

Welcome

Topics

- [Who Should Read This Guide](#)
- [How This Guide Is Organized](#)
- [Amazon Elastic Load Balancing Resources](#)

Elastic Load Balancing is a cost-effective and easy-to-use web service that distributes application loads between two or more EC2 instances.

This is the *Elastic Load Balancing Developer Guide*. This section describes who should read this guide, how the guide is organized, and other resources related to Elastic Load Balancing.

The following Amazon Web Services (AWS) products will occasionally be referred to using the following abbreviated forms; all copyrights and legal protections still apply.

Full Name	Abbreviated Form
Amazon Elastic Compute Cloud	Amazon EC2
Amazon Machine Image	AMI

Who Should Read This Guide

This guide is intended for developers, as well as those in IT/Operations, who are building web applications or services hosted in the Amazon Elastic Compute Cloud.

Required Knowledge and Skills

Use of this guide assumes you are familiar with the following:

- XML. For an overview, go to the [*W3 Schools XML Tutorial*](#).
- Web services. For an overview, go to the [*W3 Schools Web Services Tutorial*](#).
- A programming language for consuming a web service and any related tools.
- Programming Amazon Elastic Cloud Computing (EC2) applications or services that use Amazon EC2 web service APIs. For information about Amazon EC2, go to the [*Amazon EC2 product information page*](#).

How This Guide Is Organized

This guide is organized into several major sections described in the table below.

Information	Relevant Sections
General information about Elastic Load Balancing	What is Elastic Load Balancing?
Conceptual information about Elastic Load Balancing	<ul style="list-style-type: none">• Conceptual Overview of Elastic Load Balancing• Terminology and Key Concepts
API Reference and Programming Related Information	<ul style="list-style-type: none">• API Reference• Using Elastic Load Balancing
Typographic and symbol conventions	Document Conventions

There are more sections under each category to help you with more specific information as well as a glossary and an index. Most sections are written to stand on their own with some cross dependencies. You should, in general, be able to look up the information you need and go back to work. However, you can also read through the major sections sequentially to get in-depth knowledge about the service.

Amazon Elastic Load Balancing Resources

The table below lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon Elastic Load Balancing Technical FAQ	The FAQ covers the top 20 questions developers have asked about this product.
Release notes	The release notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.
Premium Support	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
Amazon Elastic Load Balancing product information	The primary web page for information about Amazon Elastic Load Balancing.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse etc.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

What's New

This What's New is associated with the 2009-05-15 release of Elastic Load Balancing. This guide was last updated on September 17, 2009.

The following table describes the important changes since the last release of Elastic Load Balancing Developer Guide.

Change	Description	Release Date
Updated/added content	Bug fixes; added new region information.	09 September 2009
Updated content	Bug fixes.	21 July 2009
Added content	Instructions for Command Line Tools have been added to the User Scenarios section.	28 May 2009
New Service	This is the first release of Elastic Load Balancer Developer Guide. Future updates and changes will be noted here.	18 May 2009

Introduction to Elastic Load Balancing

Topics

- [What is Elastic Load Balancing?](#)
- [Elastic Load Balancing Concepts](#)
- [Elastic Load Balancing Conceptual Overview](#)
- [Availability Zones and Regions](#)

What is Elastic Load Balancing?

Elastic Load Balancing is a cost-effective and easy to use web service to help you improve availability and scalability of your application. It makes it easy for you to distribute application loads between two or more EC2 instances. Elastic Load Balancing enables availability through redundancy and supports traffic growth of your application.

Benefits

The core benefits of Elastic Load Balancing:

- **Application scalability**—Support traffic growth and distribute the application load among multiple EC2 instances by sending requests to the most suitable instance. Add new instances without interrupting your application.
- **Application availability**—Increase the availability of your application by protecting against application and instance failure.
- **Cost-effective**—Pay only for what you use, as you use it, with no up-front commitments. There are no pre-purchase commitments required and there is no minimum amount of use required.

Elastic Load Balancing Concepts

This section introduces you to Elastic Load Balancing terminology and concepts. Many of the concepts introduced in this chapter are discussed in more specific contexts in later chapters. The concepts are briefly presented here to give you a basic understanding of common Amazon Load Balancing Service terms.

For more information, please see the introductory and conceptual overviews - [What is Elastic Load Balancing?](#) and [Conceptual Overview of Elastic Load Balancing](#).

LoadBalancer

A *LoadBalancer* is represented by a DNS name and a set of ports and provides the destination to which all requests intended for your application should be directed. Each LoadBalancer can distribute requests to multiple application instances. LoadBalancers can span multiple [Availability Zones](#) within an EC2 [region](#), but cannot span multiple regions.



Note

To create or work with a load balancer in a specific region, the corresponding regional service endpoint should be used. In the United States, the endpoint is:

- <https://us-east-1.elasticloadbalancing.amazonaws.com>

In Europe, the endpoint is:

- <https://eu-west-1.elasticloadbalancing.amazonaws.com>

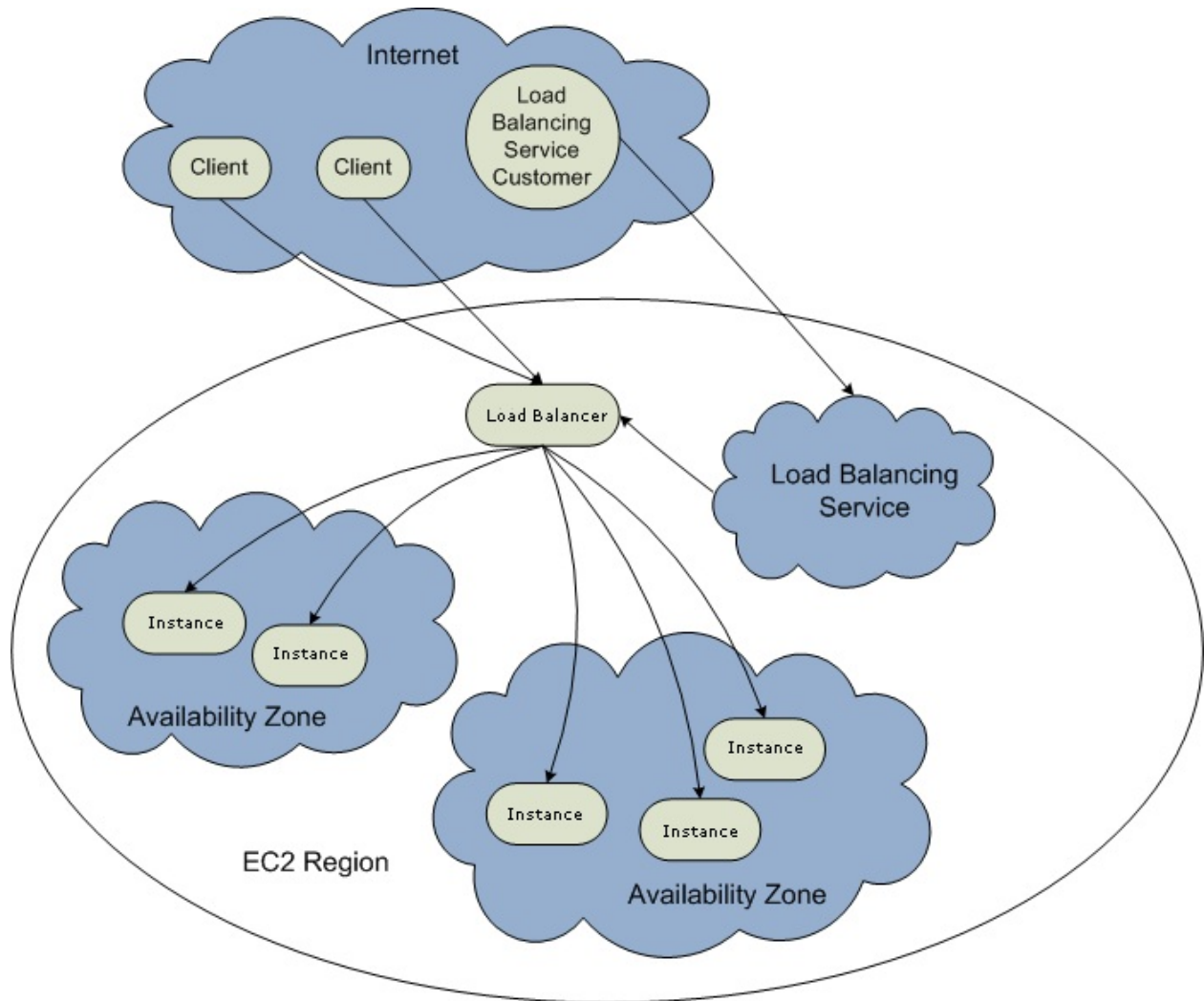
If no endpoint is explicitly specified, the US endpoint will be used by default.

When a LoadBalancer is created, it is associated with an auto generated DNS name. You are free to map that to any other domain name (such as www.mywebsite.com) using CNAME or some other technique.

Elastic Load Balancing Conceptual Overview

Elastic Load Balancing lets you automatically distribute and balance the incoming application traffic among all the instances you are running. The service also makes it easy to add new instances when you need to increase the capacity of your application. You can dynamically register or deregister instances from the LoadBalancer as the capacity requirements of your application change with time.

In the following figure, requests enter the LoadBalancer and are routed to instances within the selected Availability Zones. You can make changes to the LoadBalancer configuration through Elastic Load Balancing.



The LoadBalancer is represented by a DNS name and a set of ports. You will still need a CNAME, or equivalent, in order to map the *generated* DNS name to a name that is *meaningful* to you and your customers (e.g. *www.mywebsite.com*). After you have created your LoadBalancer, you will take the information returned by the service and map it to the public facing DNS name that your customers will see.

Once you create your LoadBalancer, you need to register

your instances with it.

The LoadBalancer also monitors the health of your instances registered with your LoadBalancer. When the LoadBalancer detects a problem with an instance, it stops distributing traffic to it. When the instance is healthy again, the LoadBalancer restarts distributing traffic to it. This process allows your application to automatically react to issues that might affect your customers without your having to be involved beyond configuring the healthcheck.

Availability Zones and Regions

A Load Balancer can distribute traffic to instances across all Availability Zones within a region.



Note

Elastic Load Balancing does not distribute traffic across regions.

Incoming traffic is load balanced equally across all Availability Zones enabled for your LoadBalancer, so it is important to have *equivalent* numbers of instances in each zone. For example, if you have 10 instances in AvailabilityZone us-east-1a and 2 in us-east-1b, the traffic will still be equally distributed between the two Availability Zones. As a result, the two instances in us-east-1b will have to serve the same amount of traffic as the 10 instances in us-east-1a. As a best practice, we recommend you keep equivalent or nearly equivalent number of instances in each of your Availability Zones.

We recommend, for critical applications, that you distribute the requests across multiple availability zones.

