type. As a result, Adhere to launch template cannot be chosen. You can continue by adding an instance type above.

Additional instance types

[Redo recommendations](#)

2.  1vCPU 1 Gib Memory

## 2.5. Choose VPC and subnets for instances

**Network** [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

**VPC**

vpc-e305fc9a  
172.31.0.0/16 Default

[Create a VPC](#)

**Subnets**

Select subnets

[Create a subnet](#)

Cancel Previous Skip to review Next

## 2.6. Choose **Attach to a new load balancer**

Step 1  
Choose launch template or configuration

Step 2  
Choose instance launch options

Step 3 (optional)  
**Configure advanced options**

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
Review

**Configure advanced options** [Info](#)

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

**Load balancing - optional** [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**Attach to a new load balancer**

Define a new load balancer to create for attachment to this Auto Scaling group.

**Load balancer type**

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, [visit the Load Balancing console](#).

Application Load Balancer  
HTTP, HTTPS

Network Load Balancer  
TCP, UDP, TLS

## 2.7. Choose **Application Load balancer** type, set the name, choose **Internet-facing**, set availability zones and corresponding subnets



### Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

**Load balancer type**  
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, [visit the Load Balancing console](#).

**Application Load Balancer**  
HTTP, HTTPS
  **Network Load Balancer**  
TCP, UDP, TLS

**Load balancer name**  
Name cannot be changed after the load balancer is created.

TestAppLb

**Load balancer scheme**  
Scheme cannot be changed after the load balancer is created.

Internal
  **Internet-facing**

**Network mapping**  
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

**VPC**  
[vpc-5e65c237](#)

**Availability Zones and subnets**  
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

eu-north-1a 
  
 eu-north-1c 
  
 eu-north-1b

2.8. In **Listeners and routing** section set Websocket port (8080), choose **Create a target group** and set the target group name to be created

**Listeners and routing**  
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

**Protocol** 
**Port** 
**Default routing (forward to)**

**New target group name**  
An instance target group with default settings will be created.

**Tags - optional**  
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

50 remaining

Then click **Next**

### Additional settings - optional

Monitoring [Info](#)

Enable group metrics collection within CloudWatch

Cancel Previous Skip to review Next

## 2.9. Set the maximum group size

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template or configuration

Step 2  
Configure settings

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
**Configure group size and scaling policies**

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
Review

### Configure group size and scaling policies [Info](#)

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

#### Group size - optional [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

Minimum capacity

Maximum capacity

#### Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

**Target tracking scaling policy**  
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

None

Scaling policy name

## 2.10. Select scaling policy by CPU utilization, set target value and instance warming time

### Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

**Target tracking scaling policy**  
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

None

Scaling policy name

Metric type

Target value

Instances need  
 seconds warm up before including in metric

Disable scale in to create only a scale-out policy

### Instance scale-in protection - *optional*

**Instance scale-in protection**  
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

Enable instance scale-in protection

## 2.11. Review group parameters

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template or configuration

Step 2  
Configure settings

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
**Review**

## Review [Info](#)

Step 1: Choose launch template or configuration Edit

**Group details**

Auto Scaling group name  
TestAutoscalingGroup

**Launch template**

Launch template	Version	Description
<a href="#">TestTemplate</a> lt-04d44d426947bae18	Latest	

Step 2: Configure settings Edit

**Purchase options and instance types**

**Instances distribution**

On-Demand base	On-Demand and Spot percentages	Spot allocation strategy
Designate the first 0 instances as On-Demand	100 % On-Demand 0 % Spot	Capacity optimized

**Instance types**

Instance type	vCPUs	Memory	Network performance
1. t2.micro	1 vCPU	1 GiB	Low to Moderate

