HP Operations Orchestration Software

Software Version: 9.00.05

Amazon Elastic Load Balancing Integration Guide

Document Release Date: April 2011

Software Release Date: April 2011

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notices

Trademark Notices

For information on open-source and third-party software acknowledgements, see in the documentation set for this release, Open-Source and Third-Party Software Acknowledgements (3rdPartyOpenNotices.pdf).

On the Web: Finding OO support and documentation

There are two Web sites where you can find support, patches, and documentation, including updates to OO Help systems, guides, and tutorials. Both sites provide all documentation, including tutorials. In addition:

- OO Support site provides platform patches.
- HP Live Network site (in the OO area) provides content patches.

Support

You can obtain platform patches (that is, patches that fix issues with OO Central, Studio, or other components of OO) and the HP OO documentation set and tutorials from the HP Software Product Manuals Web site. You will need an HP Passport to log in to the Web site.

To obtain HP OO documentation and tutorials

1. Go to the HP Software Product Manuals Web site (*http://support.openview.hp.com/selfsolve/manuals*).
2. Log in with your HP Passport user name and password.

   OR

   If you do not have an HP Passport, click **New users – please register** to create an HP Passport, then return to this page and log in.

   If you need help getting an HP Passport, see your HP OO contact.
3. In the **Product** list box, scroll down to and select **Operations Orchestration**.
4. In the **Product Version** list, click the version of the manuals that you're interested in.
5. In the **Operating System** list, click the relevant operating system.
6. Click the **Search** button.
7. In the **Results** list, click the link for the file that you want.

HP Live Network

For content patches (that is, patches that fix issues with operations or flows), troubleshooting aids, support contract management, product manuals and more, visit the following site: *https://www.www2.hp.com*.

This is the **HP Live Network** Web page. To sign in:

1. Click **Login**.
2. On the **HP Passport sign-in** page, enter your HP Passport user ID and password and then click **Sign-in**.
3. If you do not already have an HP Passport account, do the following:
   a. On the **HP Passport sign-in** page, click **New user registration**.
   b. On the **HP Passport new user registration** page, enter the required information and then click **Continue**.
   c. On the confirmation page that opens, check your information and then click **Register**.

d. On the **Terms of Service** page, read the Terms of use and legal restrictions, select the **Agree** button, and then click **Submit**.

4. On the **HP Live Network** page, click **Operations Orchestration Community.**

   The Operations Orchestration Community page contains links to announcements, discussions, downloads, documentation, help, and support.

Note: Contact your OO contact if you have any difficulties with this process.

In OO: How to find Help, PDFs, and tutorials

The HP Operations Orchestration software (HP OO) documentation set is made up of:

- Help for Central

  Central Help provides information to the following:

  - Finding and running flows
  - For HP OO administrators, configuring the functioning of HP OO
  - Generating and viewing the information available from the outcomes of flow runs

  The Central Help system is also available as a PDF document in the HP OO home directory, in \Central\docs\ .

- Help for Studio

  Studio Help instructs flow authors at varying levels of programming ability.

  The Studio Help system is also available as a PDF document in the HP OO home directory, in \Studio\docs\.

- Animated tutorials for Central and Studio

  HP OO tutorials can each be completed in less than half an hour and provide basic instruction on the following:

  - In Central, finding, running, and viewing information from flows
  - In Studio, modifying flows

  The tutorials are available in the:

  - Studio Welcome pane
  - HP OO\Studio home directory, in the Tutorials subdirectory

- Self-documentation for HP OO operations, flows, and Accelerator Packs

  Self-documentation is available in the descriptions of the operations and steps that are included in the flows.

Table of Contents

Overview of Amazon Elastic Load Balancing integration

Amazon Elastic Load Balancing makes it possible to distribute application load between two or more Amazon Elastic Compute Cloud (EC2) instances.

This integration uses the Elastic Load Balancing Query API released on 11/25/2009. The operations implemented in this integration are the ones currently available in this API release.

Use cases and scenarios

This section describes some user scenarios and shows how Elastic Load Balancing operations can be used to accomplish several common tasks.

Set up an HTTP load balancer

Set up a load balancer for an HTTP service by specifying the ports used by the load balancer, defining a health check, and registering some instances.

