

HP Operations Orchestration

for the Windows and Linux operating systems

Software Version: 9.00.06

Amazon Elastic Compute Cloud Integration Guide

Document Release Date: October 2011

Software Release Date: October 2011



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2010-2011 Hewlett-Packard Development Company, L.P.

Trademark Notices

For information on open-source and third-party software acknowledgements, see *Open-Source and Third-Party Software Acknowledgements* (HPOO_OpenSrc_3rd-PartyAcks.pdf) in the documentation set for this release.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Web site at:

www.hp.com/go/hpsoftwaresupport

This Web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support Web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

1	Introduction	8
	Purpose of the Amazon Elastic Compute Cloud (EC2) Integration	9
	Supported Versions	9
	Downloading OO Releases and Documents on HP Live Network.....	9
2	Getting Started with the Amazon EC2 Integration	10
	Installing and Configuring the Integration	11
	Amazon EC2 — OO Integration Architecture	11
	Amazon EC2 Use Cases	11
3	Using the Amazon EC2 – OO Integration.....	15
	Location of Amazon EC2 Integration Operations and Flows	16
	Common Inputs in the Integration.....	17
	Descriptions of Amazon EC2 Operations and Flows.....	17
	Availability zones and regions	18
	Enumerate Availability Zones	18
	Enumerate Regions	19
	Elastic block store	20
	Snapshots	20
	Volumes	27
	Elastic IPs.....	33
	Allocate New Elastic IP.....	33
	Associate Elastic IP	34
	Disassociate Elastic IP	35
	Enumerate Elastic IPs	35
	Release Elastic IP	36
	Images.....	37
	Create Image.....	37
	Enumerate Images	38
	Get Image Details.....	40
	Modify Image Attribute.....	42
	Register Image	43
	Remove Image.....	44
	Reset Image Attribute	44
	Instances.....	45
	Describe Instances.....	45
	Get Instance Details.....	48
	Get System Log.....	50
	Modify Instance Attribute.....	50
	Monitor Instance.....	51
	Reset Instance Attribute	52
	Restart Instance	53

Run Instances	53
Run Instances From Template	57
Start Instance	58
Stop Instance	58
Terminate Instance	59
Unmonitor Instance.....	59
Key Pairs	60
Allocate New Key Pair	60
Delete Key Pair	61
Enumerate Key Pairs	61
Import Key Pair	62
Placement Groups	63
Create Placement Group	63
Delete Placement Group	64
Describe Placement Groups	64
Security groups.....	66
Allow Access To Security Group	66
Create Security Group.....	68
Delete Security Group	68
Describe Security Groups.....	69
Get Security Group Details.....	70
Revoke Access To Security Group	72
Spot Instances	74
Cancel Spot Instance Requests.....	74
Create Spot Datafeed Subscription	74
Delete Spot Datafeed Subscription.....	75
Describe Spot Datafeed Subscription.....	76
Describe Spot Instance Requests.....	77
Describe Spot Price History	79
Request Spot Instances	81
Windows.....	84
Bundle Windows Instance.....	84
Cancel Bundle Task.....	86
Enumerate Bundle Tasks	87
Get Bundle Task Details	88
EC2 Dashboard	89
Tags.....	90
Create Tags	90
Delete Tags.....	91
Enumerate Tags	92

4 Troubleshooting.....94

Troubleshooting Overview	95
General Troubleshooting Procedures and Tools	95
Error Messages	95

5 Security.....97

About Amazon EC2 Security	98
---------------------------------	----

6 OO Tools	99
OO Tools You Can Use with the Amazon EC2 Integration	100

1 Introduction

This section includes the following topics:

- [Purpose of the Amazon Elastic Compute Cloud \(EC2\) Integration](#)
- [Supported Versions](#)
- [Downloading OO Releases and Documents on HP Live Network](#)

Purpose of the Amazon Elastic Compute Cloud (EC2) Integration

With this integration, you can build HP Operations Orchestration (OO) flows that are integrated into the Amazon Elastic Compute Cloud (EC2).

The EC2 integration uses the EC2 Query API released on 08/31/2010 to integrate with OO. To use this integration successfully, you should have knowledge of the EC2 technology.

This document will explain how this integration has been implemented and how the operations included communicate between OO and EC2.



The Amazon EC2 operations provide support for the following regions: US East, US West, EU West, and Asia Pacific.

Supported Versions

Table 1 Supported Versions

Operations Orchestration Version	Amazon EC2 Version
9.00.06	EC2 API Version 2010-08-31

Downloading OO Releases and Documents on HP Live Network

HP Live Network provides an **Operations Orchestration Community** page where you can find and download supported releases of OO and associated documents.

To download OO releases and documents, visit the following site:

<https://www.www2.hp.com/>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

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.

2 Getting Started with the Amazon EC2 Integration

This section includes the following topics:

- [Installing and Configuring the Integration](#)
- [Amazon EC2 – OO Integration Architecture](#)
- [Amazon EC2 Use Cases](#)

Installing and Configuring the Integration

No special installation and configuration instructions are required for the Amazon EC2 integration. 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.

