Elastic Load Balancing

API Reference

API Version 2011-11-15

Elastic Load Balancing: API Reference

Table of Contents

Welcome

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.

This document was last updated on December 20, 2011.

Actions

The actions described in this guide are called using the AWS Query protocol.

The following actions are supported:

ApplySecurityGroupsToLoadBalancer

Description

Associates one or more security groups with your LoadBalancer in VPC. The provided security group IDs will override any currently applied security groups.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes
SecurityGroups.member.N	A list of security group IDs to associate with your LoadBalancer in VPC. The security group IDs must be provided as the ID and not the security group name (For example, sg-1234). Type: String list	Yes

Response Elements

The following elements come wrapped in a `ApplySecurityGroupsToLoadBalancerResult` structure.

Name	Description
`SecurityGroups`	A list of security group IDs associated with your LoadBalancer. Type: String list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`InvalidSecurityGroup`	One or more specified security groups do not exist.	400
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

AttachLoadBalancerToSubnets

Description

Adds one or more subnets to the set of configured subnets in the VPC for the LoadBalancer.

The Loadbalancers evenly distribute requests across all of the registered subnets.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes
Subnets.member.N	A list of subnet IDs to add for the LoadBalancer. Type: String list	Yes

Response Elements

The following elements come wrapped in a `AttachLoadBalancerToSubnetsResult` structure.

Name	Description
`Subnets`	A list of subnet IDs added for the LoadBalancer. Type: String list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`InvalidSubnet`	The VPC has no Internet gateway.	400
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400
`SubnetNotFound`	One or more subnets were not found.	400

ConfigureHealthCheck

Description

Enables the client to define an application healthcheck for the instances.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
HealthCheck	A structure containing the configuration information for the new healthcheck. Type: HealthCheck (p. 35)	Yes
LoadBalancerName	The mnemonic name associated with the LoadBalancer. This name must be unique within the client AWS account. Type: String	Yes

Response Elements

The following elements come wrapped in a `ConfigureHealthCheckResult` structure.

Name	Description
`HealthCheck`	The updated healthcheck for the instances. Type: HealthCheck (p. 35)

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

CreateAppCookieStickinessPolicy

Description

Generates a stickiness policy with sticky session lifetimes that follow that of an application-generated cookie. This policy can be associated only with HTTP/HTTPS listeners.

This policy is similar to the policy created by CreateLBCookieStickinessPolicy, except that the lifetime of the special Elastic Load Balancing cookie follows the lifetime of the application-generated cookie specified in the policy configuration. The LoadBalancer only inserts a new stickiness cookie when the application response includes a new application cookie.

If the application cookie is explicitly removed or expires, the session stops being sticky until a new application cookie is issued.

Note

An application client must receive and send two cookies: the application-generated cookie and the special Elastic Load Balancing cookie named `AWSELB`. This is the default behavior for many common web browsers.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
CookieName	Name of the application cookie used for stickiness. Type: String	Yes
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes
PolicyName	The name of the policy being created. The name must be unique within the set of policies for this LoadBalancer. Type: String	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`DuplicatePolicyName`	Policy with the same name exists for this LoadBalancer. Please choose another name.	400
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400
`TooManyPolicies`	Quota for number of policies for this LoadBalancer has already been reached.	400

CreateLBCookieStickinessPolicy

Description

Generates a stickiness policy with sticky session lifetimes controlled by the lifetime of the browser (user-agent) or a specified expiration period. This policy can be associated only with HTTP/HTTPS listeners.

When a LoadBalancer implements this policy, the LoadBalancer uses a special cookie to track the backend server instance for each request. When the LoadBalancer receives a request, it first checks to see if this cookie is present in the request. If so, the LoadBalancer sends the request to the application server specified in the cookie. If not, the LoadBalancer sends the request to a server that is chosen based on the existing load balancing algorithm.

A cookie is inserted into the response for binding subsequent requests from the same user to that server. The validity of the cookie is based on the cookie expiration time, which is specified in the policy configuration.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
CookieExpirationPeriod	The time period in seconds after which the cookie should be considered stale. Not specifying this parameter indicates that the sticky session will last for the duration of the browser session. Type: Long	No
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes
PolicyName	The name of the policy being created. The name must be unique within the set of policies for this LoadBalancer. Type: String	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
DuplicatePolicyName	Policy with the same name exists for this LoadBalancer. Please choose another name.	400
InvalidConfigurationRequest	Requested configuration change is invalid.	409
LoadBalancerNotFound	The specified LoadBalancer could not be found.	400
TooManyPolicies	Quota for number of policies for this LoadBalancer has already been reached.	400

CreateLoadBalancer

Description

Creates a new LoadBalancer.