1. Run the **Create Load Balancer** operation with the following inputs and values:
   - **availabilityZones** = us-east-1a
   - **protocol** = HTTP
   - **instancePort** = 8080
   - **loadBalancerPort** = 80
   - **loadBalancerName** = MyLoadBalancer
2. Run the **Configure Health Check** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **target** = http:80/ping
   - **interval** = 30
   - **timeout** = 3
   - **healthyThreshold** = 2
   - **unhealthyThreshold =** 2
3. Run the **Register Instances With Load Balancer** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **instanceIds** = The instances that are running in availability zone us-east-1a.

Set up an HTTPS load balancer

Elastic Load Balancing does not have SSL termination capability. So, for HTTPS traffic you must set the protocol for the load balancer listener to TCP. This scenario is similar to the previous one; you create a load balancer with the protocol set to TCP, configure the health check, and then register instances.

1. Run the **Create Load Balancer** operation with the following inputs and values:
   - **availabilityZones** = us-east-1a
   - **protocol** = TCP
   - **instancePort** = 8443
   - **loadBalancerPort** = 443

- **loadBalancerName** = MyLoadBalancer

2. Run the **Configure Health Check** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **target** = TCP:8443 (make sure that your instances accept TCP connections on port 8443).
   - **interval** = 30
   - **timeout** = 3
   - **healthyThreshold** = 2
   - **unhealthyThershold** = 2
3. Run the **Register Instances With Load Balancer** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **instanceIds** = The instances that are running in availability zone us-east-1a.

Expand a load balanced application to an additional availability zone

Register instances with the load balancer, wait for them to be in the **OutOfService** state, and then enable a new availability zone for the load balancer. You must register the instances before you enable the availability zone because when you run the **Enable Availability Zones** operation, the load balancer begins to route traffic to the new zone (the traffic is distributed equally among all availability zones). If no instances are registered in the new zone, all requests assigned there will fail.

1. Run the **Register Instances With Load Balancer** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **instanceIds** = The instances you want to work in the new availability zone.
2. Run the **Describe Instance Health** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **instanceIds** = The instances used in the previous step.

   You must wait until the instances are in the **OutOfService** state.
3. Run the **Enable Availability Zones For Load Balancer** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **availabilityZones** = The new availability zone in which your instances are located (for example us-east-1b).

Disable an availability zone from a load balanced application

Disable an availability zone from a load balanced application. When you deregister all the instances from an availability zone, the load balancer still routes traffic to that zone. To make sure that no invalid requests are made, you must first disable the availability zone from the load balancer, wait for the instances to go in the **OutOfService** state, and then deregister the instances from the load balancer.

1. Run the **Disable Availability Zones For Load Balancer** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **availabilityZones** = The zone you want to disable from the load balancer.
2. Run the **Describe Instance Health** operation with the following inputs and values:

- **loadBalancerName** = MyLoadBalancer
- **instanceIds** = The instances that are in the availability zone you want to disable.

You must wait until are these instances are in the **OutOfService** state.

3. Run the **Deregister Instances From Load Balancer** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **instanceIds** = The instances that are in the availability zone you want to disable.

Tear down an existing load balancer

When you delete a load balancer, the instances are automatically deregistered.

- Run the **Delete load balancer** operation with the following input and value:
  - **loadBalancerName** = MyLoadBalancer

Enable duration-based session stickiness

Create a stickiness policy and use it to enable sticky sessions for a load balancer.

1. Run the **Create LB Cookie Stickiness Policy** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **policyName** = MyLoadBalancerPolicy
   - **cookieExpirationPeriod** = 60

     This makes a cookie stickiness policy with a cookie expiration period of 60 seconds.
2. Run the **Set Load Balancer Policies Of Listener** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **loadBalancerPort** = 80
   - **policyNames** = MyLoadBalancerPolicy

Enable application-controlled session stickiness

This is similar to the previous scenario, but the life of the session follows that of an application-generated cookie.