Amazon EC2 — OO Integration Architecture

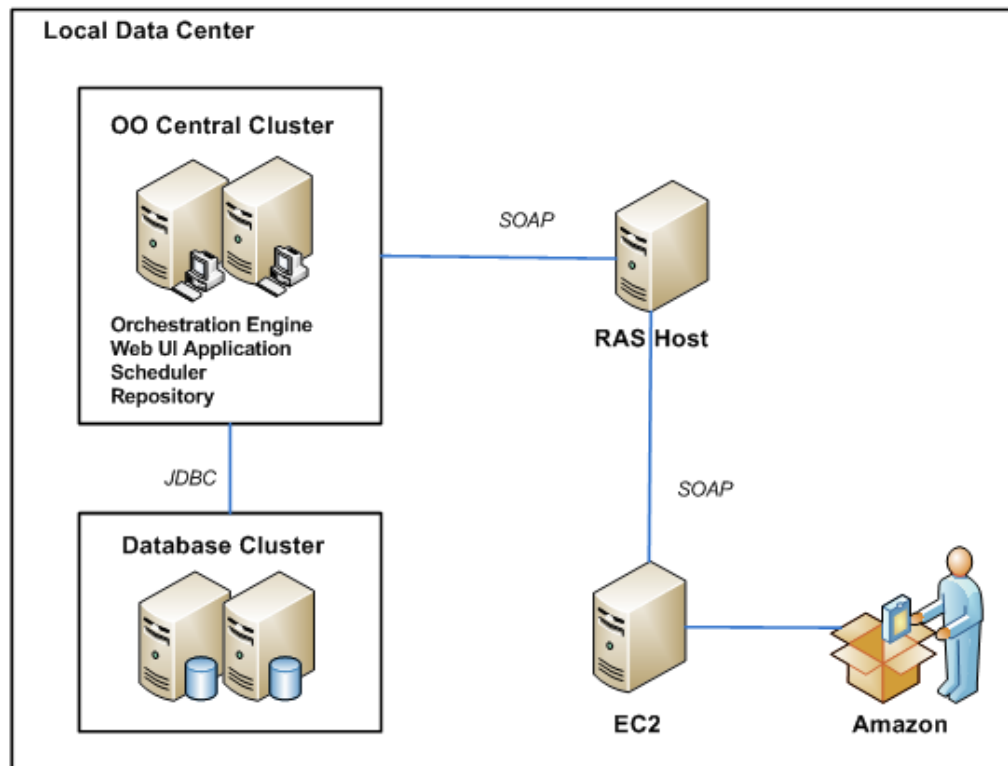


Figure 1 - Amazon EC2 - OO Integration Architecture

Amazon EC2 Use Cases

The following are the major use cases for the Amazon EC2 integration, and the operations and flows that you can use to implement them.

- 1 Enumerate availability zones and regions:
 - Enumerate Availability Zones
 - Enumerate Regions
- 2 Manage elastic block store:
 - Snapshots

- Create Shared Snapshot
 - Create Snapshot
 - Delete Snapshot
 - Describe Snapshots
 - Get Snapshot Details
 - Modify Snapshot Attribute
 - Reset Snapshot Attribute
- Volumes
 - Attach Volume
 - Create Volume
 - Delete Volume
 - Describe Volumes
 - Detach Volume
 - Get Volumes Details
- 3 Work with elastic IPs:
 - Allocate New Elastic IP
 - Associate Elastic IP
 - Disassociate Elastic IP
 - Enumerate Elastic IPs
 - Release Elastic IP
- 4 Manage images:
 - Create Image
 - Enumerate Images
 - Get Image Details
 - Modify Image Attribute
 - Register Image
 - Remove Image
 - Reset Image Attribute
- 5 Manage instances:
 - Describe Instances
 - Get Instance Details
 - Get System Log
 - Modify Instance Attribute
 - Monitor Instance
 - Reset Instance Attribute
 - Restart Instance

- Run Instances
- Run Instances From Template
- Start Instance
- Stop Instance
- Terminate Instance
- Unmonitor Instance
- 6 Manage key pairs:
 - Allocate New Key Pair
 - Delete Key Pair
 - Enumerate Key Pairs
 - Import Key Pair
- 7 Work with cluster placement groups:
 - Create Placement Group
 - Delete Placement Group
 - Describe Placement Groups
- 8 Manage security groups:
 - Allow Access To Security Group
 - Create Security Group
 - Delete Security Group
 - Describe Security Groups
 - Get Security Group Details
 - Revoke Access To Security Group
- 9 Manage spot instances
 - Cancel Spot Instance Requests
 - Create Spot Datafeed Subscription
 - Delete Spot Datafeed Subscription
 - Describe Spot Datafeed Subscription
 - Describe Spot Instance Requests
 - Describe Spot Price History
 - Request Spot Instances
- 10 Manage Windows bundle tasks:
 - Bundle Windows Instance
 - Cancel Bundle Task
 - Enumerate Bundle Tasks
 - Get Bundle Task Details
- 11 Display the EC2 Dashboard information
 - EC2 Dashboard
- 12 Manage tags:

- Create Tags
- Delete Tags
- Enumerate Tags

3 Using the Amazon EC2 – OO Integration

This section includes the following topics:

- [Location of Amazon EC2 Integration Operations and Flows](#)
- [Common Inputs in the Integration](#)
- [Descriptions of Amazon EC2 Operations and Flows](#)

Location of Amazon EC2 Integration Operations and Flows

The EC2 integration includes the following operations in the OO Studio Library/Integrations/Amazon/EC2/ folder.