After the call has completed successfully, a new LoadBalancer is created; however, it will not be usable until at least one instance has been registered. When the LoadBalancer creation is completed, the client can check whether or not it is usable by using the DescribeInstanceHealth API. The LoadBalancer is usable as soon as any registered instance is *InService*.

Note

Currently, the client's quota of LoadBalancers is limited to ten per Region.

Note

LoadBalancer DNS names vary depending on the Region they're created in. For LoadBalancers created in the United States, the DNS name ends with:

- *us-east-1.elb.amazonaws.com* (for the US Standard Region)
- *us-west-1.elb.amazonaws.com* (for the Northern California Region)

For LoadBalancers created in the EU (Ireland) Region, the DNS name ends with:

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

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
`AvailabilityZones.member.N`	A list of Availability Zones. At least one Availability Zone must be specified. Specified Availability Zones must be in the same EC2 Region as the LoadBalancer. Traffic will be equally distributed across all zones. This list can be modified after the creation of the LoadBalancer. Type: String list	No
`Listeners.member.N`	A list of the following tuples: LoadBalancerPort, InstancePort, and Protocol. Type: Listener (p. 37) list	Yes
`LoadBalancerName`	The name associated with the LoadBalancer. The name must be unique within your set of LoadBalancers. Type: String	Yes

Name	Description	Required
SecurityGroups.member.N	The security groups assigned to your LoadBalancer within your VPC. Type: String list	No
Subnets.member.N	A list of subnet IDs in your VPC to attach to your LoadBalancer. Type: String list	No

Response Elements

The following elements come wrapped in a `CreateLoadBalancerResult` structure.

Name	Description
`DNSName`	The DNS name for the LoadBalancer. Type: String

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`CertificateNotFound`	The specified SSL ID does not refer to a valid SSL certificate in the AWS Identity and Access Management Service.	400
`DuplicateLoadBalancerName`	LoadBalancer name already exists for this account. Please choose another name.	400
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`InvalidSecurityGroup`	One or more specified security groups do not exist.	400
`InvalidSubnet`	The VPC has no Internet gateway.	400
`SubnetNotFound`	One or more subnets were not found.	400
`TooManyLoadBalancers`	The quota for the number of LoadBalancers has already been reached.	400

CreateLoadBalancerListeners

Description

Creates one or more listeners on a LoadBalancer for the specified port. If a listener with the given port does not already exist, it will be created; otherwise, the properties of the new listener must match the properties of the existing listener.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
Listeners.member.N	A list of `LoadBalancerPort`, `InstancePort`, `Protocol`, and `SSLCertificateId` items. Type: Listener (p. 37) list	Yes
LoadBalancerName	The name of the new LoadBalancer. The name must be unique within your AWS account. Type: String	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`CertificateNotFound`	The specified SSL ID does not refer to a valid SSL certificate in the AWS Identity and Access Management Service.	400
`DuplicateListener`	A `Listener` already exists for the given `LoadBalancerName` and `LoadBalancerPort`, but with a different `InstancePort`, `Protocol`, or `SSLCertificateId`.	400
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

CreateLoadBalancerPolicy

Description

Creates a new policy that contains the necessary attributes depending on the policy type. Policies are settings that are saved for your Elastic LoadBalancer and that can be applied to the front-end listener, or the back-end application server, depending on your policy type.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The name associated with the LoadBalancer for which the policy is being created. This name must be unique within the client AWS account. Type: String	Yes
PolicyAttributes.member.N	A list of attributes associated with the policy being created. Type: PolicyAttribute (p. 41) list	No
PolicyName	The name of the LoadBalancer policy being created. The name must be unique within the set of policies for this LoadBalancer. Type: String	Yes
PolicyTypeName	The name of the base policy type being used to create this policy. To get the list of policy types, use the DescribeLoadBalancerPolicyTypes (p. 18) action. Type: String	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
DuplicatePolicyName	Policy with the same name exists for this LoadBalancer. Please choose another name.	400
InvalidConfigurationRequest	Requested configuration change is invalid.	409
LoadBalancerNotFound	The specified LoadBalancer could not be found.	400
PolicyTypeNotFound	One or more of the specified policy types do not exist.	400
TooManyPolicies	Quota for number of policies for this LoadBalancer has already been reached.	400

DeleteLoadBalancer

Description

Deletes the specified LoadBalancer.

If attempting to recreate the LoadBalancer, the client must reconfigure all the settings. The DNS name associated with a deleted LoadBalancer will 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 client instances. The client will not receive the same DNS name even if a new LoadBalancer with same LoadBalancerName is created.