For more information, please see the [Using Elastic Load Balancing](#) and, [API Reference](#). to use the service to its fullest capability.

Using Elastic Load Balancing

Topics

- [Using the Query API](#)
- [Using the SOAP API](#)
- [User Scenarios](#)

This section provides task-oriented descriptions of how to use Elastic Load Balancing operations. For a complete description of Elastic Load Balancing operations, refer to the [API Reference](#).

Using the Query API

Query requests are HTTP or HTTPS requests that use the HTTP verb GET or POST and a Query parameter named Action or Operation. Action is used throughout this documentation, although Operation is supported for backward compatibility with other AWS Query APIs.

Query Parameters

Each Query request must include some common parameters to handle authentication and selection of an action. For more information, see [Common Request Parameters](#).



Note

Some API operations take lists of parameters. These lists are specified using the following notation: `param.member.n`. Values of `n` are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:

```
&attribute.member.1=this  
&attribute.member.2=that
```


The Request ID

In every response from Amazon Web Services (AWS), you will find `ResponseMetadata`, which contains a string element called `RequestId`. This is simply a unique identifier AWS assigns to this request for tracking and troubleshooting purposes.

To improve readability of the API documentation and reduce redundancy, `RequestId` is not listed on the individual API documentation pages.

Query API Authentication

You can send Query requests over either HTTP or HTTPS. Regardless of which protocol you use, you must include a signature in every Query request. This section describes how to create the signature. The method described in the following procedure is known as *signature version 2*.

To create the signature

1. Create the canonicalized query string that you need later in this procedure:
 - a. Sort the UTF-8 query string components by parameter name with natural byte ordering.

The parameters can come from the GET URI or from the POST body (when Content-Type is `application/x-www-form-urlencoded`).

- b. URL encode the parameter name and values according to the following rules:
 - Do not URL encode any of the unreserved characters that RFC 3986 defines.

These unreserved characters are A-Z, a-z, 0-9, hyphen (-), underscore (_), period (.), and tilde (~).

- Percent encode all other characters with %XY, where X and Y are hex characters 0-9 and uppercase A-F.
- Percent encode extended UTF-8 characters in the form %XY%ZA....
- Percent encode the space character as %20 (and not +, as common encoding schemes do).



Note

Currently all AWS service parameter names use unreserved characters, so you don't need to encode them. However, you might want to include code to handle parameter names that use reserved characters, for possible future use.

- c. Separate the encoded parameter names from their encoded values with the equals sign (=) (ASCII character 61), even if the parameter value is empty.
- d. Separate the name-value pairs with an ampersand (&) (ASCII code 38).

2. Create the string to sign according to the following

pseudo-grammar (the "\n" represents an ASCII newline).

```
StringToSign = HTTPVerb + "\n" +  
ValueOfHostHeaderInLowercase + "\n" +  
HTTPRequestURI + "\n" +  
CanonicalizedQueryString <from the preceding step>
```

The HTTPRequestURI component is the HTTP absolute path component of the URI up to, but not including, the query string. If the HTTPRequestURI is empty, use a forward slash (/).

3. Calculate an RFC 2104-compliant HMAC with the string you just created, your **Secret Access Key** as the key, and SHA256 or SHA1 as the hash algorithm.

For more information, go to <http://www.ietf.org/rfc/rfc2104.txt>.

4. Convert the resulting value to base64.
5. Use the resulting value as the value of the *Signature* request parameter.



Important

The final signature you send in the request must be URL encoded as specified in RFC 3986 (for more information, go to <http://www.ietf.org/rfc/rfc3986.txt>). If your toolkit

URL encodes your final request, then it handles the required URL encoding of the signature. If your toolkit doesn't URL encode the final request, then make sure to URL encode the signature before you include it in the request. Most importantly, make sure the signature is URL encoded *only once*. A common mistake is to URL encode it manually during signature formation, and then again when the toolkit URL encodes the entire request.

Query Example

Example EnableAvailabilityZoneForLoadBalancer API Request

This example uses the Elastic Load Balancing API [EnableAvailabilityZonesForLoadBalancer](#).

```
https://elasticloadbalancing.amazonaws.com/?AvailabilityZonesForLoadBalancer
&LoadBalancerName=ReferenceLB1
&Action=EnableAvailabilityZonesForLoadBalancer
&Version=2009-05-15
&AWSAccessKeyId=<Your AWS Access Key ID>
&SignatureVersion=2
&SignatureMethod=HmacSHA1
&Timestamp=2009-02-17T05%3A13%3A00.000Z
```

Following is the string to sign.

```
GET\n
elasticloadbalancing.amazonaws.com\n
/>\n
AWSAccessKeyId=<Your AWS Access Key ID>
&Action=EnableAvailabilityZonesForLoadBalancer
&AvailabilityZones.member.1=us-east-1c
&LoadBalancerName=ReferenceLB1
&SignatureMethod=HmacSHA1
&SignatureVersion=2
&Timestamp=2009-02-17T05%3A13%3A00.000Z
&Version=2009-05-15
```

Following is the signed request.

```
https://elasticloadbalancing.amazonaws.com/?Action=Ena
&AvailabilityZones.member.1=us-east-1c
&AWSAccessKeyId=<Your AWS Access Key ID>
&LoadBalancerName=ReferenceLB1
&SignatureVersion=2
&SignatureMethod=HmacSHA1
&Timestamp=2009-10-17T05%3A13%3A00.000Z
&Signature=<URLEncode(Base64Encode(Signature))>
&Version=2009-05-15
```

Using the SOAP API

Topics

-

WSDL and Schema Definitions

- Programming Language Support
- Request Authentication
- The Response Structure
- Web Services References

WSDL and Schema Definitions

You can access the Elastic Load Balancing web service using the SOAP web services messaging protocol. This interface is described by a Web Services Description Language (WSDL) document, which defines the operations and security model for the particular service. The WSDL references an XML Schema document, which strictly defines the data types that might appear in SOAP requests and responses. For more information on WSDL and SOAP, see [Web Services References](#).



Note

Elastic Load Balancing supports SOAP only through HTTPS.

All schemas have a version number. The version number appears in the URL of a schema file and in a schema's target namespace. This makes upgrading easy by differentiating requests based on the version number.

Programming Language Support

Since the SOAP requests and responses in Elastic Load Balancing follow current standards, any programming language with the appropriate library support can be used. Languages known to have this support include C++, C#, Java, Perl, Python and Ruby.

Request Authentication

Elastic Load Balancing complies with the current WS-Security standard, which requires you to hash and sign SOAP requests for integrity and non-repudiation. WS-Security defines profiles which are used to implement various levels of security. Secure SOAP messages use the BinarySecurityToken profile, consisting of an X.509 certificate with an RSA public key.

The following is the content of an insecure RunInstances operation (using EC2 as an example):

```
<RunInstances xmlns="http://ec2.amazonaws.com/doc/2009-03-31/">
  <instancesSet>
    <item>
      <imageId>ami-60a54009</imageId>
      <minCount>1</minCount>
      <maxCount>3</maxCount>
    </item>
  </instancesSet>
  <groupSet/>
</RunInstances>
```

To secure the request, we add the BinarySecurityToken element.

The secure version of the request begins with the

following:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/soap-environment#Security"
      <wsse:BinarySecurityToken
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/soap-environment#Signature"
        EncodingType="http://docs.oasis-open.org/wss/2004/01/soap-environment#Base64Binary"
        ValueType="http://docs.oasis-open.org/wss/2004/01/soap-environment#X509Certificate"
        wsu:Id="CertId-1064304">...many, many lines of
        X.509 certificate...</wsse:BinarySecurityToken>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc17#Canonicalization" />
          <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha1" />
          <ds:Reference URI="#id-17984263">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc17#Canonicalization" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha1" />
            <ds:DigestValue>0pjZ1+TvgPf6uG7o+Yp3l2YdGz...</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#id-15778003">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc17#Canonicalization" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha1" />
            <ds:DigestValue>HhRbxBBmc200348f8nLNZyo4AC...</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>bmVx24Qom4kd9QQ0tclxWIlgLk4C...
        <ds:KeyInfo Id="KeyId-17007273">
          <wsse:SecurityTokenReference
            xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/soap-environment#Signature"
            <wsse:Reference URI="#CertId-1064304">
```

```
ValueType="http://docs.oas
    </wsse:Reference>
    </wsse:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
<wsu:Timestamp
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/08-07/soap-environment#wsu"
  <wsu:Created>2006-06-09T10:57:35Z</wsu:Created>
  <wsu:Expires>2006-06-09T11:02:35Z</wsu:Expires>
</wsu:Timestamp>
</wsse:Security>
</SOAP-ENV:Header>
```

If you are matching this against requests generated by Elastic Load Balancing supplied libraries, or those of another vendor, the following are the most important elements.

Elements

- **BinarySecurityToken**—Contains the X.509 certificate in base64 encoded PEM format
- **Signature**—Contains an XML digital signature created using the canonicalization, signature algorithm, and digest method
- **Timestamp**—Requests to Elastic Load Balancing are valid within 5 minutes of this value to help prevent replay attacks

The Response Structure

In response to a request, the Elastic Load Balancing service returns an XML data structure that conforms to an XML schema defined as part of the Elastic Load Balancing WSDL. The structure of a XML response is specific to the associated request.

The following is an example response (using EC2 as an example):

```
<RuninstancesResponse xmlns="http://ec2.amazon
<reservationId>r-47a5402e</reservationId>
<ownerId>UYY3TLBUXIE0N5NQVUUX60MPWBZIQNFM</ownerId>
<groupSet>
  <item>
    <groupId>default</groupId>
  </item>
</groupSet>
<instancesSet>
  <item>
    <InstanceId>i-2ba64342</InstanceId>
    <imageId>ami-60a54009</imageId>
    <InstanceState>
      <code>0</code>
      <name>pending</name>
    </InstanceState>
    <DNSName></DNSName>
  </item>
  <item>
    <InstanceId>i-2bc64242</InstanceId>
    <imageId>ami-60a54009</imageId>
```