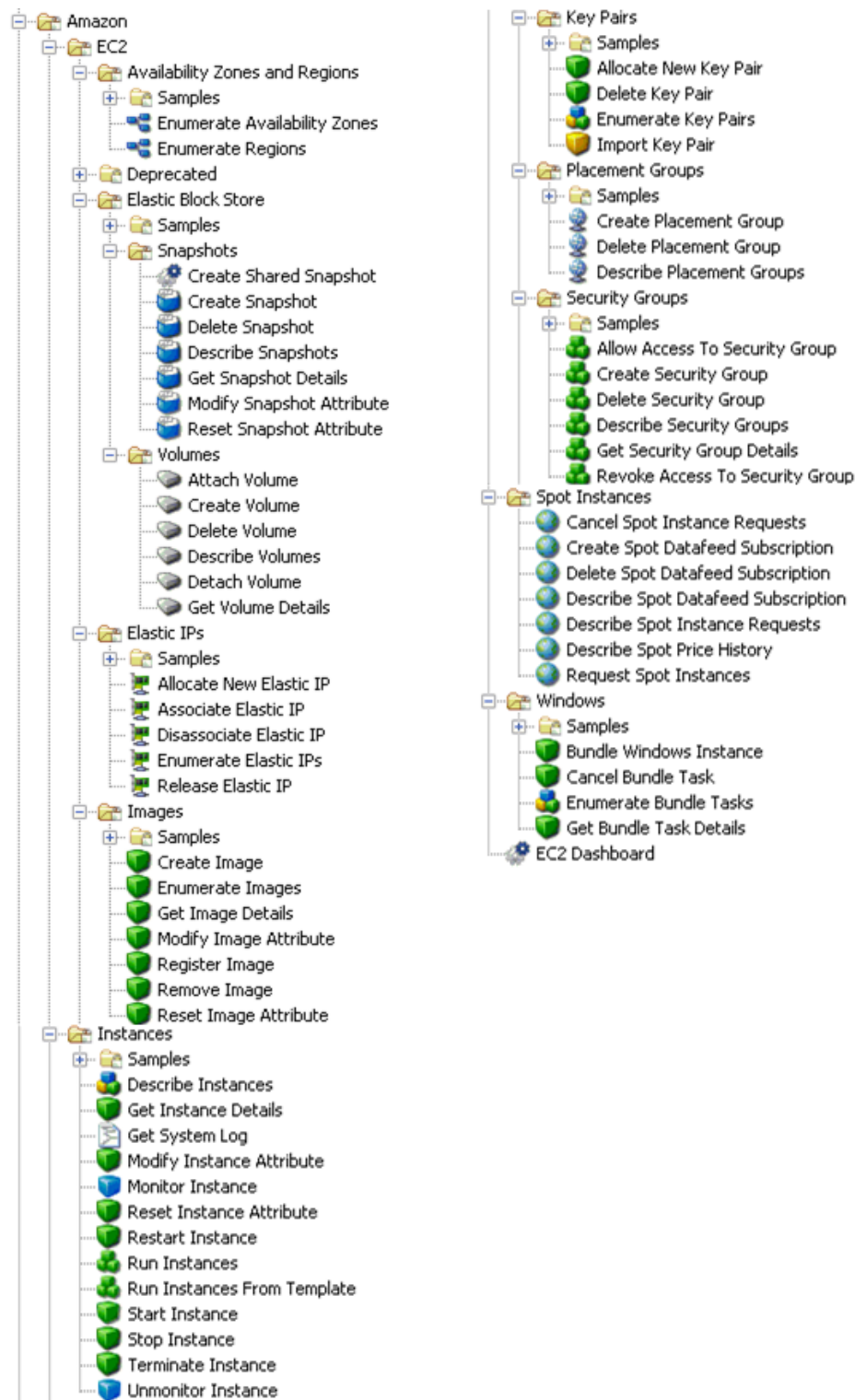


Figure 2 – Location of Amazon EC2 Integration Operations and Flows

Common Inputs in the Integration

OO flows and operations use inputs to specify how they obtain the data that they need and when the data is obtained. The following inputs are used consistently throughout the EC2 integration's operations and flows.

`accessKeyId`

The ID of the access key that uniquely identifies the user account. This can be obtained from the Amazon console.

`accessKey`

The secret access key that uniquely identifies the user. This can be obtained from the Amazon console.

`proxyHost`

The proxy server used to connect to Amazon.

`proxyPort`

The port to connect to the proxy server. You must specify a value for both the **proxyHost** and **proxyPort** inputs or leave both of them empty.

`proxyUsername`

A username to use when connecting to the proxy if it requires authentication.

`proxyPassword`

A password to use when connecting to the proxy if it requires authentication.

`serviceEndpoint`

The endpoint to which the requests are sent. The default value is **ec2.amazonaws.com** which corresponds to the United States region (another equivalent endpoint for the United States is **ec2.us-east-1.amazonaws.com**). The default value for Europe is **ec2.eu-west-1.amazonaws.com**. (These values may change in the future).

For the **Placement Groups** operations, the default is `ec2.amazonaws.com` which corresponds to the United States region. Currently, the **Placement Group** operations are available only in the US-East (Northern Virginia) region. This may change in the future.