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

Note

By design, if the LoadBalancer does not exist or has already been deleted, DeleteLoadBalancer still succeeds.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes

DeleteLoadBalancerListeners

Description

Deletes listeners from the LoadBalancer for the specified port.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The mnemonic name associated with the LoadBalancer. Type: String	Yes
LoadBalancerPorts.member.N	The client port number(s) of the LoadBalancerListener(s) to be removed. Type: Integer list	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
LoadBalancerNotFound	The specified LoadBalancer could not be found.	400

DeleteLoadBalancerPolicy

Description

Deletes a policy from the LoadBalancer. The specified policy must not be enabled for any listeners.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The mnemonic name associated with the LoadBalancer. The name must be unique within your AWS account. Type: String	Yes
PolicyName	The mnemonic name for the policy being deleted. Type: String	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
InvalidConfigurationRequest	Requested configuration change is invalid.	409
LoadBalancerNotFound	The specified LoadBalancer could not be found.	400

DeregisterInstancesFromLoadBalancer

Description

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

In order to successfully call this API, the same account credentials as those used to create the LoadBalancer must be provided.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
Instances.member.N	A list of EC2 instance IDs consisting of all instances to be deregistered. Type: Instance (p. 36) list	Yes
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes

Response Elements

The following elements come wrapped in a `DeregisterInstancesFromLoadBalancerResult` structure.

Name	Description
`Instances`	An updated list of remaining instances registered with the LoadBalancer. Type: Instance (p. 36) list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidInstance`	The specified EndPoint is not valid.	400
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

DescribeInstanceHealth

Description

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.

Note

The client must have created the specified input LoadBalancer in order to retrieve this information; the client must provide the same account credentials as those that were used to create the LoadBalancer.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
Instances.member.N	A list of instance IDs whose states are being queried. Type: Instance (p. 36) list	No
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes

Response Elements

The following elements come wrapped in a `DescribeInstanceHealthResult` structure.

Name	Description
`InstanceStates`	A list containing health information for the specified instances. Type: InstanceState (p. 36) list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidInstance`	The specified EndPoint is not valid.	400
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

DescribeLoadBalancerPolicies

Description

Returns detailed descriptions of the policies. If you specify a LoadBalancer name, the operation returns either the descriptions of the specified policies, or descriptions of all the policies created for the LoadBalancer. If you don't specify a LoadBalancer name, the operation returns descriptions of the specified sample policies, or descriptions of all the sample policies. The names of the sample policies have the `ELBSample-` prefix.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
`LoadBalancerName`	The mnemonic name associated with the LoadBalancer. If no name is specified, the operation returns the attributes of either all the sample policies pre-defined by Elastic Load Balancing or the specified sample polices. Type: String	No
`PolicyNames.member.N`	The names of LoadBalancer policies you've created or Elastic Load Balancing sample policy names. Type: String list	No

Response Elements

The following elements come wrapped in a `DescribeLoadBalancerPoliciesResult` structure.

Name	Description
`PolicyDescriptions`	A list of policy description structures. Type: PolicyDescription (p. 42) list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400
`PolicyNotFound`	One or more specified policies were not found.	400

DescribeLoadBalancerPolicyTypes

Description

Returns meta-information on the specified LoadBalancer policies defined by the Elastic Load Balancing service. The policy types that are returned from this action can be used in a CreateLoadBalancerPolicy (p. 11) action to instantiate specific policy configurations that will be applied to an Elastic LoadBalancer.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
PolicyTypeNames.member.N	Specifies the name of the policy types. If no names are specified, returns the description of all the policy types defined by Elastic Load Balancing service. Type: String list	No

Response Elements

The following elements come wrapped in a `DescribeLoadBalancerPolicyTypesResult` structure.

Name	Description
`PolicyTypeDescriptions`	List of policy type description structures of the specified policy type. If no policy type names are specified, returns the description of all the policy types defined by Elastic Load Balancing service. Type: PolicyTypeDescription (p. 43) list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`PolicyTypeNotFound`	One or more of the specified policy types do not exist.	400

DescribeLoadBalancers

Description

Returns detailed configuration information for the specified LoadBalancers. If no LoadBalancers are specified, the operation returns configuration information for all LoadBalancers created by the caller.

Note

The client must have created the specified input LoadBalancers in order to retrieve this information; the client must provide the same account credentials as those that were used to create the LoadBalancer.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerNames.member.N	A list of names associated with the LoadBalancers at creation time. Type: String list	No
Marker	An optional parameter reserved for future use. Type: String	No

Response Elements

The following elements come wrapped in a `DescribeLoadBalancersResult` structure.