```
<InstanceState>
  <code>0</code>
</InstanceState>
<name>pending</name>
</InstanceState>
<DNSName>ec2-67-202-51-176.compute-1.amazonaws.c
</item>
<item>
  <InstanceId>i-2be64332</InstanceId>
  <imageId>ami-60a54009</imageId>
  <InstanceState>
    <code>0</code>
  </InstanceState>
  <name>pending</name>
  </InstanceState>
  <DNSName>ec2-67-202-51-122.compute-1.amazonaws.c
  <keyName>example-key-name</keyName>
  <instanceType>m1.small</instanceType>
  <launchTime>2007-08-07T11:54:42.000Z</launchTime
</item>
</instancesSet>
</RunInstancesResponse>
```

Web Services References

For more information about using web services, go to any of the following resources:

- [Web Service Description Language \(WSDL\)](#)
- [WS-Security BinarySecurityToken Profile](#)

User Scenarios

Topics

-

[How to Set Up an HTTP LoadBalancer](#)

- [How to Set Up an HTTPS LoadBalancer](#)
- [How to Expand a Load Balanced Application to an Additional Availability Zone](#)
- [How to Disable an Availability Zone from a Load Balanced Application](#)
- [How to Tear Down an Existing LoadBalancer](#)

The following discuss some common user scenarios for using the Elastic Load Balancing API. These scenarios demonstrate the API sequences needed to accomplish the given tasks.



Note

The examples in the following sections assume that your instances are in the US region. If your instances are in Europe, you must specify the eu-west1 region by using the `--region eu-west-1` parameter or setting the `EC2_REGION` environment variable.

How to Set Up an HTTP LoadBalancer

In this example, you create a LoadBalancer for an HTTP service. You specify that the LoadBalancer listens on port 80 and distributes traffic to port 8080 on the instances. You also specify that Availability Zone us-east-1a is enabled for your LoadBalancer. Once the LoadBalancer is created, you configure an application health check for the instances, and register the instances with the LoadBalancer.

Preconditions:

- Your AWS account is signed up for Amazon EC2
- In Availability Zone us-east-1a, you have launched the instances you intend to register with your LoadBalancer
- The instances to be registered with your LoadBalancer respond to the target of the health check with an HTTP status code 200

API Example

To set up an HTTP LoadBalancer

1. Call [CreateLoadBalancer](#) with the following parameters:
 - *AvailabilityZones* = us-east-1a
 - *Listeners*
 - *Protocol* = HTTP
 - *InstancePort* = 8080
 - *LoadBalancerPort* = 80
 - *LoadBalancerName* = MyLoadBalancer

The operation returns the DNS name of your LoadBalancer. You can then map that to any other domain name (such as www.mywebsite.com) using a CNAME or some other technique.

2. Call [ConfigureHealthCheck](#) with the following parameters:

- *LoadBalancerName* = MyLoadBalancer
- *Target* = http:8080/ping



Note

Make sure your instances respond to /ping on port 8080 with an HTTP 200 status code.

- *Interval* = 30
 - *Timeout* = 3
 - *HealthyThreshold* = 2
 - *UnhealthyThreshold* = 2
3. Call **RegisterInstancesWithLoadBalancer** with the following parameters:
- *LoadBalancerName* = MyLoadBalancer
 - *Instances* = [i-4f8cf126, i-0bb7ca62]

Command Line Tools Example

To set up an HTTP LoadBalancer

1. Use the **elb-create-lb** command as in the following example.

```
PROMPT> elb-create-lb MyLoadBalancer --headers --
```

Elastic Load Balancing returns the following:

```
DNS-NAME  DNS-NAME  
DNS-NAME  MyLoadBalancer-2111276808.us-east-1.elb.
```

2. Use the **elb-configure-healthcheck** command as in the following example.

```
PROMPT> elb-configure-healthcheck MyLoadBalancer
```

Elastic Load Balancing returns the following:

```
HEALTH-CHECK TARGET INTERVAL TIMEOUT HEALTHY-THRESI  
HEALTH-CHECK  HTTP:8080/ping  30      3      2
```

3. Use the **elb-register-instances-with-lb** command as in the following example.

```
PROMPT> elb-register-instances-with-lb MyLoadBalanc
```

Elastic Load Balancing returns the following:

```
INSTANCE  INSTANCE-ID  
INSTANCE  i-4f8cf126  
INSTANCE  i-0bb7ca62
```

Give Us Your Feedback

Your input is important to us to help make our documentation helpful and easy to use. Please take a minute to give us your feedback on how well we were able to help you use Elastic Load Balancing. Just click this [feedback](#) link. Thank you.

How to Set Up an HTTPS LoadBalancer

In this example, you create a LoadBalancer for an HTTPS service. Currently, Elastic Load Balancing does not have SSL termination capability. For HTTPS traffic, you need to set the LoadBalancer listener protocol to TCP. You specify that the LoadBalancer listens on port 443 and distributes traffic to port 8443 on the instances. You also specify that Availability Zone us-east-1a is enabled for your LoadBalancer. Once the LoadBalancer is created, you configure an application health check for the instances, and register the instances with the LoadBalancer.

Preconditions:

- Your AWS account is signed up for Amazon EC2
- In Availability Zone us-east-1a, you have launched the instances you intend to register with your LoadBalancer
- The instances to be registered with your LoadBalancer accept TCP connections on the port

specified in the target of the health check

API Example

To set up an HTTPS LoadBalancer

1. Call [CreateLoadBalancer](#) with the following parameters:
 - *AvailabilityZones* = us-east-1a
 - *Listeners*
 - *Protocol* = TCP
 - *InstancePort* = 8443
 - *LoadBalancerPort* = 443
 - *LoadBalancerName* = MyLoadBalancer

The operation returns the DNS name of your LoadBalancer. You can then map that to any other domain name (such as www.mywebsite.com) using CNAME or some other technique.

2. Call [ConfigureHealthCheck](#) with the following parameters:

- *LoadBalancerName* = MyLoadBalancer
- *Target* = TCP:8443



Note

Make sure your instances accept TCP connections on port 8443.

- *Interval* = 30
 - *Timeout* = 3
 - *HealthyThreshold* = 2
 - *UnhealthyThreshold* = 2
3. Call **RegisterInstancesWithLoadBalancer** with the following parameters:
- *LoadBalancerName* = MyLoadBalancer
 - *Instances* = [i-4f8cf126, i-0bb7ca62]

Command Line Tools Example

To set up an HTTPS LoadBalancer

1. Use the **elb-create-lb** command as in the following example.

```
PROMPT> elb-create-lb MyLoadBalancer --headers --
```

Elastic Load Balancing returns the following:

```
DNS-NAME  DNS-NAME  
DNS-NAME  MyLoadBalancer-2111276808.us-east-1.elb.
```

2. Use the **elb-configure-healthcheck** command as in the following example.

```
PROMPT> elb-configure-healthcheck MyLoadBalancer
```

Elastic Load Balancing returns the following:

```
HEALTH-CHECK TARGET INTERVAL TIMEOUT HEALTHY-THRESI  
HEALTH-CHECK TCP:8443 30 3 2 2
```

3. Use the **elb-register-instances-with-lb** command as in the following example.

```
PROMPT> elb-register-instances-with-lb MyLoadBalanc
```

Elastic Load Balancing returns the following:

```
INSTANCE  INSTANCE-ID  
INSTANCE  i-4f8cf126  
INSTANCE  i-0bb7ca62
```

Give Us Your Feedback

Your input is important to us to help make our documentation helpful and easy to use. Please take a minute to give us your feedback on how well we were able to help you use Elastic Load Balancing. Just click this [feedback](#) link. Thank you.

How to Expand a Load Balanced Application to an Additional Availability Zone

In this example, you expand your EC2 application to run in an additional Availability Zone (us-east-1b). To do so, you first register the instances in the Availability Zone us-east-1b with the LoadBalancer. You wait for the instances to show up in the OutOfService state for the LoadBalancer. Finally you enable Availability Zone us-east-1b for your LoadBalancer.



Note

It is important to register instances in the new Availability Zone with your LoadBalancer *before* adding the Availability Zone. When you call `EnableAvailabilityZonesForLoadBalancer`, the LoadBalancer begins to route traffic equally amongst all the enabled Availability Zones. If the instances have not been registered, requests going to the new Availability Zone will fail.

Preconditions:

- You have set up an HTTP LoadBalancer in Availability Zone us-east-1a as in [Setting up a Load Balance to Use HTTP](#).

- In Availability Zone us-east-1b, you have launched the instances you intend to register with your LoadBalancer.

API Example

To expand a load balanced application to an additional Availability Zone

1. Call [RegisterInstancesWithLoadBalancer](#) with the following parameters:
 - *LoadBalancerName* = MyLoadBalancer
 - *Instances* = [i-3a8cf324, i-2603ca33]
2. Call [DescribeInstanceHealth](#) with the following parameters.
 - *LoadBalancerName* = MyLoadBalancer
 - *Instances* = i-3a8cf324, i-2603ca33
3. When the instances from the previous step are in the *OutOfService* state, you can proceed to the next step. Call [EnableAvailabilityZonesForLoadBalancer](#).
 - *LoadBalancerName* = MyLoadBalancer
 - *Availability Zones* = us-east-1b

The operation returns the updated list of Availability Zones enabled for your LoadBalancer.

Command Line Tools Example

To expand a load balanced application to an additional Availability Zone

1. Use the **elb-register-instances-with-lb** command as in the following example.

```
PROMPT> elb-register-instances-with-lb MyLoadBalanc
```

Elastic Load Balancing returns the following:

```
INSTANCE  INSTANCE-ID  
INSTANCE  i-3a8cf324  
INSTANCE  i-2603ca33  
INSTANCE  i-4f8cf126  
INSTANCE  i-0bb7ca62
```

2. Use the **elb-describe-instance-health** command as in the following example.

```
PROMPT> elb-describe-instance-health MyLoadBalanc
```

Elastic Load Balancing returns the following:

```
INSTANCE  INSTANCE-ID  STATE
INSTANCE  i-3a8cf324  OutOfService
INSTANCE  i-2603ca33  OutOfService
```

3. Use the **elb-enable-zones-for-lb** command as in the following example.

```
PROMPT>elb-enable-zones-for-lb  MyLoadBalancer  --l
```

Elastic Load Balancing returns the following:

```
AVAILABILITY_ZONES  AVAILABILITY-ZONES
AVAILABILITY_ZONES  us-east-1a, us-east-1b
```

Give Us Your Feedback

Your input is important to us to help make our documentation helpful and easy to use. Please take a minute to give us your feedback on how well we were able to help you use Elastic Load Balancing. Just click this [feedback](#) link. Thank you.

How to Disable an Availability Zone from a Load Balanced Application

In this example, you disable the Availability Zone us-east-1a for your EC2 application.

This scenario assumes that you have an HTTP LoadBalancer enabled in Availability Zones us-east-1a and us-east-1b.

You disable the Availability Zone for the LoadBalancer first, then give the instances time to go into the OutOfService state before deregistering them from your LoadBalancer.



Note

Your LoadBalancer always distributes traffic to all the enabled Availability Zones. If all the instances in an Availability Zone are deregistered or unhealthy before that Availability Zone is disabled for the LoadBalancer, all requests sent to that Availability Zone will fail until `DisableAvailabilityZonesForLoadBalancer` calls for that Availability Zone.

API Example

To disable an availability zone from a Load Balanced Application

1. Call [DisableAvailabilityZonesForLoadBalancer](#) with the following parameters:
 - *LoadBalancerName* = MyLoadBalancer
 - *Availability Zones* = us-east-1a

The operation returns the updated list of Availability Zones enabled for your LoadBalancer.

2. Call [DescribeInstanceHealth](#) with the following parameters. You have to wait until all of the instances in the disabled Availability Zones are in the *OutOfService* state.
 - *LoadBalancerName* = MyLoadBalancer
 - *Instances* = i-4f8cf126, i-0bb7ca62
3. Call [DeregisterInstancesFromLoadBalancer](#) with the following parameters:

- *LoadBalancerName* = MyLoadBalancer
- Instances = i-4f8cf126, i-0bb7ca62

Command Line Tools Example

To disable an availability zone from a Load Balanced Application

1. Use the **elb-disable-zones-for-lb** command as in the following example.