1. Run the **Create Application Cookie Stickiness Policy** operation with the following inputs and values:
   - **loadBalancerName** = MyLoadBalancer
   - **policyName** = myAppCookiePolicy
   - **cookieName** = myCookie
2. Run the **Set Load Balancer Policies Of Listener** operation with the following inputs and values:
   - **loadBalancerName** = myLoadBalancer
   - **loadBalancerPort** = 80
   - **policyNames** = myAppCookiePolicy

Installation and configuration instructions

No special installation and configuration instructions are required for the Amazon Elastic Load Balancing integration. You can also use Elastic Load Balancing through the Amazon EC2 console. To access the Amazon EC2 console, go to *http://aws.amazon.com/console/* and use your EC2 credentials to log on. Amazon has created specialized plug-ins for several browsers, such as Elasticfox for Firefox, that allow easy interaction with Amazon EC2.

Versions

Operations Orchestration Version	Amazon ELB Version
9.00.05	Elastic Load Balancing API Version 2009-11-25

Architecture

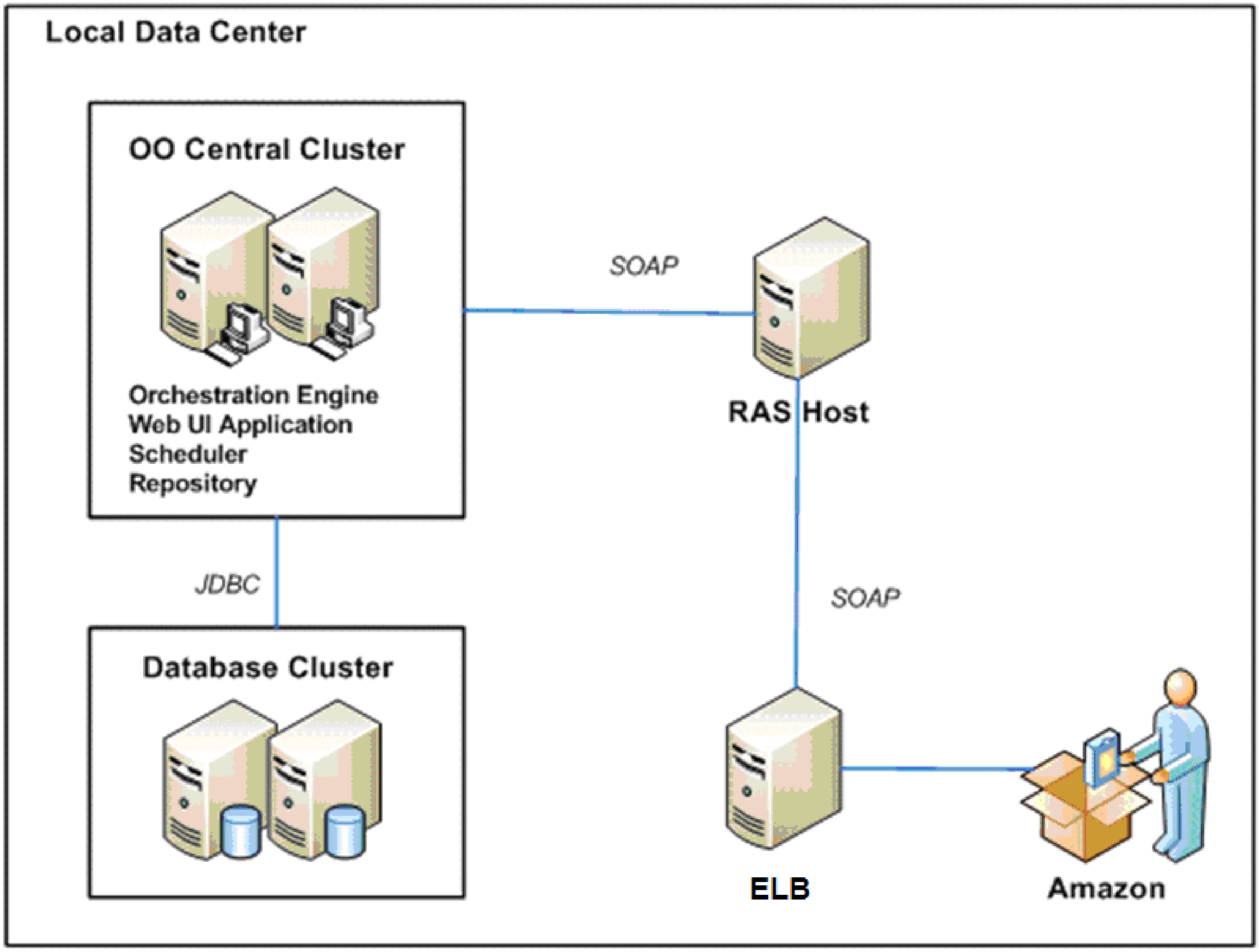
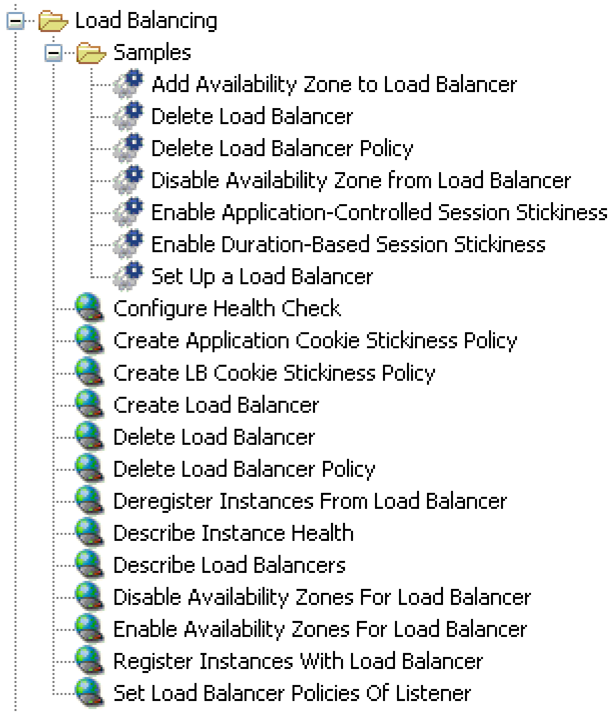