Name	Description
`LoadBalancerDescriptions`	A list of LoadBalancer description structures. Type: LoadBalancerDescription (p. 39) list
`NextMarker`	An optional parameter reserved for future use. Type: String

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

DetachLoadBalancerFromSubnets

Description

Removes subnets from the set of configured subnets in the VPC for the LoadBalancer.

After a subnet is removed all of the EndPoints registered with the LoadBalancer that are in the removed subnet will go into the *OutOfService* state. When a subnet is removed, the LoadBalancer will balance the traffic among the remaining routable subnets for the LoadBalancer.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The name associated with the LoadBalancer to be detached. The name must be unique within the client AWS account. Type: String	Yes
Subnets.member.N	A list of subnet IDs to remove from the set of configured subnets for the LoadBalancer. Type: String list	Yes

Response Elements

The following elements come wrapped in a `DetachLoadBalancerFromSubnetsResult` structure.

Name	Description
`Subnets`	A list of subnet IDs removed from the configured set of subnets for the LoadBalancer. Type: String list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

DisableAvailabilityZonesForLoadBalancer

Description

Removes the specified EC2 Availability Zones from the set of configured Availability Zones for the LoadBalancer.

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

Note

In order for this call to be successful, the client must have created the LoadBalancer. The client must provide the same account credentials as those that were used to create the LoadBalancer.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
AvailabilityZones.member.N	A list of Availability Zones to be removed from the LoadBalancer.  **Note** There must be at least one Availability Zone registered with a LoadBalancer at all times. The client cannot remove all the Availability Zones from a LoadBalancer. Specified Availability Zones must be in the same Region. Type: String list	Yes
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes

Response Elements

The following elements come wrapped in a `DisableAvailabilityZonesForLoadBalancerResult` structure.

Name	Description
`AvailabilityZones`	A list of updated Availability Zones for the LoadBalancer. Type: String list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

EnableAvailabilityZonesForLoadBalancer

Description

Adds 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, the client must ensure that its 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.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
AvailabilityZones.member.N	A list of new Availability Zones for the LoadBalancer. Each Availability Zone must be in the same Region as the LoadBalancer. Type: String list	Yes
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes

Response Elements

The following elements come wrapped in a `EnableAvailabilityZonesForLoadBalancerResult` structure.

Name	Description
`AvailabilityZones`	An updated list of Availability Zones for the LoadBalancer. Type: String list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

RegisterInstancesWithLoadBalancer

Description

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.

Note

In order for this call to be successful, the client must have created the LoadBalancer. The client 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

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
`Instances.member.N`	A list of instance IDs that should be registered with the LoadBalancer. **Note** When the instance is stopped and then restarted, the IP addresses associated with your instance changes. Elastic Load Balancing cannot recognize the new IP address, which prevents it from routing traffic to your instances. We recommend that you de-register your Amazon EC2 instances from your load balancer after you stop your instance, and then register the load balancer with your instance after you've restarted. To de-register your instances from load balancer, use DeregisterInstancesFromLoadBalancer (p. 15) action. Type: Instance (p. 36) list	Yes
`LoadBalancerName`	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes

Response Elements

The following elements come wrapped in a `RegisterInstancesWithLoadBalancerResult` structure.

Name	Description
`Instances`	An updated list of instances for the LoadBalancer. Type: Instance (p. 36) list

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidInstance`	The specified EndPoint is not valid.	400
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400

SetLoadBalancerListenerSSLCertificate

Description

Sets the certificate that terminates the specified listener's SSL connections. The specified certificate replaces any prior certificate that was used on the same LoadBalancer and port.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The name of the the LoadBalancer. Type: String	Yes
LoadBalancerPort	The port that uses the specified SSL certificate. Type: Integer	Yes
SSLCertificateId	The ID of the SSL certificate chain to use. For more information on SSL certificates, see Managing Server Certificates in the AWS Identity and Access Management documentation. Type: String	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
CertificateNotFound	The specified SSL ID does not refer to a valid SSL certificate in the AWS Identity and Access Management Service.	400
InvalidConfigurationRequest	Requested configuration change is invalid.	409
ListenerNotFound	LoadBalancer does not have a listener configured at the given port.	400
LoadBalancerNotFound	The specified LoadBalancer could not be found.	400