```
PROMPT> elb-disable-zones-for-lb MyLoadBalancer
```

Elastic Load Balancing returns the following:

```
AVAILABILITY_ZONES  AVAILABILITY-ZONES  
AVAILABILITY_ZONES  us-east-1b
```

2. Use the **elb-describe-instance-health** command as in the following example.

```
PROMPT> elb-describe-instance-health MyLoadBalancer
```

Elastic Load Balancing returns the following:

```
INSTANCE  INSTANCE-ID  STATE
```

```
INSTANCE i-4f8cf126 OutOfService
INSTANCE i-0bb7ca62 OutOfService
```



Note

Only when the instances are in the OutOfService state can you progress to the next step.

3. Use the **elb-deregister-instances-from-lb** command as in the following example.

```
PROMPT> elb-deregister-instances-from-lb MyLoadBa
```

Elastic Load Balancing returns the following:

```
INSTANCE  INSTANCE-ID
INSTANCE  i-3a8cf324
INSTANCE  i-2603ca33
```

Give Us Your Feedback

Your input is important to us to help make our documentation helpful and easy to use. Please take a minute to give us your feedback on how well we were able to help you use Elastic Load Balancing. Just click this [feedback](#) link. Thank you.

How to Tear Down an Existing LoadBalancer

In this example, you stop using Elastic Load Balancing on a currently load balanced EC2 fleet. You delete the LoadBalancer, which automatically deregisters the associated instances from the LoadBalancer.

API Example

To tear down an existing LoadBalancer

- Call [DeleteLoadBalancer](#) with *LoadBalancerName* = MyLoadBalancer.

The operation returns an empty response.

Command Line Tools Example

To tear down an existing LoadBalancer

- Use the **elb-delete-lb** command as in the following example.

```
PROMPT> elb-delete-lb MyLoadBalancer
```

Elastic Load Balancing returns the following:

```
Warning: Deleting a LoadBalancer can lead to servi
```

Enter Y to delete the Load Balancer

Elastic Load Balancing returns the following:

```
OK-Deleting LoadBalancer
```

Give Us Your Feedback

Your input is important to us to help make our documentation helpful and easy to use. Please take a minute to give us your feedback on how well we were able to help you use Elastic Load Balancing. Just click this [feedback](#) link. Thank you.

API Reference

Topics

- [WSDL Locations](#)
- [API Conventions](#)
- [Common Request Parameters](#)
- [Error Codes](#)
- [Elastic Load Balancing Actions](#)
- [Data Types](#)

The following sections of the guide provide reference material for Elastic Load Balancing. For more information about any concepts or programming tasks associated with the reference material, refer to the previous chapters in this guide.

WSDL Locations

The WSDL for Elastic Load Balancing is located at <https://elasticloadbalancing.amazonaws.com/doc/2009-05-15/ElasticLoadBalancing.wsdl>

API Conventions

Data Types and Elastic Load Balancing WSDL

Values provided as parameters to the various operations must be of the indicated type. Standard XSD types (like string, boolean, int) are prefixed with `xsd:`. Complex types defined by the Elastic Load Balancing Service WSDL are prefixed. Please look at the individual WSDL for guidance.



Note

The following example uses an EC2 element, not an element from this service. This section will be expanded later into the beta. However, the information here using this example is still valid.

For example, the `<imagesSet>` element in the following XML snippet is of type `xsd:string[]`.

```
<imagesSet>
  <item>
    <imageId>ami-61a54008</imageId>
  </item>
  <item>
    <imageId>ami-61b54608</imageId>
  </item>
</imagesSet>
```

The `<instancesSet>` element in the following XML

snippet is of type `xsd:string[]`.

```
<instancesSet>
  <item>
    <imageId>ami-60a54009</imageId>
    <minCount>10</minCount>
    <maxCount>30</maxCount>
  </item>
  <item>
    <imageId>ami-60b54209</imageId>
    <minCount>5</minCount>
    <maxCount>20</maxCount>
  </item>
</instancesSet>
```

Common Request Parameters

All Query operations share a set of common parameters that must be present in each call:

Name	Description	Required
<i>Action</i>	Indicates the action to perform. Example: CreateLoadBalancer	Yes
<i>AWSSessionToken</i>	The Access Key ID for the request sender. This identifies the account which will be charged for usage of the service. The account with which the Access Key ID is associated must be signed up for Amazon EC2, or requests will not be accepted. Example: 10QMXFEV71ZS32XQFTR2	Yes
<i>Expires</i>	The date and time at which the signature included in the request expires, in the format YYYY-MM-DDThh:mm:ssZ. Condition: Either <i>Expires</i> or <i>Timestamp</i> must be present in the request, but not both. Example: 2009-01-07T15:04:56Z	Conditional
<i>Timestamp</i>	The date and time at which the request is signed, in the format YYYY-MM-DDThh:mm:ssZ. For more information, go to ISO 8601 . Condition: Either <i>Timestamp</i> or <i>Expires</i> must be present in the request, but not both. Example: 2009-01-07T15:04:56Z	Conditional
<i>Signature</i>	The request signature. For more information, see Query API Authentication . Example: Qnp14Qk/7tINHzfXCiT7VbBatDA=	Yes
<i>SignatureMethod</i>	The hash algorithm you use to create the request signature. Valid values: HmacSHA256 HmacSHA1. For more information, see Query API Authentication . Example: HmacSHA256	Yes

<i>SignatureVersion</i>	The signature version you use to sign the request. Set this value to 2. For more information, see Query API Authentication . Example: 2	Yes
<i>Version</i>	The API version to use, as specified in the WSDL. Example: 2009-05-15	Yes

Parameter values must be URL-encoded. This is true for any Query parameter passed to Elastic Load Balancing; and is typically necessary in the *Signature* parameter. Some clients do this automatically, but this is not the norm.

Error Codes

Topics

- [Overview](#)
- [Elastic Load Balancing Client Error Codes](#)
- [Common Server Error Codes](#)
- [Common Client Error Codes](#)

Overview

There are two types of error codes: client and server.

Client error codes suggest that the error was caused by something the client did, such as an authentication failure or an invalid parameter. In the SOAP API, these error codes are prefixed with `Client.`, such as `Client.LoadBalancerNotFound`. In the Query API, these errors are accompanied by a 400-series HTTP response code.

Server error codes suggest a server-side issue caused the error and should be reported. In the SOAP API, these error codes are prefixed with `Server.` For example: `Server.ServiceUnavailable`. In the Query API, these errors are accompanied by a 500-series HTTP response code.

Elastic Load Balancing Client Error Codes

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
DuplicateLoadBalancerName	LoadBalancer name already exists for this account. Please choose another name.	400 Bad Request	Client
LoadBalancerNotFound	LoadBalancer name does not exist for the account. (For more information, please see		

[LoadBalancer](#).) 400 Bad Request Client
InvalidConfigurationRequest
Requested configuration change is invalid.
409 ConflictClient InvalidInstance Invalid instance.
400 Bad Request Client TooManyLoadBalancers You
have attempted to create more LoadBalancers than
allowed. For more information, please see [LoadBalancer](#).
400 Bad Request Client ValidationError One or more
input parameters are invalid. 400 Bad Request Client

Common Server Error Codes

The following server errors can be returned in addition to the specific errors listed for each service.

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
InternalFailure	Indicates that the request processing has failed due to some unknown error, exception or failure.	500 Internal Failure Error	Server
ServiceUnavailable	Indicates that the request has failed due to a temporary failure of the server.	503 Service Unavailable Error	Server

Common Client Error Codes

The following client errors are common across all the services in this Public Beta and may be returned in addition to the specific errors listed for each service.

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
ElementNotSigned	An element that is marked/configured to be signed is not signed (AWS SOAP only).	403 Forbidden	Client
IncompleteSignature	The request signature does not conform to AWS standards.	400 Bad Request	Client
InvalidAction	The action or operation requested is invalid.	400 Bad Request	Client
InvalidClientTokenId	The X.509 certificate or AWS Access Key Id provided does not exist in our records.	403 Forbidden	Client
InvalidParameterCombination	Parameters that must not be used together were used together.	400 Bad Request	Client
InvalidParameterValue	Bad or out of range value was supplied for the input parameter.	400 Bad Request	Client
InvalidQueryParameter	AWS query string is malformed, does not adhere to AWS standards.	400 Bad Request	Client
InvalidSecurity	The security token provided in the input is missing or expired (AWS SOAP only).	403 Forbidden	Client
InvalidSecurityToken	Incorrect or invalid data is supplied for the security token.	400 Bad Request	Client
InvalidSOAPRequest	The SOAP request is invalid or malformed.	400 Bad Request	Client
MalformedInput	The request is invalid or malformed.	400 Bad Request	Client
MalformedQueryString	The query string is malformed.	400 Bad Request	Client
MalformedSOAPSiganture	The signature element in the SOAP request is malformed.	400 Bad Request	Client

MissingAction	The request is missing an action or operation parameter.	400 Bad Request	Client
MissingAuthenticationToken	Request must contain either a valid (registered) AWS Access Key ID or X.509 certificate.	403 Forbidden	Client
MissingParameter	An input parameter that is mandatory for processing the request is not supplied.	400 Bad Request	Client
OptInRequired	The AWS Access Key Id needs a subscription for the service	403 Forbidden	Client
RequestExpired	Request is past expired date or more than 15 minutes past request date.	400 Bad Request	Client
RequiresSSL	The request must be made over SSL connection.	401 Unauthorized	Client
Throttling	Request was denied due to request throttling.	400 Bad Request	Client
WSecurityMultipleCredentialError	More than one credential was received in the WS-Security header of the request.	400 Bad Request	Client
WSecurityEncodingTypeError	The BinarySecurityToken is not base64 encoded.	400 Bad Request	Client
WSecurityMultipleX.509Error	More than one X.509 binary security token received in WS-Security header of the request.	400 Bad Request	Client
UnknownEnvelopeNamespace	The envelope namespace used to define the SOAP request is invalid.	400 Bad Request	Client

Elastic Load Balancing

Actions

The following actions are available from Elastic Load Balancing.

Creation and Deletion Actions:

- [CreateLoadBalancer](#)
- [DeleteLoadBalancer](#)

Registration Actions:

- [RegisterInstancesWithLoadBalancer](#)
- [DeregisterInstancesFromLoadBalancer](#)

Descriptive Actions (LoadBalancers):

- [DescribeLoadBalancers](#)
- [DescribeInstanceHealth](#)

Availability Zone Actions:

- [EnableAvailabilityZonesForLoadBalancer](#)
- [DisableAvailabilityZonesForLoadBalancer](#)

Healthcheck Actions:

- [ConfigureHealthCheck](#)

ConfigureHealthCheck

Description

This API enables you to define an application healthcheck for the instances.



Note

Completion of this API does not guarantee that operation has completed. Rather, it means that the request has been registered and the changes will happen shortly.

Request Parameters

Name	Description	Required
<i>LoadBalancerName</i>	<p>The mnemonic name associated with the LoadBalancer. The name must be unique within your AWS account.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last character. Maximum length of name is 32 characters (characters are 16-bit Unicode).</p>	Yes
<i>HealthCheck</i>	<p>Structure containing the configuration information for the new healthcheck.</p> <p>Type:</p>	

HealthCheck

Yes

Response Elements

Name	Description
<i>HealthCheck</i>	Updated healthcheck for the instances. Type: HealthCheck Availability: Always returned

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
LoadBalancerNotFound	LoadBalancer name does not exist for the account in this region. (For more information, please see LoadBalancer .)	400 Bad Request	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com/  
?SignatureVersion=2  
&Action=ConfigureHealthCheck  
&Version=2009-05-15  
&HealthCheck.Timeout=2  
&HealthCheck.Target=HTTP%3A80%2Fservlets-examples%2Fse  
&HealthCheck.Interval=5  
&HealthCheck.UnhealthyThreshold=2  
&HealthCheck.HealthyThreshold=2  
&LoadBalancerName=manual010AP  
&Timestamp=2009-02-09T22%3A26%3A28.000Z  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&Signature=%3DXXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<ConfigureHealthCheckResult>  
  <HealthCheck>  
    <Interval>5</Interval>  
    <Target>HTTP:80/servlets-examples/serv  
    <HealthyThreshold>2</HealthyThreshold>  
    <Timeout>2</Timeout>
```

```
        <UnhealthyThreshold>2</UnhealthyThresh  
    </HealthCheck>  
</ConfigureHealthCheckResult>
```