2.12. Click **Create Auto Scaling group**

Step 5: Add notifications Edit

**Notifications**

No notifications

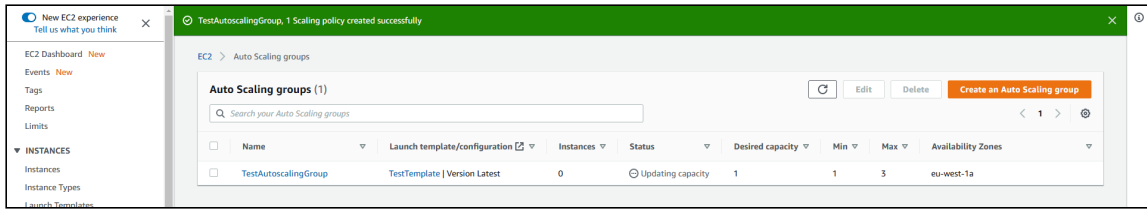
Step 6: Add tags Edit

**Tags (0)**

Key	Value	Tag new instances
No tags		

Cancel Create Auto Scaling group

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



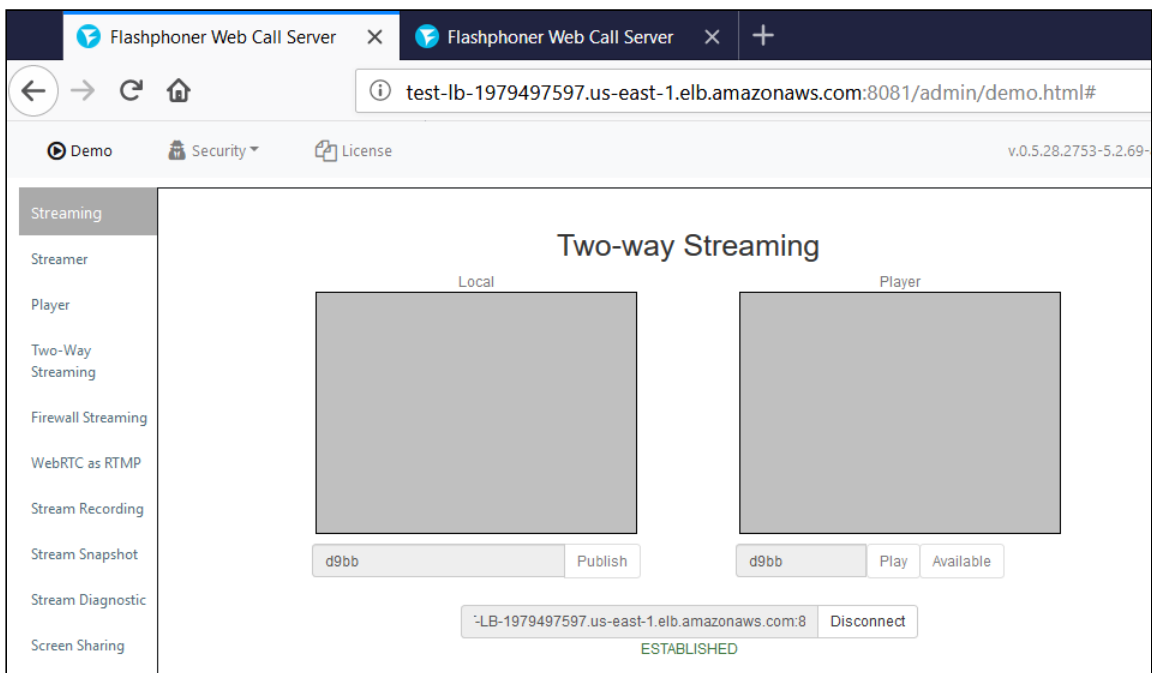
2.13. Configure load balancer listener as described [above](#)

## Testing

If load balancer has no running instances, then a new instance will be started when an auto scaling group receiving traffic from the load balancer is created. More instances will be started in case scaling is triggered. (For testing purposes, streaming with transcoding – e.g., streaming RTMP to [auto created mixer](#) – can be used to load server CPU). All the started instances will be auto added to the corresponding load balancer.

When an instance (one or more of the added to the balancer) is in service, ws-connection can be done to, e.g., `ws://Load balancer DNS name:8080`.

Two-way Streaming example opened either by the balancer or an instance address can be used to establish Websocket connection:



To verify that the connections are distributed between active load balancer instances, use the stats page: `http://WCS_ADDRESS:8081/?action=stat`

Open the page for each of the instances to see the `connections_websocket` number:

The image shows a browser window with two tabs. The active tab is titled "ec2-3-82-201-0.compute-1.am: X". The address bar contains the URL "ec2-3-82-201-0.compute-1.amazonaws.com:8081/?action=stat". The main content area displays the following text:

```
-----Connection Stats-----
connections=2
connections_rtmp=0
connections_websocket=2
-----Port Stats-----
ports_media_free=499
ports_media_busy=0
ports_media_quarantine=0
-----Stream Stats-----
```

The line "connections\_websocket=2" is highlighted with a red rectangular box.