Figure 1 - Amazon Elastic Load Balancing architecture

Elastic Load Balancing integration operation and flow infrastructure

The Elastic Load Balancing integration contains the following operations and flows in the OO Studio Library/Integrations/Amazon/Load Balancing/ folder:



- Load Balancing
  - Samples
    - Add Availability Zone to Load Balancer
    - Delete Load Balancer
    - Delete Load Balancer Policy
    - Disable Availability Zone from Load Balancer
    - Enable Application-Controlled Session Stickiness
    - Enable Duration-Based Session Stickiness
    - Set Up a Load Balancer
  - Configure Health Check
  - Create Application Cookie Stickiness Policy
  - Create LB Cookie Stickiness Policy
  - Create Load Balancer
  - Delete Load Balancer
  - Delete Load Balancer Policy
  - Deregister Instances From Load Balancer
  - Describe Instance Health
  - Describe Load Balancers
  - Disable Availability Zones For Load Balancer
  - Enable Availability Zones For Load Balancer
  - Register Instances With Load Balancer
  - Set Load Balancer Policies Of Listener

Figure 2 – Amazon Elastic Load Balancing integration operation and flow infrastructure

Common inputs in the integration

The following inputs are used in every Load Balancing integration operation.

accessKeyId

The ID of the secret access key associated with the Amazon AWS account.

accessKey

The secret access key associated with the Amazon AWS account.

proxyHost

The proxy server used to connect to Amazon.

proxyPort

The port to connect to the proxy server; **proxyHost** and **proxyPort** must be either empty or both specified.

serviceEndpoint

The endpoint to which requests are sent. The default value is **elasticloadbalancing.us-east-1.amazonaws.com** which corresponds to the United States region.

Operation and flow specifics

This section describes the Amazon Elastic Load Balancing integration's operations, including any operation- or flow-specific inputs.

The sample flows in the OO Library/Integrations/Amazon/Load Balancing/ folder implement the common user scenarios presented in the *Use cases and scenarios* section.

Configure Health Check

The **Configure Health Check** operation modifies the parameters of the health check associated with the load balancer.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name associated with the load balancer.

healthyThreshold

The number of consecutive health probe successes required before moving the instance to the **Healthy** state. The valid values are **2** through **10**. The default value is **10**.

interval

The approximate interval (in seconds) between health checks of an individual instance. The valid values are **6** through **300** seconds. The default value is **30** seconds. The interval value must be greater than the value of the **timeout** input.

target

The protocol, port, and path to ping for the health check. The protocol is either **TCP** or **HTTP**. The range of valid ports is **1** through **65535**.

timeout

The amount of time (in seconds) during which no response means a failed health probe. The valid values are **2** through **60** seconds. The default value is **5** seconds. The **timeout** value must be less than the **interval** input value.

unhealthyThreshold

The number of consecutive health probe failures that move the instance to the **Unhealthy** state. The valid values are **2** through **10**. The default value is **2**.