SetLoadBalancerPoliciesForBackendServer

Description

Replaces the current set of policies associated with a port on which the back-end server is listening with a new set of policies. After the policies have been created using CreateLoadBalancerPolicy (p. 11), they can be applied here as a list. At this time, only the back-end server authentication policy type can be applied to the back-end ports; this policy type is composed of multiple public key policies.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
InstancePort	The port number associated with the back-end server. Type: Integer	Yes
LoadBalancerName	The mnemonic name associated with the LoadBalancer. This name must be unique within the client AWS account. Type: String	Yes
PolicyNames.member.N	List of policy names to be set. If the list is empty, then all current polices are removed from the back-end server. Type: String list	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
InvalidConfigurationRequest	Requested configuration change is invalid.	409
LoadBalancerNotFound	The specified LoadBalancer could not be found.	400
PolicyNotFound	One or more specified policies were not found.	400

SetLoadBalancerPoliciesOfListener

Description

Associates, updates, or disables a policy with a listener on the LoadBalancer. You can associate multiple policies with a listener.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 45).

Name	Description	Required
LoadBalancerName	The name associated with the LoadBalancer. The name must be unique within the client AWS account. Type: String	Yes
LoadBalancerPort	The external port of the LoadBalancer with which this policy applies to. Type: Integer	Yes
PolicyNames.member.N	List of policies to be associated with the listener. Currently this list can have at most one policy. If the list is empty, the current policy is removed from the listener. Type: String list	Yes

Errors

For information about the common errors that all actions use, see Common Errors (p. 47).

Error	Description	HTTP Status Code
`InvalidConfigurationRequest`	Requested configuration change is invalid.	409
`ListenerNotFound`	LoadBalancer does not have a listener configured at the given port.	400
`LoadBalancerNotFound`	The specified LoadBalancer could not be found.	400
`PolicyNotFound`	One or more specified policies were not found.	400

Data Types

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

Note

The order of each element in the response is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

AppCookieStickinessPolicy

Description

The AppCookieStickinessPolicy data type.

Contents

Name	Description
`CookieName`	The name of the application cookie used for stickiness. Type: String
`PolicyName`	The mnemonic name for the policy being created. The name must be unique within a set of policies for this LoadBalancer. Type: String

ApplySecurityGroupsToLoadBalancerResult

Description

The out for the ApplySecurityGroupsToLoadBalancer (p. 3) action.

Contents

Name	Description
`SecurityGroups`	A list of security group IDs associated with your LoadBalancer. Type: String list

AttachLoadBalancerToSubnetsResult

Description

The output for the AttachLoadBalancerToSubnets (p. 4) action.

Contents

Name	Description
`Subnets`	A list of subnet IDs added for the LoadBalancer. Type: String list

BackendServerDescription

Description

This data type is used as a response element in the DescribeLoadBalancers (p. 19) action to describe the configuration of the back-end server.

Contents

Name	Description
`InstancePort`	Provides the port on which the back-end server is listening. Type: Integer
`PolicyNames`	Provides a list of policy names enabled for the back-end server. Type: String list

ConfigureHealthCheckResult

Description

The output for the ConfigureHealthCheck (p. 5) action.

Contents

Name	Description
`HealthCheck`	The updated healthcheck for the instances. Type: HealthCheck (p. 35)

CreateLoadBalancerPolicyResult

Description

The output for the CreateLoadBalancerPolicy (p. 11) action.

Contents

CreateLoadBalancerResult

Description

The output for the CreateLoadBalancer (p. 8) action.

Contents

Name	Description
DNSName	The DNS name for the LoadBalancer. Type: String

DeregisterInstancesFromLoadBalancerResult

Description

The output for the DeregisterInstancesFromLoadBalancer (p. 15) action.

Contents

Name	Description
Instances	An updated list of remaining instances registered with the LoadBalancer. Type: Instance (p. 36) list

DescribeInstanceHealthResult

Description

The output for the DescribeInstanceHealth (p. 16) action.

Contents

Name	Description
`InstanceStates`	A list containing health information for the specified instances. Type: InstanceState (p. 36) list

DescribeLoadBalancerPoliciesResult

Description

The output for the DescribeLoadBalancerPolicies (p. 17) action.

Contents

Name	Description
`PolicyDescriptions`	A list of policy description structures. Type: PolicyDescription (p. 42) list

DescribeLoadBalancerPolicyTypesResult

Description

The output for the DescribeLoadBalancerPolicyTypes (p. 18) action.

Contents

Name	Description
`PolicyTypeDescriptions`	List of policy type description structures of the specified policy type. If no policy type names are specified, returns the description of all the policy types defined by Elastic Load Balancing service. Type: PolicyTypeDescription (p. 43) list

DescribeLoadBalancersResult

Description

The output for the DescribeLoadBalancers (p. 19) action.