Descriptions of Amazon EC2 Operations and Flows

This section describes the Amazon EC2 integration's operations and flows, including any operation- or flow-specific inputs. The flows and operations are grouped by their basic functionality:

- Availability zones and regions
- Elastic block store—snapshots and volumes
- Elastic IPs
- Images
- Instances

- Key pairs
- Cluster placement groups
- Security groups
- Spot instances
- Windows bundle tasks
- EC2 Dashboard
- Tags

The sample flows in the OO Library/Integrations/Amazon/EC2/ folder perform some of the most common tasks that need to be automated when using EC2, such as starting instances, allocating elastic IPs, and working with images. Each of these sample flows has a description that describes in detail what it does. You can use these flows as they are or as templates for new operations.

Availability zones and regions

Enumerate Availability Zones

The **Enumerate Availability Zones** operation retrieves a list of your Amazon EC2 availability zones.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

`filterName1`

The first filter name for the operation. The valid values are **message**, **region-name**, **state**, and **zone-state**.

`filterValues1`

A first value or values list for the corresponding filter name.

`filterName[number]`

To add additional filters, you must add extra **filterName** inputs using the convention **filterName[number]** where you replace **[number]** with increasing numbers (for example, **filterName2**, **filterName3**).

`filterValues[number]`

To add additional values, you must add extra **filterValues** inputs using the naming convention **filterValues[number]** where you replace **[number]** with increasing numbers (for example, **filterValues2**, **filterValues3**).

`delimiter`

The delimiter used for the results lists and the filter values list. The default is a comma (,).

Results

The operation returns the following results:

`requestId`

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

`returnResult`

A delimiter-separated list of your availability zones.

`notAvailableZones`

A delimiter-separated list of availability zones which are not available for use, but are visible to you (this list usually has no elements).



- Like all Amazon EC2 operations, the results might not be visible instantly.
- Availability zones are not the same across accounts. The availability zone **us-east-1a** for account A is not necessarily the same as **us-east-1a** for account B.
- You can find more details about the filters that can be applied on this operation in the **DescribeAvailabilityZones** section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.

Enumerate Regions

The **Enumerate Regions** operation lists the regions that are currently available to you.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

`rowDelimiter`

The delimiter to place between the rows of the results table. The default is the line separator character for the current operating system.

`colDelimiter`

The delimiter to be placed between the columns of the results table. The default is a comma (,).

`filterName1`

The first filter name for the operation. The valid values are **endpoint** and **region-name**.

`filterValues1`

A first value or values list for the corresponding filter name.

`filterName[number]`

To add additional filters, you must add extra **filterName** inputs using the convention **filterName[number]** where you replace **[number]** with increasing numbers (for example, **filterName2**, **filterName3**).

`filterValues[number]`

To add additional values, you must add extra **filterValues** inputs using the naming convention **filterValues[number]** where you replace **[number]** with increasing numbers (for example, **filterValues2**, **filterValues3**).

`delimiter`

The delimiter used for the lists of volume IDs results and filter values. The default is a comma (,).

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The available regions in a table having the following columns: Region name and Region endpoint.

Examples:

eu-west-1,ec2.eu-west-1.amazonaws.com

us-east-1,ec2.us-east-1.amazonaws.com



- Like all Amazon EC2 operations, the results might not be visible instantly.
- You can find more details about the filters that can be applied on this operation in the **DescribeRegions** section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.

Elastic block store

Snapshots

Create Shared Snapshot

The **Create Shared Snapshot** flow creates a snapshot of an EBS volume and makes the snapshot public or grants permission to it for a given user. The flow returns the ID of the generated snapshot. If you enter a value of all for the groups input, the snapshot is shared publicly. Otherwise, the snapshot is private and only the developers whose account numbers are supplied through the `userIds` input are granted access to it.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`volumeId`

The ID of the volume from which the snapshot is generated.

userIds

A list of the users to whom you grant access. This list can only contain IDs, which must be numeric.

groups

A list of the groups to which you grant access (according to Amazon, currently the only supported value is all; other values may be supported in the future).

delimiter

The delimiter for the elements of the inputs that contain lists.

Results

Theflow returns the following results:

returnResult

The ID of the generated snapshot.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- At least one of the userIds or groups inputs should not be empty.

Create Snapshot

The **Create Snapshot** operation creates a snapshot of an EBS volume and stores it in S3. It returns the ID of the generated snapshot.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

volumeId

The ID of the volume from which to generate the snapshot.

snapshotDescription

The description of the snapshot to create.

Results

The operation returns the following results:

returnCode

The ID of the generated snapshot.

requestId

The Amazon request ID.

volumeId

The ID of the volume from which the snapshot is generated.

`status`

The snapshot state.

`startTime`

The timestamp when the snapshot was initiated.

`progress`

The progress of the snapshot, in percentage.

`ownerId`

The ID of the snapshot's owner.

`volumeSize`

The size of the volume from which the snapshot is generated.

`description`

The description of the snapshot.



Like all Amazon EC2 operations, the results might not be visible instantly.

Delete Snapshot

The **Delete Snapshot** operation deletes a snapshot of an Amazon EBS volume. It returns **true** on success and **false** on failure.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`snapshotId`

The ID of the snapshot to be deleted.

Results

The operation returns the following results:

`returnCode`

Returns **true** for success and **false** for failure.

`requestId`

The Amazon request ID.



Like all Amazon EC2 operations, the results might not be visible instantly.