The operation returns the following:

returnResult

Returns **true** if the operation completes successfully; otherwise an error message is returned.

requestId

The ID of the request you have sent to Amazon by using this operation.

healthyThreshold

The number of consecutive health probe successes required before moving the instance to the **Healthy** state.

interval

The approximate interval (in seconds) between health checks of an individual instance.

target

The protocol, port, and path pinged for the health check.

timeout

The amount of time (in seconds) during which no response means a failed health probe.

unhealthyThreshold

The number of consecutive health probe failures that move the instance to the **Unhealthy** state.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- For the **target** input, if you specify a value of **TCP** use the format `TCP:port` (for example, **TCP:5000**). In this case, a health check attempts to open a TCP connection to the instance on the specified port. Failure to connect within the configured timeout is considered unhealthy.

  If you specify a value of **HTTP**, use a path to a file located on the instance (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 response other than **200 OK** within the timeout period is considered unhealthy.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is ...." message is returned.

Create Application Cookie Stickiness Policy

The **Create Application Cookie Stickiness Policy** operation creates a stickiness policy with sticky session lifetime that follows an application-generated cookie. This policy can only be associated with HTTP listeners.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name associated with the load balancer.

policyName

The name for the policy being created. The name must be unique within the set of policies for this load balancer.

cookieName

The name of the application cookie used for stickiness.

The operation returns the following:

returnResult

Returns **true** if the operation completes successfully; otherwise an error message is returned.

requestId

The ID of the request you have sent to Amazon by using this operation.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is ...." message is returned.

Create LB Cookie Stickiness Policy

The **Create LB Cookie Stickiness Policy** operation creates a stickiness policy with the sticky session lifetime controlled by the lifetime of the browser or a specified expiration period. This policy can only be associated with HTTP listeners.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name associated with the load balancer.

policyName

The name for the policy being created. The name must be unique within the set of policies for this load balancer.

cookieExpirationPeriod

The time period in seconds after which the cookie should be considered stale. If you do not specify a value for this input, the sticky session lasts for the duration of the browser session.

The operation returns the following:

returnResult

Returns **true** if the operation completes successfully; otherwise an error message is returned.

requestId

The ID of the request you have sent to Amazon by using this operation.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is ...." message is returned.

Create Load Balancer

The **Create Load Balancer** operation creates a load balancer. The load balancer is not usable until at least one instance has been registered.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

availabilityZones

A list of availability zones. You can modify this list after you create the load balancer. The availability zones specified must be in the same EC2 region as the load balancer.

loadBalancerName

The mnemonic name associated with the load balancer.

protocol1

The protocol for which the load balancer is created.

loadBalancerPort1

The port that the load balancer listens on.

instancePort1

The port on the instances on which the load balancer distributes traffic.

protocol[number]

To add more protocols, you must add extra inputs using the naming convention `protocol[number]` where you replace `[number]` with an increasing value (for example, **protocol2**, **protocol3**).

loadBalancerPort[number]

To add more load balancer ports, you must add extra inputs using the naming convention `loadBalancerPort[number]` where you replace `[number]` with an increasing value (for example, **loadBalancerPort2**, **loadBalancerPort3**).

instancePort[number]

To add more instance ports, you must add extra inputs using the naming convention `instancePort[number]` where you replace `[number]` with an increasing value (for example, **instancePort2**, **instancePort3**).

delimiter

The delimiter used to separate the availability zones. The default value is a comma (,).

The operation returns the following:

requestId

The ID of the request you have sent to Amazon by using this operation.

returnResult

The DNS name for the load balancer.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is ...." message is returned.
- The **loadBalancerName** value must be unique within the set of load balancers. You can only use alphanumeric characters (case sensitive) and a dash (-). The dash cannot be the first or last character of the name. The maximum length of the name is 32 characters.
- If you add a **protocol[number]**, **loadBalancerPort[number]**, or **instancePort[number]** input with a valid value, then you must also add all of the other **[number]** inputs and give them valid values as well.