Contents

Name	Description
`LoadBalancerDescriptions`	A list of LoadBalancer description structures. Type: LoadBalancerDescription (p. 39) list
`NextMarker`	An optional parameter reserved for future use. Type: String

DetachLoadBalancerFromSubnetsResult

Description

The output for the DetachLoadBalancerFromSubnets (p. 20) action.

Contents

Name	Description
`Subnets`	A list of subnet IDs removed from the configured set of subnets for the LoadBalancer. Type: String list

DisableAvailabilityZonesForLoadBalancerResult

Description

The output for the DisableAvailabilityZonesForLoadBalancer (p. 21) action.

Contents

Name	Description
`AvailabilityZones`	A list of updated Availability Zones for the LoadBalancer. Type: String list

EnableAvailabilityZonesForLoadBalancerResult

Description

The output for the EnableAvailabilityZonesForLoadBalancer (p. 23) action.

Contents

Name	Description
`AvailabilityZones`	An updated list of Availability Zones for the LoadBalancer. Type: String list

HealthCheck

Description

The HealthCheck data type.

Contents

Name	Description
`HealthyThreshold`	Specifies the number of consecutive health probe successes required before moving the instance to the *Healthy* state. Type: Integer
`Interval`	Specifies the approximate interval, in seconds, between health checks of an individual instance. Type: Integer
`Target`	Specifies the instance being checked. The protocol is either TCP, HTTP, HTTPS, or SSL. The range of valid ports is one (1) through 65535.  **Note** 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 unhealthy. SSL is also specified as SSL: port pair, for example, SSL:5000. For HTTP or HTTPS protocol, the situation is different. You have to include a ping path in the string. 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 unhealthy. The total length of the HTTP ping target needs to be 1024 16-bit Unicode characters or less. Type: String

Name	Description
`Timeout`	Specifies the amount of time, in seconds, during which no response means a failed health probe. **Note** This value must be less than the *Interval* value. Type: Integer
`UnhealthyThreshold`	Specifies the number of consecutive health probe failures required before moving the instance to the *Unhealthy* state. Type: Integer

Instance

Description

The Instance data type.

Contents

Name	Description
`InstanceId`	Provides an EC2 instance ID. Type: String

InstanceState

Description

The InstanceState data type.

Contents

Name	Description
`Description`	Provides a description of the instance. Type: String
`InstanceId`	Provides an EC2 instance ID. Type: String
`ReasonCode`	Provides information about the cause of *OutOfService* instances. Specifically, it indicates whether the cause is Elastic Load Balancing or the instance behind the LoadBalancer. Type: String

Name	Description
`State`	Specifies the current status of the instance. Type: String

LBCookieStickinessPolicy

Description

The LBCookieStickinessPolicy data type.

Contents

Name	Description
`CookieExpirationPeriod`	The time period in seconds after which the cookie should be considered stale. Not specifying this parameter indicates that the stickiness session will last for the duration of the browser session. Type: Long
`PolicyName`	The name for the policy being created. The name must be unique within the set of policies for this LoadBalancer. Type: String

Listener

Description

The Listener data type.

Contents

Name	Description
`InstancePort`	Specifies the TCP port on which the instance server is listening. This property cannot be modified for the life of the LoadBalancer. Type: Integer

Name	Description
`InstanceProtocol`	Specifies the protocol to use for routing traffic to back-end instances - HTTP, HTTPS, TCP, or SSL. This property cannot be modified for the life of the LoadBalancer. **Note** If the front-end protocol is HTTP or HTTPS, `InstanceProtocol` has to be at the same protocol layer, i.e., HTTP or HTTPS. Likewise, if the front-end protocol is TCP or SSL, InstanceProtocol has to be TCP or SSL. **Note** If there is another listener with the same `InstancePort` whose `InstanceProtocol` is secure, i.e., HTTPS or SSL, the listener's `InstanceProtocol` has to be secure, i.e., HTTPS or SSL. If there is another listener with the same `InstancePort` whose `InstanceProtocol` is HTTP or TCP, the listener's `InstanceProtocol` must be either HTTP or TCP. Type: String
`LoadBalancerPort`	Specifies the external LoadBalancer port number. This property cannot be modified for the life of the LoadBalancer. Type: Integer
`Protocol`	Specifies the LoadBalancer transport protocol to use for routing - HTTP, HTTPS, TCP or SSL. This property cannot be modified for the life of the LoadBalancer. Type: String
`SSLCertificateId`	The ARN string of the server certificate. To get the ARN of the server certificate, call the AWS Identity and Access Management UploadServerCertificate API. Type: String

ListenerDescription

Description

The ListenerDescription data type.