Related Operations

The following operations also return information on the LoadBalancer and/or its instances.

- [DescribeLoadBalancers](#)
- [DescribeInstanceHealth](#)

CreateLoadBalancer

Description

This API creates a new LoadBalancer.



Note

Currently, your quota of LoadBalancers is limited to five per region.

Once the call has completed successfully, a new LoadBalancer will be created, but it will *not* be usable until at least one instance has been registered.

When the LoadBalancer creation is completed, you can check whether it is usable by using the

[DescribeInstanceHealth](#) API. The LoadBalancer is usable as soon as any registered instance is *InService*.



Note


Load balancer DNS names will vary depend on the region in which they were created. For load balancers created in the United States, the DNS name will end with:

- us-east-1.elb.amazonaws.com

For load balancers created in Europe, the DNS name will end with:

- eu-west-1.elb.amazonaws.com

Request Parameters

Name	Description	Required
<i>AvailabilityZones</i>	<p>List of Availability Zones. This list can be modified after the creation of the LoadBalancer. The Availability Zones specified must be in the same EC2 region as the LoadBalancer. You must specify at least one Availability Zone. Traffic will be equally distributed across all zones.</p> <p>Type: AvailabilityZones</p> <p>Default: None</p> <p> Note</p> <p>Some API operations take lists of parameters. These lists are specified using the following notation: param.member.n. Values of n are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre>&attribute.member.1=this &attribute.member.2=that</pre> </div>	Yes
<i>LoadBalancerName</i>	<p>The mnemonic name associated with the LoadBalancer. The name must be unique within your set of LoadBalancers requests on the specified protocol and received by Elastic Load Balancing on the LoadBalancerPort are load balanced across the registered instances and sent to port InstancePort.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last character. Maximum length of name is 32 characters (characters are 16-bit Unicode).</p>	Yes
<i>Listeners</i>	<p>This parameter is used to denote a list of the following tuples LoadBalancerPort, InstancePort, and Protocol.</p>	Yes

Type: Listeners



Note

Some API operations take lists of parameters. These lists are specified using the following notation: param.member.n. Values of n are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:

```
&attribute.member.1=this  
&attribute.member.2=that
```

Response Elements

Name	Description
<i>DNSName</i>	DNS name for the LoadBalancer. Type: String Availability: Always returned

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
DuplicateLoadBalancerName	<p>The LoadBalancer name requested already exists in this AWS account.</p> <p>This error is returned if you try creating a LoadBalancer with the same name as a current LoadBalancer that you have, but with a different set of parameters. If you try creating a LoadBalancer with the same name <i>and parameters</i> of a LoadBalancer you already have, then the system will simply return the same output as it did when you created the LoadBalancer the first time. In other words, in such a case, you won't get an error, but you simply will get the same information back (but still only one LoadBalancer of that name and with that set of parameters.)</p>	400 Bad Request	Client
TooManyLoadBalancers	Quota of LoadBalancers has already been reached.	400 Bad Request	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /?Signature
&Action=CreateLoadBalancer
&Version=2009-05-15
&LoadBalancerName=TestLoadBalancer
&Listeners.member.1.Protocol=HTTP
&Listeners.member.1.LoadBalancerPort=80
&Listeners.member.1.InstancePort=80
&AvailabilityZones.member.1=us-east-1b
&Timestamp=2009-04-03T18%3A20%3A54.000Z
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<CreateLoadBalancerResult>
<DNSName> TestLoadBalancer-380544827.us-east-1.elb.ama
</CreateLoadBalancerResult>
```

Related Operations

- [DeleteLoadBalancer](#)
- [RegisterInstancesWithLoadBalancer](#)
- [DeregisterInstancesFromLoadBalancer](#)
- [EnableAvailabilityZonesForLoadBalancer](#)
- [DisableAvailabilityZonesForLoadBalancer](#)

DeleteLoadBalancer

Description

This API deletes the specified LoadBalancer. On deletion, all of the configured properties of the LoadBalancer will be deleted. If you attempt to recreate the LoadBalancer, you need to reconfigure all the settings. The DNS name associated with a deleted LoadBalancer is no longer be usable. Once deleted, the name and associated DNS record of the LoadBalancer no longer exist and traffic sent to any of its IP addresses will no longer be delivered to your instances. You will not get the same DNS name even if you create a new LoadBalancer with same *LoadBalancerName*.

To successfully call this API, you must provide the same account credentials as those that were used to create the LoadBalancer.



Note

Because this API has been designed to be idempotent, even if the LoadBalancer does not exist or has been deleted, *DeleteLoadBalancer* still returns a success.

Request Parameters

Name	Description	Required
<i>LoadBalancerName</i>	<p>The mnemonic name associated with the LoadBalancer. The name must be unique within your AWS account.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last character. Maximum length of name is 32 characters (characters are 16-bit Unicode).</p>	Yes

Response Elements

None.

Special Errors

Common errors for all calls are listed and described in a separate section, please see

[API Error Codes.](#)

Error	Description	HTTP Status Code	SOAP Prefix
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /  
?LoadBalancerName=ReferanceAP1  
&Action=DeleteLoadBalancer  
&Version=2009-05-15  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&SignatureVersion=2  
&SignatureMethod=HmacSHA1  
&Timestamp=2009-10-17T05%3A47%3A28.000Z  
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<DeleteLoadBalancerResult/>
```

Related Operations

[CreateLoadBalancer](#)

DeregisterInstancesFromL

Description

This API deregisters instances from the LoadBalancer. Trying to deregister an instance that is not registered with the LoadBalancer does nothing.

In order to successfully call this API, you must provide the same account credentials as those that were used to create the LoadBalancer.

Once the instance is deregistered, it will stop receiving traffic from the LoadBalancer.

Request Parameters

Name	Description	Required
<i>Instances</i>	List of <i>EC2 instance</i> IDs consisting of all instances you want to be deregistered. Type:	

Instances

Default: None



Note

Some API operations take lists of parameters. These lists are specified using the following notation: `param.member.n`. Values of `n` are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:

```
&attribute.member.1=this  
&attribute.member.2=that
```

Yes

LoadBalancerName

The mnemonic name associated with the LoadBalancer.

The name must be unique within your AWS account.

Type: String

Default: None

Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last

character. Maximum length of name is 32 characters
(characters are 16-bit Unicode).

Yes

Response Elements

Name	Description
<i>Instances</i>	Updated list of remaining instances registered with the LoadBalancer. Type: Instances

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
LoadBalancerNotFound	LoadBalancer name does not exist for the account in this region. (For more information, please see LoadBalancer .)	400 Bad Request	Client
InvalidInstance	This error is returned if any of the specified instances are invalid.	400 Bad Request	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /  
?LoadBalancerName=ReferenceAPI  
&Instances.member.1.InstanceId=i-6055fa09  
&Action=DeregisterInstancesFromLoadBalancer  
&Version=2009-05-15  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&SignatureVersion=2  
&SignatureMethod=HmacSHA1  
&Timestamp=2009-10-17T05%3A42%3A57.000Z  
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<DeregisterInstancesFromLoadBalancerResult>  
  <Instances/>  
</DeregisterInstancesFromLoadBalancerResult>
```

Related Operations

[RegisterInstancesWithLoadBalancer](#)

DescribeLoadBalancers


Description

This API returns detailed configuration information for the specified LoadBalancers, or if no LoadBalancers are specified, then the API returns configuration information for all LoadBalancers created by the caller. For more information, please see

[LoadBalancer](#).

You must have created the specified input LoadBalancers in order to retrieve this information. In other words, in order to successfully call this API, you must provide the same account credentials as those that were used to create the LoadBalancer.

Request Parameters

Name	Description	Required
<i>LoadBalancerNames</i>	<p>A list of <i>LoadBalancerNames</i>, the mnemonic name that was associated with the LoadBalancer at creation time.</p> <p>Type: LoadBalancerNames</p> <p>Default: If the list is empty, the call will return descriptions for all the LoadBalancers active.</p> <p> Note</p> <p>Some API operations take lists of parameters. These lists are specified using the following notation: param.member.n. Values of n are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:</p> <div data-bbox="651 1215 1235 1367" style="border: 1px solid black; padding: 5px; margin: 10px 0;"><pre>&attribute.member.1=this &attribute.member.2=that</pre></div>	No

Response Elements

Name	Description
<i>LoadBalancerDescriptions</i>	List of LoadBalancer description structures. Type: LoadBalancerDescription

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
LoadBalancerNotFound	LoadBalancer name does not exist for the account in this region. (For more information, please see LoadBalancer .)	400 Bad Request	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /  
?Action=DescribeLoadBalancers  
&Version=2009-05-15  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&SignatureVersion=2  
&SignatureMethod=HmacSHA1  
&Timestamp=2009-10-17T05%3A14%3A21.000Z  
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<DescribeLoadBalancersResult>  
  <LoadBalancersDescriptions>  
    <member>  
      <Listeners>  
        <member>  
          <Protocol>HTTP</Protocol>  
          <LoadBalancerPort>80</LoadBalancerPort>  
          <InstancePort>80</InstancePort>  
        </member>  
      </Listeners>  
    </LoadBalancerName>TestLoadBalancer
```

```
</Instances/>
</CreatedTime>2009-04-03T18:2
<HealthCheck>
  <Interval>30</Interval>
  <Target>TCP:80</Target>
  <HealthyThreshold>10</
  <Timeout>5</Timeout>
  <UnhealthyThreshold>2<
</HealthCheck>
<DNSName>TestLoadBalancer-406
<AvailabilityZones>
  <member>us-east-1b</member>
</AvailabilityZones>
</member>
</LoadBalancersDescriptions>
</DescribeLoadBalancersResult>
```

Related Operations

The following operations also return information on the LoadBalancer and/or its instances.