Delete Load Balancer

The **Delete Load Balancer** operation deletes the specified load balancer.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name of the load balancer.

The operation returns the following:

requestId

The ID of the request you have sent to Amazon by using this operation.

returnResult

Returns **true** if the operation completes successfully; otherwise an error message is returned.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is …." message is returned.
- The DNS name associated with a deleted load balancer is no longer usable. Even if you recreate a load balancer with the same name, the DNS name is different.
- Because the Amazon load balancer API was designed to be idempotent, even if the load balancer does not exist or was deleted, the operation still returns with success.

Delete Load Balancer Policy

The **Delete Load Balancer Policy** operation deletes a policy from the load balancer. The specified policy must not be enabled for any listeners.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name of the load balancer.

policyName

The name of the policy to delete.

The operation returns the following:

requestId

The ID of the request you have sent to Amazon by using this operation.

returnResult

Returns **true** if the operation completes successfully; otherwise an error message is returned.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is …." message is returned.
- The operation does not fail if the specified policy does not exist.

Deregister Instances From Load Balancer

The **Deregister Instances From Load Balancer** operation removes the specified instances from the specified load balancer.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

instanceIds

The IDs of the instances to be deregistered from the load balancer.

loadBalancerName

The name of the load balancer.

delimiter

The delimiter used to separate the instance IDs. The default value is a comma (,).

The operation returns the following:

requestId

The ID of the request you have sent to Amazon by using this operation.

returnResult

A list of all instance IDs registered with the load balancer, separated by the delimiter.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is …." message is returned.
- Trying to deregister an instance that is not registered with the load balancer does nothing and returns with success.

Describe Instance Health

The **Describe Instance Health** operation returns the current state of the instances of the specified load balancer. If no instances are specified, the state of all the instances for the load balancer is returned.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name associated with the load balancer.

instanceIds

The list of instance IDs whose state is being queried. If the list is empty, it returns the state of all the instances registered with the load balancer.

delimiter

The delimiter for the list of instance IDs. The default value is a comma (,).

rowDelimiter

The delimiter used between rows in the output table. The default value is line separator of the current operating system.

colDelimiter

The delimiter used between columns in the output table. The default value is a semicolon (;).

The operation returns the following:

returnResult

A table containing the following columns: **instanceId**, **state** (**InService**, **OutOfService**), **reasonCode** (provides more information about the cause of **OutOfService** instances), and **description**. Each row in the table describes an instance state.

requestId

The ID of the request you have sent to Amazon by using this operation.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is …." message is returned.

Describe Load Balancers

The **Describe Load Balancers** operation returns a detailed description of the specified load balancers. If no load balancers are specified, a description of all the available load balancers is returned.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerNames

A list containing the names of the load balancers.

delimiter

The delimiter used to separate the names of the load balancers and elements in the same table cell. The default value is a comma (,).

colDelimiter

The delimiter that separates the fields in a description in the result string. The default is a semicolon (;).

rowDelimiter

The delimiter that separates two descriptions in the output string. The default is the line separator for the current operating system.

auxDelimiter

The delimiter used to separate members of elements which are lists in a table cell. The default is '|'.

The operation returns the following:

requestId

The ID of the request you have sent to Amazon by using this operation.

returnResult

The description in a table having the following columns: **loadBalancerName**, **createdTime**, **dnsName**, **interval**, **target**, **healthyThreshold**, **timeout**, **unhealthyThreshold**, a list of **listenerDescriptions** (policy names, protocol, load balancer port, and instance port), a list of **instances**, a list of **AppCookieStickinessPolicies** (policyName, cookieName), a list of **LBCookieStickinessPolicies** (policyName, cookieExpirationPeriod), and a list of **availability zones**.

Notes:

- Here is an example of **returnResult** where all delimiters have their default value:

```
loadBalancerName:TestBal01;createdTime:2010-08-13T05:34:57.890Z;dnsName:TestBal01-1103237715.us-east-1.elb.amazonaws.com;interval:60;target:TCP:80;healthyThreshold:3;timeout:15;unhealthyThreshold:6;activePolicyNames.1:null,protocol.1:HTTP,loadBalancerPort.1:80,instancePort.1:80;instances:i-33e07e59;policyName.1:pol,cookieName.1:cook;policyName.1:bbpol,cookieExpirationPeriod.1:0;availabilityZones:us-east-1a,us-east-1c
```

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is …." message is returned.