Describe Snapshots

The **Describe Snapshots** operation describes the status of Amazon EBS snapshots. If you provide a list of snapshot IDs, only those snapshots are described. Otherwise, all of the existing snapshots are described.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

snapshotIds

An optional list of snapshot IDs. If you provide this list, only these snapshots are described.

filterName1

The first filter name for the operation. The valid values are:

- description
- owner-alias
- owner-id
- progress
- snapshot-id
- start-time
- status
- tag-key
- tag-value
- volume-id
- volume-size
- tag:<tag-name> where <tag-name> is the name of a tag that a snapshot may have. See the notes below for an example of filtering the snapshot based on a specific tag/value combination.

filterValues1

A first value or values list for the corresponding filter name.

filterName[number]

To add additional filters, you must add extra **filterName** inputs using the convention **filterName[number]** where you replace **[number]** with increasing numbers (for example, **filterName2**, **filterName3**).

filterValues[number]

To add additional values, you must add extra **filterValues** inputs using the naming convention **filterValues[number]** where you replace **[number]** with increasing numbers (for example, **filterValues2**, **filterValues3**).

delimiter

The delimiter to place between the elements of the **snapshotIds**, **snapshotOwners**, and **restorableBy** inputs. The default is a comma (,).

rowDelimiter

The delimiter to place between the rows of the output table. The default is the line separator of the current operating system.

colDelimiter

The delimiter to place between the columns of the output table. The default is a comma (,).

snapshotOwners

The owner of the snapshots to return. You can specify multiple owners using the delimiter value specified in the delimiter input. The valid values are **self**, **amazon**, and **AWS Account ID**.

restorableBy

A list of restorable users separated by the delimiter specified in the delimiter input. If you specify values for this input, only users who have create snapshot permissions for the snapshots are returned.

Results

The operation returns the following results:

returnResult

A table containing the following columns: the snapshot ID, the ID of the volume from which the snapshot was created, the volume size, the snapshot status (pending, completed, or error), the owner ID, the owner alias, description, the timestamp when the snapshot was initiated, and the progress of the snapshot creation process, in percentage. Each row in the table corresponds to a snapshot. If a column's result is empty, 'null' is returned instead.

requestId

The Amazon request ID.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- To list the resources assigned tag Purpose=X or Purpose=Y specify:
filterName1=tag:Purpose
filterValues1=X,Y
- You can find more details on the filters that can be applied on this operation in the DescribeSnapshots section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.
- If you assign a value of start-time to a **filterName** input, the timestamps supplied to the corresponding **filterValues** must have the local timezone and must use the format yyyy/MM/dd HH:mm:ss. In this case, you cannot use wildcards for **filterValues**.

Get Snapshot Details

The **Get Snapshot Details** operation describes the status of an EBS snapshot.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

snapshotId

The ID of the snapshot to be described.

delimiter

The delimiter for the elements of the results that are lists.

Results

The operation returns the following results:

requestId

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

returnResult

The snapshot identifier.

snapshotId

The snapshot identifier.

status

The status of the snapshot (for example, pending, completed, or error).

volumeId

The ID of the volume from which the snapshot was created.

startTime

The timestamp when the snapshot was initiated.

progress

The progress of the snapshot creation process, in percentage.

ownerId

The AWS account ID of the user who owns the required snapshot.

ownerAlias

The AWS account alias or AWS account ID that owns the AMI (for example, amazon, redhat, or self.)

description

The description of the snapshot.

tags

The tags assigned to the resource. Each tag is present in the format tagKey=tagValue in the returned list (for example, Owner=DbAdmin,Stack=Production).



Like all Amazon EC2 operations, the results might not be visible instantly.

Modify Snapshot Attribute

The **Modify Snapshot Attribute** operation modifies the (permissions) attribute of a snapshot. You can make your snapshots public or grant permission to them for a given user.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

snapshotId

The ID of the snapshot you want to modify.

attribute

The attribute to modify. Currently, the only attribute you can modify is `createVolumePermission`.

operationType

The operation to perform on the attribute. The valid values are `add` and `remove`. You should specify this input only if the attribute input is set to `createVolumePermission`.

userIds

A list of users you want to grant to or revoke access from. This list can only contain IDs that are numeric. You should only specify this input if the attribute input is set to `createVolumePermission`.

groups

A list of the groups you want to grant to or revoke access from. According to Amazon, the only currently supported value is `all`. Other values may be supported in the future. You should specify this input only if the attribute input is set to `createVolumePermission`.

delimiter

The delimiter for the elements of the inputs that are lists.

Results

The operation returns the following results:

requestId

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

returnResult

Returns true if the operation completed successfully or false if it fails.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- If the attribute input is set to `createVolumePermission`, the `operationType` input and at

least one of the `userIds` or `groups` inputs should not be empty.

Reset Snapshot Attribute

The **Reset Snapshot Attribute** operation resets an attribute of a snapshot to its default value.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`snapshotId`

The ID of the snapshot you want to modify.

`attribute`

The attribute to reset. The only currently supported value is **createVolumePermission**. Other values may be supported in the future.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true if the operation completed successfully or false if it fails.



- Like all Amazon EC2 operations, the results might not be visible instantly.

Volumes

Attach Volume

The **Attach Volume** operation attaches an Amazon EBS volume to a running instance and exposes it as the specified device.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`volumeId`

The ID of the Amazon EBS volume. The volume must not be attached to any instance.

`instanceId`

The ID of the instance to which the volume attaches. The volume and instance must be within the same availability zone, and the instance must be running.

deviceName

Specifies how the device is exposed to the instance (for example, `/dev/sdh`, `/dev/sdi`, `xvdh`, or `xvdi`).