- [DescribeInstanceHealth](#)

DescribeInstanceHealth

Description

This API returns the current state of the instances of the specified LoadBalancer. If no instances are specified, the state of all the instances for the LoadBalancer is returned.

You must have been the one who created in the LoadBalancer. In other words, in order to successfully call this API, you must provide the same account credentials as those that were used to create the LoadBalancer.

Request Parameters

Name	Description	Required
<i>Instances</i>	List of instances IDs whose state is being queried. Type:	

Instances

Default: If the list is empty, it returns the state of all the instances registered with this LoadBalancer

No

LoadBalancerName

The mnemonic name associated with the LoadBalancer. The name must be unique within your AWS account.

Type: String

Default: None

Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last character. Maximum length of name is 32 characters (characters are 16-bit Unicode).

Yes

Response Elements

Name	Description
<i>InstanceStates</i>	<p>Structure containing instance Health information for the specified instances, such as <i>InService</i> <i>OutOfService</i> state.</p> <p>Type: InstanceStates</p> <p>Availability: Always returned</p> <div data-bbox="428 674 1433 831" style="border: 1px solid black; background-color: #e6f2ff; height: 75px; margin-top: 10px;"></div>

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
LoadBalancerNotFound	LoadBalancer name does not exist for the account. (For more information, please see LoadBalancer .)	400 Bad Request	Client
InvalidInstance	This error is returned if any of the specified instances are invalid. This error is returned if any of the specified instances are not registered with this LoadBalancer.	400 Bad Request	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /  
?SignatureVersion=2  
&Action=DescribeInstanceHealth  
&Version=2009-05-15  
&LoadBalancerName=manual006AP  
&Timestamp=2009-10-08T17%3A58%3A07.000Z  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<DescribeInstanceHealthResult>  
  <InstanceStates>  
    <member>  
      <Description>N/A</Description>  
      <State>InService</State>  
      <InstanceId>i-8e0ea3e7</Instar  
      <ReasonCode>N/A</ReasonCode>  
    </member>  
    <member>
```

```
<Description>Instance registra
<State>OutOfService</State>
<InstanceId>i-6055fa09</Instar
<ReasonCode>ELB</ReasonCode>
</member>
</InstanceStates>
</DescribeInstanceHealthResult>
```

Related Operations

The following operations also return information on the LoadBalancer and/or its instances.

- [DescribeLoadBalancers](#)

DisableAvailabilityZonesFor

Description

This API removes the specified EC2 Availability Zones from the set of configured Availability Zones for the LoadBalancer. Once an Availability Zone is removed, all the instances registered with the LoadBalancer that are in the removed Availability Zone go into the *OutOfService* state. Upon Availability Zone removal, the LoadBalancer attempts to equally balance the traffic among its remaining usable Availability Zones. Trying to remove an Availability Zone that was not associated with the LoadBalancer does nothing.

There must be at least one Availability Zone registered with a LoadBalancer at all times. You cannot remove all the Availability Zones from a LoadBalancer.

In order for this call to be successful, you must have created the LoadBalancer. In other words, in order to successfully call this API, you must provide the same account credentials as those that were used to create the LoadBalancer.

Request Parameters

Name	Description	Required
<i>AvailabilityZones</i>	List of Availability Zones to be removed from the LoadBalancer. There must be at least one Availability Zone registered with a LoadBalancer at all times. You cannot remove all the Availability Zones from a LoadBalancer. Specified Availability Zones must be in the same region. Default: None Type:	

AvailabilityZones



Note

Some API operations take lists of parameters. These lists are specified using the following notation: param.member.n. Values of n are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:

```
&attribute.member.1=this  
&attribute.member.2=that
```

Yes

LoadBalancerName

The mnemonic name associated with the LoadBalancer. The name must be unique within your AWS account.

Type: String

Default: None

Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last

character. Maximum length of name is 32 characters
(characters are 16-bit Unicode).

Yes

Response Elements

Name	Description
<i>AvailabilityZones</i>	List of updated Availability Zones for the LoadBalancer. Type: AvailabilityZones

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
LoadBalancerNotFound	LoadBalancer name does not exist for the account in this region. (For more information, please see LoadBalancer .)	400 Bad Request	Client
InvalidConfigurationRequest	Requested configuration change is invalid. This error is returned if you attempt to remove all the registered Availability Zones for the LoadBalancer.	409 Conflict	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /  
?AvailabilityZones.member.1=us-east-1a  
&LoadBalancerName=ReferenceAP2  
&Action=DisableAvailabilityZonesForLoadBalancer  
&Version=2009-05-15  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&SignatureVersion=2  
&SignatureMethod=HmacSHA1  
&Timestamp=2009-10-17T05%3A13%3A43.000Z  
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<DisableAvailabilityZonesForLoadBalancerResult>  
  <AvailabilityZones>  
    <member>us-east-1c</member>  
  </AvailabilityZones>  
</DisableAvailabilityZonesForLoadBalancerResult>
```

Related Operations

[EnableAvailabilityZonesForLoadBalancer](#)

EnableAvailabilityZonesFc

Description

This API is used to add one or more EC2

Availability Zones to the LoadBalancer.

The LoadBalancer evenly distributes requests across all its registered Availability Zones that contain instances. As a result, you must ensure that your LoadBalancer is appropriately scaled for each registered Availability Zone.



Note

The new EC2 Availability Zones to be added must be in the same EC2 [region](#) as the Availability Zones for which the LoadBalancer was created.

In order to successfully call this API, you must provide the same account credentials as those that were used to create the LoadBalancer.

Request Parameters

Name	Description	Required
<i>AvailabilityZones</i>	<p>List of new Availability Zones for the LoadBalancer. The Availability Zone must be in the same region as the LoadBalancer.</p> <p>Type: AvailabilityZones</p> <p>Default: None</p> <p> Note</p> <p>Some API operations take lists of parameters. These lists are specified using the following notation: param.member.n. Values of n are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"><pre>&attribute.member.1=this &attribute.member.2=that</pre></div>	Yes
<i>LoadBalancerName</i>	<p>The mnemonic name associated with the LoadBalancer. The name must be unique within your AWS account.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last character. Maximum length of name is 32 characters (characters are 16-bit Unicode).</p>	Yes

Response Elements

Name	Description
<i>AvailabilityZones</i>	Updated list of Availability Zones for the LoadBalancer. Type: AvailabilityZones Availability: Always returned

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
LoadBalancerNotFound	LoadBalancer name does not exist for the account in this region. (For more information, please see LoadBalancer .)	400 Bad Request	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /  
?AvailabilityZones.member.1=us-east-1c  
&LoadBalancerName=ReferenceAP1  
&Action=EnableAvailabilityZonesForLoadBalancer  
&Version=2009-05-15  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&SignatureVersion=2  
&SignatureMethod=HmacSHA1  
&Timestamp=2009-10-17T05%3A13%3A00.000Z  
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<EnableAvailabilityZonesForLoadBalancerResult>  
  <AvailabilityZones>  
    <member>us-east-1c</member>  
    <member>us-east-1b</member>  
    <member>us-east-1a</member>  
  </AvailabilityZones>  
</EnableAvailabilityZonesForLoadBalancerResult>
```



Related Operations

- [DisableAvailabilityZonesForLoadBalancer](#)
- [CreateLoadBalancer](#)
- [RegisterInstancesWithLoadBalancer](#)

RegisterInstancesWithLoa

Description

This API adds new instances to the LoadBalancer.

Once the instance is registered, it starts receiving traffic and requests from the LoadBalancer. Any instance that is not in any of the

Availability Zones registered for the LoadBalancer will be moved to the *OutOfService* state. It will move to the *InService* state when the *Availability Zone* is added to the LoadBalancer.


You must have been the one who created the LoadBalancer. In other words, in order to successfully call this API, you must provide the same account credentials as those that were used to create the LoadBalancer.



Note

Completion of this API does not guarantee that operation has completed. Rather, it means that the request has been registered and the changes will happen shortly.

Request Parameters

Name	Description	Required
<i>Instances</i>	<p>List of instances IDs that should be registered with the LoadBalancer.</p> <p>Type: Instances</p> <p>Default: None</p> <p> Note</p> <p>Some API operations take lists of parameters. These lists are specified using the following notation: param.member.n. Values of n are integers starting from 1. All lists of parameters must follow this notation, including lists that only contain one parameter. For example, a Query parameter list looks like this:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"><pre>&attribute.member.1=this &attribute.member.2=that</pre></div>	Yes
<i>LoadBalancerName</i>	<p>The mnemonic name associated with the LoadBalancer. The name must be unique within your AWS account.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must only use alphanumeric characters (case sensitive) and dash (-). Dash cannot be first or last character. Maximum length of name is 32 characters (characters are 16-bit Unicode).</p>	Yes

Response Elements

Name	Description
<i>Instances</i>	Updated list of instances for the LoadBalancer. Type: Instances

Special Errors

Common errors for all calls are listed and described in a separate section, please see [API Error Codes](#).

Error	Description	HTTP Status Code	SOAP Prefix
LoadBalancerNotFound	LoadBalancer name does not exist for the account in this region. (For more information, please see LoadBalancer .)	400 Bad Request	Client
InvalidInstance	This error is returned if any of the specified instances are invalid.	400 Bad Request	Client
ValidationError	One or more input parameters are invalid.	400 Bad Request	Client

Examples

Sample Request

Example Request (Query)

```
https://elasticloadbalancing.amazonaws.com /  
?LoadBalancerName=ReferenceAP1  
&Instances.member.1.InstanceId=i-6055fa09  
&Action=RegisterinstancesWithLoadBalancer  
&Version=2009-05-15  
&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX  
&SignatureVersion=2  
&SignatureMethod=HmacSHA1  
&Timestamp=2009-10-17T05%3A40%3A29.000Z  
&Signature=%XXX YOUR SIGNATURE XXX%3D
```

Sample Response

Example Query Response