Disable Availability Zones For Load Balancer

The **Disable Availability Zones For Load Balancer** operation removes the specified EC2 Availability Zones from the load balancer.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name associated with the load balancer.

availabilityZones

The list of Availability Zones to be removed from the load balancer.

delimiter

The delimiter for the list of availability zones. The default is a comma (,).

The operation returns the following:

returnResult

A list of updated Availability Zones for the load balancer.

requestId

The ID of the request you have sent to Amazon by using this operation.

Notes:

- The operation completes successfully even if the Availability Zone you want to remove is not enabled for the load balancer.
- There must be at least one Availability Zone registered with a load balancer at all times. You cannot remove all the Availability Zones from a load balancer.
- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is ...." message is returned.

Enable Availability Zones For Load Balancer

The **Enable Availability Zones For Load Balancer** operation enables one or more EC2 Availability Zones for the load balancer.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name associated with the load balancer.

availabilityZones

The list of new Availability Zones for the load balancer. The Availability Zones must be in the same Region as the load balancer.

delimiter

The delimiter for the list of Availability Zones. The defaults are a comma (,).

The operation returns the following:

returnResult

The list of new Availability Zones for the load balancer.

requestId

The ID of the request you have sent to Amazon by using this operation.

Notes:

- The operation completes successfully even if the Availability Zone you want to add is already enabled for the load balancer.
- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is ...." message is returned.

Register Instances With Load Balancer

The **Register Instances With Load Balancer** operation adds new instances to the specified load balancer. Any instance that is not in any of the Availability Zones registered for the load balancer is moved to the **OutOfService** state. When the Availability Zone is added to the Load Balance, the instance is moved to the **InService** state.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

instanceIds

The IDs of the instances to be registered with the load balancer.

loadBalancerName

The name of the load balancer.

delimiter

The delimiter used to separate the instance IDs. The default is a comma (,).

The operation returns the following:

requestId

The ID of the request you have sent to Amazon by using this operation.

returnResult

A list of all instance IDs registered with the load balancer, separated by the delimiter.

Notes:

- Success does not guarantee that the operation has completed. Rather, it means that the request has been registered and that the changes will happen shortly.
- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is ...." message is returned.

Set Load Balancer Policies Of Listener

The **Set Load Balancer Policies Of Listener** operation associates, updates, or disables a policy with a listener on the load balancer. A listener can have only one or zero associated policies.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

loadBalancerName

The name of the load balancer.

loadBalancerPort

The external port of the load balancer with which this policy has to be associated.

policyNames

The list of policies to be associated with the listener. The list can have at most one policy. If the list is empty, the current policy is removed from the listener.

The operation returns the following:

requestId

The ID of the request you have sent to Amazon by using this operation.

returnResult

Returns **true** if the operation completes successfully; otherwise an error message is returned.

Notes:

- Like all Amazon operations, the results might not be visible instantly.
- Since all Amazon requests must include a timestamp, the RAS machine system clock should be properly set, otherwise a "Request has expired. Timestamp date is …." message is returned.

Troubleshooting

This section provides troubleshooting procedures and tools you can use to solve problems you may encounter while using this integration. It also includes a list of the error messages you may receive while using the integration and offers descriptions and possible fixes for the errors.

General troubleshooting procedures and tools

When troubleshooting issues related to the Elastic Load Balancing integration occur, there are several steps that you should verify:

1. Verify that the same operation works when it is performed through the EC2 console using the same parameters.
2. Verify that your access key and proxy parameters are correct.
3. Verify that the RAS machine system clock is properly set since all Amazon queries must include a timestamp.

Error codes

This section lists the error codes you may receive while using this integration. Each error code includes possible causes and fixes for the error.

CertificateNotFound

The specified SSL ID does not refer to a valid SSL certificate in the AWS Identity and Access Management Service.

DuplicateLoadBalancerName

The load balancer name already exists for this account. Please choose another name.

DuplicatePolicyName

A policy with the same name exists for this load balancer. Please choose another name.

Failed to open HTTP connection; failed to GET resource at ...