Contents

Name	Description
`Listener`	The Listener data type. Type: Listener (p. 37)

Name	Description
PolicyNames	A list of policies enabled for this listener. An empty list indicates that no policies are enabled. Type: String list

LoadBalancerDescription

Description

Contains the result of a successful invocation of DescribeLoadBalancers (p. 19).

Contents

Name	Description
AvailabilityZones	Specifies a list of Availability Zones. Type: String list
BackendServerDescriptions	Contains a list of back-end server descriptions. Type: BackendServerDescription (p. 31) list
CanonicalHostedZoneName	Provides the name of the Amazon Route 53 hosted zone that is associated with the LoadBalancer. For information on how to associate your load balancer with a hosted zone, go to Using Domain Names With Elastic Load Balancing in the *Elastic Load Balancing Developer Guide*. Type: String
CanonicalHostedZoneNameID	Provides the ID of the Amazon Route 53 hosted zone name that is associated with the LoadBalancer. For information on how to associate or disassociate your load balancer with a hosted zone, go to Using Domain Names With Elastic Load Balancing in the *Elastic Load Balancing Developer Guide*. Type: String
CreatedTime	Provides the date and time the LoadBalancer was created. Type: DateTime
DNSName	Specifies the external DNS name associated with the LoadBalancer. Type: String
HealthCheck	Specifies information regarding the various health probes conducted on the LoadBalancer. Type: HealthCheck (p. 35)
Instances	Provides a list of EC2 instance IDs for the LoadBalancer. Type: Instance (p. 36) list

Name	Description
`ListenerDescriptions`	LoadBalancerPort, InstancePort, Protocol, InstanceProtocol, and PolicyNames are returned in a list of tuples in the ListenerDescriptions element. Type: ListenerDescription (p. 38) list
`LoadBalancerName`	Specifies the name associated with the LoadBalancer. Type: String
`Policies`	Provides a list of policies defined for the LoadBalancer. Type: Policies (p. 40)
`SecurityGroups`	The security groups the LoadBalancer is a member of (VPC only). Type: String list
`SourceSecurityGroup`	The security group that you can use as part of your inbound rules for your LoadBalancer's back-end Amazon EC2 application instances. To only allow traffic from LoadBalancers, add a security group rule to your back end instance that specifies this source security group as the inbound source. Type: SourceSecurityGroup (p. 44)
`Subnets`	Provides a list of VPC subnet IDs for the LoadBalancer. Type: String list
`VPCId`	Provides the ID of the VPC attached to the LoadBalancer. Type: String

Policies

Description

The policies data type.

Contents

Name	Description
`AppCookieStickinessPolicies`	A list of the AppCookieStickinessPolicy (p. 30) objects created with CreateAppCookieStickinessPolicy (p. 6). Type: AppCookieStickinessPolicy (p. 30) list
`LBCookieStickinessPolicies`	A list of LBCookieStickinessPolicy (p. 37) objects created with CreateAppCookieStickinessPolicy (p. 6). Type: LBCookieStickinessPolicy (p. 37) list
`OtherPolicies`	A list of policy names other than the stickiness policies. Type: String list

PolicyAttribute

Description

The PolicyAttribute (p. 41) data type. This data type contains a key/value pair that defines properties of a specific policy.

Contents

Name	Description
`AttributeName`	The name of the attribute associated with the policy. Type: String
`AttributeValue`	The value of the attribute associated with the policy. Type: String

PolicyAttributeDescription

Description

The `PolicyAttributeDescription` data type. This data type is used to describe the attributes and values associated with a policy.

Contents

Name	Description
`AttributeName`	The name of the attribute associated with the policy. Type: String
`AttributeValue`	The value of the attribute associated with the policy. Type: String

PolicyAttributeTypeDescription

Description

The `PolicyAttributeTypeDescription` data type. This data type is used to describe values that are acceptable for the policy attribute.

Contents

Name	Description
`AttributeName`	The name of the attribute associated with the policy type. Type: String
`AttributeType`	The type of attribute. For example, Boolean, Integer, etc. Type: String
`Cardinality`	The cardinality of the attribute. Valid Values: • ONE(1) : Single value required • ZERO_OR_ONE(0..1) : Up to one value can be supplied • ZERO_OR_MORE(0..*) : Optional. Multiple values are allowed • ONE_OR_MORE(1..*0) : Required. Multiple values are allowed Type: String
`DefaultValue`	The default value of the attribute, if applicable. Type: String
`Description`	A human-readable description of the attribute. Type: String

PolicyDescription

Description

The `PolicyDescription` data type.