```
<RegisterInstancesWithLoadBalancerResult>  
  <Instances>  
    <member>  
      <InstanceId>i-6055fa09  
    </member>  
  </Instances>  
</RegisterInstancesWithLoadBalancerResult>
```



Related Operations

- [DeregisterInstancesFromLoadBalancer](#)
- [CreateLoadBalancer](#)
- [EnableAvailabilityZonesForLoadBalancer](#)

Data Types

Topics

- [AvailabilityZones](#)
- [HealthCheck](#)
- [HealthCheckInterval](#)
- [HealthCheckTimeout](#)
- [HealthyThreshold](#)
- [Instance](#)
- [InstancePort](#)
- [Instances](#)
- [InstanceState](#)
- [InstanceStates](#)
- [Listener](#)
- [Listeners](#)
- [LoadBalancerDescription](#)
- [LoadBalancersDescriptions](#)
- [LoadBalancerNames](#)
- [UnhealthyThreshold](#)

The Load Balancing API contains several data types that various operations use. This section describes each data type in detail.

Since both the Query and SOAP APIs return the same

XML body, the data types described in the WSDL are used in both.

AvailabilityZones

The AvailabilityZones data type. The number of elements is *unbounded*.

Relevant Operations

- [EnableAvailabilityZonesForLoadBalancer](#)
- [DisableAvailabilityZonesForLoadBalancer](#)
- [DescribeLoadBalancers](#)

Contents

The following table describes the elements contained in AvailabilityZones.

Name	Description	Required
member	An Availability Zone . Type: string	Yes

HealthCheck

The HealthCheck data type.

Relevant Operations

- [ConfigureHealthCheck](#)
- [DescribeLoadBalancers](#)

Contents

The following table describes the elements contained in HealthCheck.

Name	Description	Required
HealthyThreshold	<p>The number of consecutive health probe successes required before moving the instance to the Healthy state. The default is three and a valid value lies between two and ten.</p> <p>Type: HealthyThreshold</p>	Yes
Interval	<p>The approximate interval (in seconds) between health checks of an individual instance. The default is 30 seconds and a valid interval must be between 5 seconds and 600 seconds. Also, the interval value must be greater than the Timeout value</p> <p>Type: HealthCheckInterval</p>	Yes
Target	<p>The instance being checked. The protocol is either TCP or HTTP. The range of valid ports is one (1) through 65535.</p> <p>Notes: TCP is the default, specified as a TCP: port pair, for example "TCP:5000". In this case a healthcheck simply attempts to open a TCP connection to the instance on the specified port. Failure to connect within the configured timeout is considered <i>unhealthy</i>.</p> <p>For HTTP, the situation is different. HTTP is specified as a HTTP:port;/;PathToPing; grouping, for example "HTTP:80/weather/us/wa/seattle". In this case, a HTTP GET request is issued to the instance on the given port and path. Any answer other than "200 OK" within the timeout period is considered <i>unhealthy</i>.</p> <p>The total length of the HTTP ping target needs to be 1024 16-bit Unicode characters or less.</p> <p>Type: string</p>	Yes
Timeout	<p>Amount of time (in seconds) during which no response means a failed health probe. The default is five seconds and a valid value must be between 2 seconds and 60 seconds. Also, the timeout value must be less than the Interval value.</p> <p>Type: HealthCheckTimeout</p>	Yes
UnhealthyThreshold	<p>The number of consecutive health probe failures that move the instance to the unhealthy state. The default is 5 and a valid value lies between 2</p>	Yes

and 10.

Type: [UnhealthyThreshold](#)

HealthCheckInterval

The HealthCheckInterval data type.

Relevant Operations

- [ConfigureHealthCheck](#)
- [DescribeLoadBalancers](#)

Name	Description	Required
HealthCheckInterval	The HealthCheckInterval data type is the approximate interval between health checks of an individual instance. The interval is specified in seconds. The default is 30 seconds and a valid interval must be between 5 seconds and 600 seconds. Also, the interval value must be greater than the Timeout value of healthcheck.	Yes

HealthCheckTimeout

The HealthCheckTimeout data type.

Relevant Operations

- [ConfigureHealthCheck](#)
- [DescribeLoadBalancers](#)

Name	Description	Required
HealthCheckTimeout	The HealthCheckTimeout data type is an integer that represents the amount of time, in seconds, during which no response means a failed health probe. The default is five seconds and a valid value must be between 2 seconds and 60 seconds. Also, the timeout value must be less than the Interval value of HealthCheck .	Yes

HealthyThreshold

The HealthyThreshold data type.

Relevant Operations

- [ConfigureHealthCheck](#)
- [DescribeLoadBalancers](#)

Name	Description	Required
HealthyThreshold	The HealthyThreshold data type is a simple type of type: integer. It is the number of consecutive health probe successes before moving the instance back to the Healthy state. The default is three (3) and a valid value lies between two (2) and ten (10).	Yes

Instance

The Instance data type.

Relevant Operations

- [DescribeLoadBalancers](#)
- [DescribeInstanceHealth](#)
- [RegisterInstancesWithLoadBalancer](#)
- [DeregisterInstancesFromLoadBalancer](#)

Contents

The following table describes the elements contained in instance.

Name	Description	Required
InstanceId	An <i>EC2 instance</i> ID. Type: string	Yes

InstancePort

The InstancePort data type.

Relevant Operations

- [CreateLoadBalancer](#)
- [DescribeLoadBalancers](#)

Name	Description	Required
InstancePort	The InstancePort data type. It is the TCP port on which the server on the instance is listening. Valid instance ports are one (1) through 65535. This property cannot be modified for the life of the LoadBalancer.	Yes

Instances

The Instances data type. This type is *unbounded*.

Relevant Operations

- [DescribeLoadBalancers](#)
- [DescribeInstanceHealth](#)
- [RegisterInstancesWithLoadBalancer](#)
- [DeregisterInstancesFromLoadBalancer](#)

Contents

The following table describes the elements contained in instances.

Name	Description	Required
member	An instance. Type: Instance	Yes

InstanceState

The InstanceState data type.

Relevant Operations

- [DescribeInstanceHealth](#)

Contents

The following table describes the elements contained in `instancestate`.

Name	Description
<code>InstanceId</code>	<p><i>EC2 instance</i> ID of the instance.</p> <p>Type: string</p>
<code>State</code>	<p>Current status of the instance:</p> <ul style="list-style-type: none">• The state can be <i>InService</i>, when traffic is being routed to the instance by the LoadBalancer• The state can be <i>OutOfService</i>, when traffic is not being routed to the instance by the LoadBalancer <p>Type: string</p>
<code>ReasonCode</code>	<p><code>ReasonCode</code> returns a string that provides more information about the cause of <i>OutOfService</i> instances. Specifically, it indicates whether the cause is Elastic Load Balancing or the instance behind the Load Balancer.</p> <p>See the <code>Description</code> element of this data type.</p> <p>Type: string</p>
<code>Description</code>	<p>Possible reasons for an <i>OutOfService</i> state include the following:</p> <ul style="list-style-type: none">• Health checks to the instance are failing, or at a minimum have consecutively failed the UnhealthyThreshold number of health checks, and therefore is not considered healthy• The registered instance has not consecutively passed the configured HealthyThreshold number of health checks, and therefore is still not yet considered healthy• The LoadBalancer (of the given LoadBalancer name) is not available• The instance has terminated• The instance is in an EC2 Availability Zone that the LoadBalancer has not been configured to route traffic to <p>For more information about health checks, please see the HealthCheck data type and the API, ConfigureHealthCheck.</p> <p>Type: string</p>

InstanceStates

The InstanceStates data type. This type is *unbounded*.

Relevant Operations

- [DescribeInstanceHealth](#)

Contents

The following table describes the elements contained in instancestates.

Name	Description	Required
member	instance state information. Type: InstanceState	Yes

Listener

The Listener data type.

Relevant Operations

- [CreateLoadBalancer](#)
- [DescribeLoadBalancers](#)

Contents

The following table describes the elements contained in Listener.

Name	Description	Required
LoadBalancerPort	<p>The external TCP port of the LoadBalancer. Valid LoadBalancer ports are - 80, 443 and 1024 through 65535. This property cannot be modified for the life of the LoadBalancer.</p> <p>Type: integer</p>	Yes
InstancePort	<p>The InstancePort data type is simple type of type: integer. It is the TCP port on which the server on the instance is listening. Valid instance ports are one (1) through 65535. This property cannot be modified for the life of the LoadBalancer.</p> <p>Type: integer</p>	Yes
Protocol	<p>LoadBalancer transport protocol to use for routing - TCP or HTTP. This property cannot be modified for the life of the LoadBalancer.</p> <p>Type: string</p>	Yes

Listeners

The Listeners data type.

Relevant Operations

- [CreateLoadBalancer](#)
- [DescribeLoadBalancers](#)

Contents

The following table describes the elements contained in Listeners.

Name	Description	Required
member	A Listener description structure. Type: Listener	Yes

LoadBalancerDescription

The LoadBalancerDescription data type. This structure contains the configuration details of a LoadBalancer that gets returned in the [DescribeLoadBalancers](#) API.

Relevant Operations

- [DescribeLoadBalancers](#)

Contents

The following table describes the elements contained in LoadBalancerDescription.

Name	Description	Required
LoadBalancerName	The mnemonic name associated with the LoadBalancer. Type: String	Yes
AvailabilityZones	List of Availability Zones . Type: AvailabilityZones	Yes
CreatedTime	Time that the LoadBalancer was created. Type: dateTime	Yes
DNSName	External DNS name associated with the LoadBalancer. Type: String	Yes
Instances	List of instance IDs. <i>EC2 instances</i> comprise the instances for the LoadBalancer. Type: Instances	Yes
HealthCheck	Structure containing the various health probes conducted on the LoadBalancer. Type: HealthCheck	Yes
Listeners	LoadBalancerPort, InstancePort, and Protocol are returned in a list of tuples in the Listeners element. Type: Listeners	Yes

LoadBalancersDescription

The LoadBalancersDescriptions data type. The type is *unbounded*.

Relevant Operations

- [DescribeLoadBalancers](#)

Contents

The following table describes the elements contained in LoadBalancersDescriptions.

Name	Description	Required
member	a LoadBalancer description structure. Type: LoadBalancerDescription	Yes

LoadBalancerNames

The LoadBalancerNames data type. The type is *unbounded*.

Relevant Operations

- [DescribeLoadBalancers](#)

Contents

The following table describes the elements contained in LoadBalancerNames.