Results

The operation returns the following results:

requestId

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

returnResult

The status of the attaching process (for example, **attaching**).

attachTime

The time the process was launched.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- The Windows devices are xvda through xvdv. xvda and xvdb are reserved by the operating system and xvdc is assigned to drive C:\. For some instance types, devices xvdd through xvde might also be reserved by the instance stores.ptv.
- The Linux and UNIX devices are /dev/sdf through /dev/sdl. The devices /dev/sda1, /dev/sda2, /dev/sdb, /dev/sdc, /dev/sdd, and /dev/sde are reserved.
- If the response of the operation is attaching, this does not guarantee that the volume attached. Use the Get Volume Details operation to check the status of the volume. If the volume has not attached it may be because you specified an invalid device name.

Create Volume

The **Create Volume** operation creates a new Amazon EBS volume.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

size

A positive integer that represents the size of the volume, in GBs. This input is required if you are not creating a volume from a snapshot.

snapshotId

The snapshot from which to create the new volume. This input is required if you do not specify the size input.

availabilityZone

The availability zone in which to create the new volume.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The ID of the newly created volume.

`volumeId`

The ID of the newly created volume.

`size`

The size of the newly created volume, in GBs.

`status`

The volume state (for example, creating, available, in-use, deleting, or error).

`createTime`

The timestamp when volume creation was initiated.

`availabilityZone`

The availability zone in which the volume was created.

`snapshotId`

The snapshot from which the volume was created, if applicable.



Like all Amazon EC2 operations, the results might not be visible instantly.

Delete Volume

The **Delete Volume** operation deletes an Amazon EBS volume.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`volumeId`

The ID of the Amazon EBS volume. The volume must not be attached to any instance.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The status of the process (for example, deleting).



Like all Amazon EC2 operations, the results might not be visible instantly.

Describe Volumes

The **Describe Volumes** operation describes the status of your Amazon EBS volumes. If you specify a list of volume IDs, only those volumes are described. Otherwise, all volumes that you own are described.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`volumes`

An optional list of volume IDs. If you specify a list of volume IDs, only those volumes will be described.

`filterName1`

The first filter name for the operation. The valid values are `attachment.attach-time`, `attachment.delete-on-termination`, `attachment.device`, `attachment.instance-id`, `attachment.status`, `availability-zone`, `create-time`, `size`, `snapshot-id`, `status`, `tag-key`, `tag-value`, `volume-id`, and `tag:<tag-name>` where `<tag-name>` stands for the name of a tag that a volume may have. See the notes below for an example of filtering the volumes based on a specific tag/value combination.

`filterValues1`

A first value/values list for the corresponding filter name.

`filterName[number]`

To add additional filters, you must add extra inputs using the following naming convention: `filterName[number]` where you replace `[Number]` with increasing numbers (for example, `filterName2`, `filterName3`).

`filterValues[number]`

To add additional values, you must add extra inputs using the following naming convention: `filterValues[number]` where you replace `[Number]` with increasing numbers (for example, `filterValues2`, `filterValues3`).

`delimiter`

The delimiter for the list of volume IDs and the list of filter values. The default is a comma (,).

`rowDelimiter`

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

colDelimiter

The delimiter used between columns in the output table. The default is a comma (,).

Results

The operation returns the following results:

requestId

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

returnResult

A table with the following columns: the volume ID, the volume size, the ID of the snapshot from which the volume was created, the availability zone, the volume's status (for example, creating, available, in-use, deleting, or error), and the attachment status (for example, attaching, attached, or detached). Each row corresponds to a different volume. If a column is empty, the result is null.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- To list the resources assigned tag Purpose=X or Purpose=Y, specify:
filterName1=tag:Purpose
filterValues1=X,Y
- You can find more details on the filters that can be applied on this operation in the DescribeVolumes section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.
- If you assign a value to a **filterName** input of **attachement.attach-time** or **create-time**, the timestamps supplied to the corresponding **filterValues** must have the local timezone and be in the format yyyy/MM/dd HH:mm:ss. In this situation, you cannot use wildcards for filterValues.

Detach Volume

The **Detach Volume** operation detaches an Amazon EBS volume from an instance.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

volumeId

The ID of the Amazon EBS volume to be detached.

instanceId

The ID of the instance to which the volume is attached.

deviceName

Specifies how the device is exposed to the instance (for example, /dev/sdh, /dev/sdi, xvdh, or xvdi).

force

Set to true if a previous detachment attempt did not occur cleanly. This option can lead to data loss or a corrupted file system. The default value is false. If you leave this input blank, it uses the default value.

Results

The operation returns the following results:

requestId

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

returnResult

The status of the process (for example, **detaching**).

attachTime

The time the association was created.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- The Windows devices are xvda through xvdv. xvda and xvdb are reserved by the operating system, and xvdc is assigned to drive C:\.
- The Linux and UNIX devices are /dev/sdf through /dev/sdl. The devices /dev/sda1, /dev/sda2, /dev/sdb, /dev/sdc, /dev/sdd, and /dev/sde are reserved.

Get Volume Details

The **Get Volume Details** operation describes the status of an EBS volume.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

volumeId

The ID of the volume to describe.

rowDelimiter

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

colDelimiter

The delimiter used between columns in the attachmentSet output table. The default is a comma (,).

delimiter

The delimiter for the elements of results that are lists. The default value is a comma (,).