Contents

Name	Description
`PolicyAttributeDescriptions`	A list of policy attribute description structures. Type: PolicyAttributeDescription (p. 41) list
`PolicyName`	The name mof the policy associated with the LoadBalancer. Type: String
`PolicyTypeName`	The name of the policy type associated with the LoadBalancer. Type: String

PolicyTypeDescription

Description

The PolicyTypeDescription (p. 43) data type.

Contents

Name	Description
`Description`	A human-readable description of the policy type. Type: String
`PolicyAttributeTypeDescriptions`	The description of the policy attributes associated with the LoadBalancer policies defined by the Elastic Load Balancing service. Type: PolicyAttributeTypeDescription (p. 41) list
`PolicyTypeName`	The name of the policy type. Type: String

RegisterInstancesWithLoadBalancerResult

Description

The output for the RegisterInstancesWithLoadBalancer (p. 24) action.

Contents

Name	Description
`Instances`	An updated list of instances for the LoadBalancer. Type: Instance (p. 36) list

SetLoadBalancerPoliciesForBackendServerResult

Description

The output for the SetLoadBalancerPoliciesForBackendServer (p. 27) action.

Contents

SourceSecurityGroup

Description

This data type is used as a response element in the DescribeLoadBalancers (p. 19) action. For information about Elastic Load Balancing security groups, go to Using Security Groups With Elastic Load Balancing in the *Elastic Load Balancing Developer Guide*.

Contents

Name	Description
`GroupName`	Name of the source security group. Use this value for the `--source-group` parameter of the `ec2-authorize` command in the Amazon EC2 command line tool. Type: String
`OwnerAlias`	Owner of the source security group. Use this value for the `--source-group-user` parameter of the `ec2-authorize` command in the Amazon EC2 command line tool. Type: String

Common Query Parameters

This section lists the request parameters that all actions use. Any action-specific parameters are listed in the topic for the action.

Parameter Name	Description	Required
Action	The action to perform. Default: None Type: String	Yes
AuthParams	The parameters required to authenticate a query request. Contains: AWSAccessKeyID SignatureVersion Timestamp Signature Default: None	Conditional
AWSAccessKeyId	The Access Key ID corresponding to the AWS Secret Access Key you used to sign the request. Default: None Type: String	Yes
Expires	The date and time at which the request signature expires, in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard. Condition: Requests must include either *Timestamp* or *Expires*, but not both. Default: None Type: String	Conditional

Parameter Name	Description	Required
SecurityToken	The temporary security token obtained through a call to AWS Security Token Service. Only available for actions in the following AWS services: Amazon EC2, Amazon Simple Notification Service, Amazon SQS, and AWS SimpleDB. Default: None Type: String	
Signature	The digital signature you created for the request. Refer to the service's developer documentation for information about how to generate the signature. Default: None Type: String	Yes
SignatureMethod	The hash algorithm you used to create the request signature. Default: None Valid Values: `HmacSHA256` | `HmacSHA1`. Type: String	Yes
SignatureVersion	The signature version you use to sign the request. Set this to the value recommended in your product-specific documentation on security. Default: None Type: String	Yes
Timestamp	The date and time the request was signed, in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard. Condition: Requests must include either *Timestamp* or *Expires*, but not both. Default: None Type: String	Conditional
Version	The API version to use, in the format YYYY-MM-DD. Default: None Type: String	Yes

Common Errors

This section lists the common errors that all actions return. Any action-specific errors are listed in the topic for the action.

Error	Description	HTTP Status Code
`IncompleteSignature`	The request signature does not conform to AWS standards.	400
`InternalFailure`	The request processing has failed due to some unknown error, exception or failure.	500
`InvalidAction`	The action or operation requested is invalid.	400
`InvalidClientTokenId`	The X.509 certificate or AWS Access Key ID provided does not exist in our records.	403
`InvalidParameterCombination`	Parameters that must not be used together were used together.	400
`InvalidParameterValue`	A bad or out-of-range value was supplied for the input parameter.	400
`InvalidQueryParameter`	AWS query string is malformed, does not adhere to AWS standards.	400
`MalformedQueryString`	The query string is malformed.	404
`MissingAction`	The request is missing an action or operation parameter.	400
`MissingAuthenticationToken`	Request must contain either a valid (registered) AWS Access Key ID or X.509 certificate.	403
`MissingParameter`	An input parameter that is mandatory for processing the request is not supplied.	400

Error	Description	HTTP Status Code
`OptInRequired`	The AWS Access Key ID needs a subscription for the service.	403
`RequestExpired`	Request is past expires date or the request date (either with 15 minute padding), or the request date occurs more than 15 minutes in the future.	400
`ServiceUnavailable`	The request has failed due to a temporary failure of the server.	503
`Throttling`	Request was denied due to request throttling.	400