Name	Description	Required
member	a LoadBalancer by name. Type: string	Yes

UnhealthyThreshold

The UnhealthyThreshold data type.

Relevant Operations

- [ConfigureHealthCheck](#)
- [DescribeLoadBalancers](#)

Name	Description	Required
UnhealthyThreshold	The UnhealthyThreshold data type is a simple type of type: integer. It is the number of consecutive health probe failures that move the instance to the Unhealthy state. The default is five (5) and a valid value lies between two (2) and ten (10).	Yes

Glossary

Access Key ID

An alphanumeric token that uniquely identifies a request sender. This ID is associated with your Secret Access Key.

Amazon Machine Image

An Amazon Machine Image (AMI) is an encrypted machine image stored in Amazon Simple Storage Service (Amazon S3). It contains all the information necessary to boot instances of your software.

Availability Zone

Amazon EC2 locations are composed of regions and Availability Zones. Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same region.

LoadBalancer

Elastic Load Balancing key term. Please see the detailed information on this term, located at key term LoadBalancer.

region

Amazon EC2 locations are composed of regions and Availability Zones. Regions are geographically dispersed and will be in separate geographic areas or countries. Regions consist of one or more Availability Zones.

Secret Access Key

A key assigned to you by Amazon Web Services (AWS) when you sign up for an AWS account. Used for request authentication.

unbounded

Term used in Web Service Definition Language (WSDL), e.g. `maxOccurs="unbounded"`, meaning that the number of potential occurrences is not limited by a set number. Very often used when defining a data type that is a list of other types, such as an unbounded list of integers (element members) or an unbounded list of other complex types that are element/members of the list being defined.

Document Conventions

This section lists the common typographical and symbol use conventions for AWS technical publications.

Typographical Conventions

This section describes common typographical use conventions.

Convention	Description/Example
Call-outs	<p>A call-out is a number in the body text to give you a visual reference. The reference point is for further discussion elsewhere.</p> <p>You can use this resource regularly. 1</p>
Code in text	<p>Inline code samples (including XML) and commands are identified with a special font.</p> <p>You can use the command <code>java -version</code>.</p>
Code blocks	<p>Blocks of sample code are set apart from the body and marked accordingly.</p> <pre># ls -l /var/www/html/index.html -rw-rw-r-- 1 root root 1872 Jun 21 09:33 /var/www/html/index.htm # date Wed Jun 21 09:33:42 EDT 2006</pre>
Emphasis	<p>Unusual or important words and phrases are marked with a special font.</p> <p>You <i>must</i> sign up for an account before you can use the service.</p>
Internal cross references	<p>References to a section in the same document are marked.</p>
Logical values, constants, and regular expressions, abstracta	<p>A special font is used for expressions that are important to identify, but are not code.</p> <p>If the value is <code>null</code>, the returned response will be <code>false</code>.</p>
Product and feature names	<p>Named AWS products and features are identified on first use.</p> <p>Create an Amazon Machine Image (AMI).</p>
Operations	<p>In-text references to operations.</p> <p>Use the <code>GetHitResponse</code> operation.</p>
Parameters	<p>In-text references to parameters.</p>

	The operation accepts the parameter <i>AccountID</i> .
Response elements	In-text references to responses. A container for one <code>CollectionParent</code> and one or more <code>CollectionItems</code> .
Technical publication references	References to other AWS publications. If the reference is hyperlinked, it is also underscored. For detailed conceptual information, see the <i>Amazon Mechanical Turk Developer Guide</i> .
User entered values	A special font marks text that the user types. At the password prompt, type MyPassword .
User interface controls and labels	Denotes named items on the UI for easy identification. On the File menu, click Properties.
Variables	When you see this style, you must change the value of the content when you copy the text of a sample to a command line. % ec2-register <your-s3-bucket>/image.manifest See also the following symbol convention.

Symbol Conventions

This section describes the common use of symbols.

Convention	Symbol	Description/Example
Mutually exclusive parameters	(Parentheses and vertical bars)	Within a code description, bar separators denote options from which one must be chosen. <pre>% data = hdfread (start stride edge)</pre>
Optional parameters XML variable text	[square brackets]	Within a code description, square brackets denote completely optional commands or parameters. <pre>% sed [-n, -quiet]</pre> Use square brackets in XML examples to differentiate them from tags. <pre><CustomerId>[ID]</CustomerId></pre>
Variables	<arrow brackets>	Within a code sample, arrow brackets denote a variable that must be replaced with a valid value. <pre>% ec2-register <your-s3-bucket>/image.manifest</pre>

Index

A

Access Key ID, [Glossary](#)

Amazon Machine Image (AMI), [Glossary](#)

API

common query parameters, [Common Request Parameters](#)

conventions, [API Conventions](#)

data types, [Data Types and Elastic Load Balancing WSDL](#)

Elastic Load Balancing data types, [Data Types](#)

Query, [Using the Query API](#)

SOAP, [Using the SOAP API](#)

User Scenarios, [User Scenarios](#), [How to Set Up an HTTP LoadBalancer](#), [How to Set Up an HTTPS LoadBalancer](#), [How to Expand a Load Balanced Application to an Additional Availability Zone](#), [How to Disable an Availability Zone from a Load Balanced Application](#), [How to Tear Down an Existing LoadBalancer](#)

APIs

ConfigureHealthCheck, [ConfigureHealthCheck](#)

CreateLoadBalancer, [CreateLoadBalancer](#)

DeleteLoadBalancer, [DeleteLoadBalancer](#)

DeregisterInstancesFromLoadBalancer,

[DeregisterInstancesFromLoadBalancer](#)

DescribeInstanceHealth, [DescribeInstanceHealth](#)

DescribeLoadBalancers, [DescribeLoadBalancers](#)

DisableAvailabilityZonesForLoadBalancer,

[DisableAvailabilityZonesForLoadBalancer](#)

EnableAvailabilityZonesForLoadBalancer,

[EnableAvailabilityZonesForLoadBalancer](#)

RegisterinstancesWithLoadBalancer,

[RegisterInstancesWithLoadBalancer](#)

authentication

Query, [Query API Authentication](#)

signature version 2, [Query API Authentication](#)

SOAP, [Request Authentication](#)

Availability Zone, [Glossary](#)

AvailabilityZones, [Availability Zones and Regions](#),

[AvailabilityZones](#)

C

client error codes, [Error Codes](#)

Conceptual Overviews

Elastic Load Balancing, [Elastic Load Balancing](#)

[Conceptual Overview](#)

ConfigureHealthCheck, [ConfigureHealthCheck](#)

conventions

API, [API Conventions](#)

CreateLoadBalancer, [CreateLoadBalancer](#)

D

data types, [Data Types and Elastic Load Balancing WSDL](#)

AvailabilityZones, [AvailabilityZones](#)

HealthCheck, [HealthCheck](#)

HealthCheckInterval, [HealthCheckInterval](#)

HealthCheckTimeout, [HealthCheckTimeout](#)

HealthyThreshold, [HealthyThreshold](#)

instance, [Instance](#)

InstancePort, [InstancePort](#)

instances, [Instances](#)

instancetype, [InstanceState](#)

InstanceStates, [InstanceStates](#)

Listener, [Listener](#)

Listeners, [Listeners](#)

LoadBalancerDescription, [LoadBalancerDescription](#)

LoadBalancerNames, [LoadBalancerNames](#)

LoadBalancersDescriptions,

[LoadBalancersDescriptions](#)

RequestId, [The Request ID](#)

UnhealthyThreshold, [UnhealthyThreshold](#)

DeleteLoadBalancer, [DeleteLoadBalancer](#)

DeregisterInstancesFromLoadBalancer,

[DeregisterInstancesFromLoadBalancer](#)

DescribeInstanceHealth, [DescribeInstanceHealth](#)
DescribeLoadBalancers, [DescribeLoadBalancers](#)
DisableAvailabilityZonesForLoadBalancer,
[DisableAvailabilityZonesForLoadBalancer](#)

E

Elastic Load Balancing

API, [Elastic Load Balancing Actions](#)

data types, [Data Types](#)

introductory overview, [What is Elastic Load Balancing?](#)

major features, [Benefits](#)

EnableAvailabilityZonesForLoadBalancer,
[EnableAvailabilityZonesForLoadBalancer](#)

error codes, [Error Codes](#)

H

HealthCheck, [HealthCheck](#)

HealthCheckInterval, [HealthCheckInterval](#)

HealthCheckTimeout, [HealthCheckTimeout](#)

HealthyThreshold, [HealthyThreshold](#)

I

instance, [Instance](#)

InstancePort, [InstancePort](#)

instances, [Instances](#)
instancestate, [InstanceState](#)
InstanceStates, [InstanceStates](#)

K

key terms

AvailabilityZone, [Availability Zones and Regions](#)
LoadBalancer, [LoadBalancer](#)

L

Listener, [Listener](#)
Listeners, [Listeners](#)
LoadBalancer, [LoadBalancer](#), [Glossary](#)
LoadBalancerDescription, [LoadBalancerDescription](#)
LoadBalancerNames, [LoadBalancerNames](#)
LoadBalancersDescriptions, [LoadBalancersDescriptions](#)

O

operations

Elastic Load Balancing, [Elastic Load Balancing
Actions](#)

Overviews

Elastic Load Balancing , [Introduction to Elastic Load
Balancing](#)

What Is Elastic Load Balancing?, [What is Elastic](#)

Load Balancing?

P

programming language support, [Programming Language Support](#)

Q

Query

API, [Using the Query API](#)

authentication, [Query API Authentication](#)

parameters, [Query Parameters](#)

query

common parameters, [Common Request Parameters](#)

R

region, EC2 region, [Glossary](#)

RegisterinstancesWithLoadBalancer,

[RegisterInstancesWithLoadBalancer](#)

RequestId, [The Request ID](#)

response structure, [The Response Structure](#)

response structure, SOAP, [The Response Structure](#)

ResponseMetadata

RequestId, [The Request ID](#)

S

Secret Access Key, [Glossary](#)
server error codes, [Error Codes](#)
signature version 2, [Query API Authentication](#)
SOAP
API, [Using the SOAP API](#)
authentication, [Request Authentication](#)
response structure, [The Response Structure](#)
WSDL, [WSDL and Schema Definitions](#)

U

unbounded, [Glossary](#)
UnhealthyThreshold, [UnhealthyThreshold](#)
User Scenarios
API, [User Scenarios](#)
How to Disable an Availability Zone from a Load
Balanced Application, [How to Disable an Availability
Zone from a Load Balanced Application](#)
How to Expand Load Balanced Application to an
Additional Availability Zone, [How to Expand a Load
Balanced Application to an Additional Availability
Zone](#)
How to Set Up an HTTP LoadBalancer, [How to Set
Up an HTTP LoadBalancer](#)
How to Set Up an HTTPS LoadBalancer, [How to Set
Up an HTTPS LoadBalancer](#)
How to Tear Down Existing An LoadBalancer, [How](#)

[to Tear Down an Existing LoadBalancer](#)

W

web services references, [Web Services References](#)
WSDL, [WSDL and Schema Definitions](#)