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The volume ID.

`volumeId`

The volume ID.

`size`

The volume size.

`snapshotId`

The snapshot from which the volume was created. This field may be empty.

`availabilityZone`

The availability zone in which the volume was created.

`status`

The volume state (for example, creating, available, in-use, deleting, or error).

`createTime`

The timestamp from when the volume was created. This is converted to the RAS machine timezone.

`attachmentSet`

A table that lists the instances to which the volume is attached. Currently, you can attach a volume to just one instance, but this result is in a table in case this changes in the future. The table has the following columns: the instance ID, the device through which the volume is exposed to the instance (for example, `/dev/sdf` on a Linux system or `xvdf` on a Windows system), the attachment state, the timestamp when the volume was attached, and the volume ID. Each row corresponds to a different instance.

`tags`

The tags assigned to the resource. Each tag is present in the format `tagKey=tagValue` in the returned list (for example, `Owner=DbAdmin,Stack=Production`).



Like all Amazon EC2 operations, the results might not be visible instantly.

Elastic IPs

Allocate New Elastic IP

The **Allocate New Elastic IP** operation acquires an elastic IP address for use with your Amazon EC2 account.

Inputs

All of the operation's inputs are described in [Common Inputs in the Integration](#)

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Your new elastic IP address.



Like all Amazon EC2 operations, the results might not be visible instantly.

Associate Elastic IP

The **Associate Elastic IP** operation associates an elastic IP address with an instance. If the IP address is currently assigned to another instance, it is reassigned to the new instance. The old instance will not have a public DNS name for a few minutes. If you call this operation more than once with the same inputs, Amazon EC2 does not return an error.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`ipAddress`

The elastic IP address to use for your instance.

`instanceId`

The ID of the instance.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true if the operation completes successfully.



Like all Amazon EC2 operations, the results might not be visible instantly.

Disassociate Elastic IP

The **Disassociate Elastic IP** operation disassociates the specified elastic IP address from the instance to which it is assigned. If the IP address is not currently assigned to an instance, the operation succeeds but does nothing. The instance to which the address was associated will not have a public DNS name for a few minutes.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

ipAddress

The elastic IP address you want to disassociate.

Results

The operation returns the following results:

requestId

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

returnResult

Returns true if the operation completes successfully.



Like all Amazon EC2 operations, the results might not be visible instantly.

Enumerate Elastic IPs

The **Enumerate Elastic IPs** operation lists the elastic IPs addresses assigned to your account.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

rowDelimiter

The row delimiter in the results table. The default value is the line separator for the current operating system.

colDelimiter

The column delimiter in the results table. The default value is a comma (,).

filterName1

The first filter name for the operation. The valid values are instance-id and public-ip.

filterValues1

A first value/values list for the corresponding filter name.

`filterName[number]`

To add additional filters, you must add extra inputs using the following naming convention: `filterName[number]` where you replace `[number]` with increasing numbers (for example, `filterName2`, `filterName3`).

`filterValues[number]`

To add additional values, you must add extra inputs using the following naming convention: `filterValues[number]` where you replace `[number]` with increasing numbers (for example, `filterValues2`, `filterValues3`).

`delimiter`

The delimiter for the list of filter values. The default is a comma (,).

Results

The flow returns the following results:

`requestId`

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

`returnResult`

Your elastic IPs in a table with the following columns: the IP Address and the Instance ID associated to that address.

Example:

174.129.17.27,i-c0ad46a8

174.129.17.39,null



- Like all Amazon EC2 operations, the results might not be visible instantly.
- You can find more details about the filters that can be applied on this operation in the `DescribeAddresses` section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.

Release Elastic IP

The **Release Elastic IP** operation releases an elastic IP address associated with your Amazon EC2 account. If you run this operation on an elastic IP address that you do not own (maybe you have released it already), the operation fails. Releasing an IP address automatically disassociates it from any instance with which it is associated. To disassociate an IP address without releasing it, use the **Disassociate Elastic IP** operation. The instance will not have a public DNS name for a few minutes.

Inputs

All of the operation's inputs except the following are described in *[Common Inputs in the Integration](#)*

`ipAddress`

The IP address to release from your account.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true if the operation completes successfully.



- Like all Amazon EC2 operations, the results might not be visible instantly.

Images

Create Image

The **Create Image** operation creates an AMI that uses an Amazon EBS root device from a running or stopped instance.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`instanceId`

The ID of the instance.

`imageName`

The name of the AMI you want to create.

`description`

The description of the AMI you want to create.

`noReboot`

By default this property is set to false, which means Amazon EC2 attempts to cleanly shut down the instance before image creation and reboots the instance afterwards. When set to true, Amazon EC2 does not shut down the instance before creating the image. When this option is used, file system integrity on the created image cannot be guaranteed.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The ID of the AMI you have created.



Like all Amazon EC2 operations, the results might not be visible instantly.

Enumerate Images

The **Enumerate Images** operation returns a list of the Amazon EC2 images that are accessible to you. Images include AMIs (Amazon Machine Images), AKIs (Amazon Kernel Images), and ARIs (Amazon RAM disk Images). You can filter the results to include only images:

- Of a certain platform
- Published by a particular user
- Based on the criterion that they are accessible only to you or that they are public

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`imageOwners`

A list of the IDs of the owners whose images you want to see (for example, 099720109477,3272169288991). If you leave this input empty, all of the images are returned.