The HTTP connection to AWS cannot be established (for instance, because of an incorrect value for the **serviceEndpoint**, **proxyHost**, or **proxyPort** input).

IncompleteSignature

The request signature does not conform to AWS standards.

InternalFailure

The request processing has failed due to some unknown error, exception, or failure.

InvalidAction

The action or operation requested is invalid.

InvalidAvailabilityZone

The **availabilityZones** input value is invalid.

InvalidClientTokenId

The X.509 certificate or AWS Access Key ID provided does not exist in our records.

InvalidConfigurationRequest

The requested configuration change is invalid.

InvalidInstance

The specified EndPoint is not valid.

InvalidParameterCombination

Parameters that must not be used together were used together.

InvalidParameterValue

A bad or out of range value was supplied for the input parameter.

InvalidQueryParameter

The AWS query string is malformed; it does not adhere to AWS standards.

ListenerNotFound

The load balancer does not have a listener configured at the given port.

LoadBalancerNotFound

The specified load balancer could not be found.

MalformedQueryString

The query string is malformed.

MissingAction

The request is missing an action or operation parameter.

MissingAuthenticationToken

The request must contain either a valid (registered) AWS Access Key ID or X.509 certificate.

MissingParameter

An input parameter that is mandatory for processing the request is not supplied.

OptInRequired

The AWS Access Key ID needs a subscription for the service.

PolicyNotFound

One or more specified policies were not found.

RequestExpired

The request is past its expiration date or the request date (either with 15 minute padding), or the request date occurs more than 15 minutes in the future.

ServiceUnavailable

The request has failed due to a temporary failure of the server.

SignatureDoesNotMatch

The calculated request signature does not match the signature you provided. Check your AWS Secret Access Key and signing method. Consult the service documentation for details.

Throttling

The request was denied due to request throttling.

TooManyLoadBalancers

The quota for the number of load balancers has already been reached.

TooManyPolicies

The quota for number of policies for this load balancer has already been reached.

With the load balancing operations, **NumberFormatExceptions** may occur when converting the values of some inputs to numeric types. For instance, the **Configure Health Check** operation may fail with the `The following input must be a valid integer: interval` or `Proxy port is not a valid integer` if the specified inputs were not assigned integer numeric values.

Security

This section describes how security is handled by the Amazon Elastic Load Balancing integration.

The Elastic Load Balancing integration uses the Elastic Load Balancing Query API. To perform its task, each Elastic Load Balancing operation sends a query to Amazon. This query is sent over HTTPS and the Symphony Client library is used for sending the actual requests. Every Elastic Load Balancing operation has two inputs: **accessKey** and **accessKeyId**. These are provided by Amazon for every Amazon account.

According to Amazon, in addition to the name of the action and the list of parameters, you must include a signature in every Query request. The signature is created by using the **accessKey** provided by the user. The steps for creating a signature are described on the Amazon Web site at *http://docs.amazonwebservices.com/AWSEC2/2009-04-04/DeveloperGuide/index.html?using-query-api.html*, and are implemented by the operations.

The Elastic Load Balancing integration uses version 2 of the signature. For calculating an RFC 2104-compliant HMAC with the query string created by using the input parameters, the integration uses the secret Access Key as the key, and SHA1 as the hash algorithm. You should not perform any special configuration; just provide the **accessKey** and **accessKeyId**.

Tools

Following are OO tools that you can use with the Amazon Elastic Load Balancing integration:

- **RSFlowInvoke.exe** and **JRSFlowInvoke.jar**

  RSFlowInvoke (RSFlowInvoke.exe or the Java version, JRSFlowInvoke.jar) is a command-line utility that allows you to start a flow without using Central (although the Central service must be running). RSFlowInvoke is useful when you want to start a flow from an external system, such as a monitoring application that can use a command line to start a flow.

- **Web Services Wizard (wswizard.exe)**

  When you run the Web Services Wizard, you provide it with the WSDL for a given Web service. The WSDL string you provide as a pointer can be a file’s location and name or a URL. The Web Services Wizard displays a list of the methods in the API of the Web service that you specify. When you run the wizard, pick the methods you want to use, and with one click for each method you have selected, the wizard creates an HP OO operation that can execute the method. This allows you to use the Web Services Wizard to create operations from your monitoring tool’s API.

These tools are available in the OO home directory in the Studio/tools/ folder.