`executableBy`

The accessibility of the images (for example, all or self).

`colDelimiter`

The column delimiter for the results table.

`rowDelimiter`

The row delimiter for the results table.

`filterName1`

The first filter name for the operation. The valid values are:

- architecture
- block-device-mapping.delete-on-termination
- block-device-mapping.device-name
- block-device-mapping.snapshot-id
- block-device-mapping.volume-size
- description
- image-id
- image-type
- is-public

- kernel-id
- manifest-location
- name
- owner-alias
- owner-id
- platform
- product-code
- ramdisk-id
- root-device-name
- root-device-type
- state
- state-reason-code
- state-reason-message
- tag-key
- tag-value
- virtualization-type
- tag:<tag-name> where <tag-name> is the name of a tag that an image may have. See the notes below for an example of filtering images based on a specific tag/value combination.

filterValues1

A first value or values list for the corresponding filter name.

filterName[number]

To add additional filters, you must add extra filterName inputs using the convention filterName[number] where you replace [number] with increasing numbers (for example, filterName2, filterName3).

filterValues[number]

To add additional values, you must add extra filterValues inputs using the naming convention filterValues[number] where you replace [number] with increasing numbers (for example, filterValues2, filterValues3).

delimiter

The delimiter for the items in the filter value lists and the elements of the results that are lists.

Results

The operation returns the following results:

requestId

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

`returnResult`

The images in a table with the following columns: AMI ID, AMI name, Manifest Location, Description, Visibility, Architecture, OwnerID, Owner Alias Name, Image Type, Image State, Kernel ID, Ramdisk ID, Platform, State Reason Name, Root Device Name, and Root Device Type.

Example:

```
aki-00806369,null,karmic-kernel-zul/ubuntu-kernel-2.6.31-300-ec2-i386-20091001-test-04.manifest.xml,null,true,i386,099720109477,null,available,kernel,null,null,null,null,null,instance-store
aki-00896a69,null,karmic-kernel-zul/ubuntu-kernel-2.6.31-300-ec2-i386-20091002-test-04.manifest.xml,null,true,i386,099720109477,null,available,kernel,null,null,null,null,null,instance-store
aki-008b6869,null,redhat-cloud/RHEL-5-Server/5.4/x86_64/kernels/kernel-2.6.18-164.x86_64.manifest.xml,null,true,x86_64,432018295444,null,available,kernel,null,null,null,null,null,instance-store
aki-00f41769,null,karmic-kernel-zul/ubuntu-kernel-2.6.31-301-ec2-i386-20091012-test-06.manifest.xml,null,true,i386,099720109477,null,available,kernel,null,null,null,null,null,instance-store
```



- Like all Amazon EC2 operations, the results might not be visible instantly.
- Visibility is true if the image is public and false if it only has implicit and explicit launch permissions. Platform is not always specified.
- To list the resources assigned tag Purpose=X or Purpose=Y, specify:
filterName1=tag:Purpose
filterValues1=X,Y
- You can find more details on the filters that can be applied on this operation in the DescribeImages section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.

Get Image Details

The **Get Image Details** operation returns details about a particular Amazon image. The image can be AMI (Amazon Machine Image), AKI (Amazon Kernel Image), or ARI (Amazon RAM disk Image).

Inputs

All of the operation's inputs except the following are described in *[Common Inputs in the Integration](#)*

imageId

The ID of the image you want details about. If you specify a nonexistent ID, the operation fails.

delimiter

The delimiter for the elements of the result which are lists. The default value is a comma (,).

Results

The operation returns the following results:

returnResult

The image ID of the Amazon image you want details about.

name

The name of the Amazon image.

description

The description of the Amazon image.

imageLocation

The location of the image.

kernelId

The operating system kernel associated with the image (only for AMIs, this is optional).

ramdiskId

The RAM disk associated with the image (only for AMIs, this is optional).

platform

The operating system platform (optional).

architecture

The architecture of the image (for example, i386 or x86_64).

ownerId

The account number of the image owner.

status

The status of the image. If the operation returns available, the image is successfully registered and available for launching. If the operation returns deregistered, the image is deregistered and no longer available for launching.

imageType

The type of the image (for example, machine, kernel, or ramdisk).

isPublic

Returns true if this image has public launch permissions and false if it only has implicit and explicit launch permissions.

productCodes

The product codes of the image.

rootDeviceType

The root device type used by the AMI. The AMI can use an Amazon EBS or instance store root device.

rootDeviceName

The root device name (for example, /dev/sda1).

stateReason

The message for the reason for the state change.

requestId

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

tags

The tags assigned to the resource. Each tag is present in the format tagKey=tagValue in the returned list (for example, Owner=DbAdmin,Stack=Production).



Like all Amazon EC2 operations, the results might not be visible instantly.

Modify Image Attribute

The **Modify Image Attribute** operation modifies an attribute of an AMI (Amazon Machine Image). This allows you to make your AMIs public, grant permission to an AMI for a given user, or associate a product code with an AMI.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

amild

The ID of the AMI you want to modify.

attribute

The attribute to modify (the valid values are launchPermission and productCode).

operationType

The operation to perform (the valid values are add and remove). Specify this input only if the attribute input is set to launchPermission.

userIds

A list of users to whom you want to grant or revoke access. The list can only contain numeric IDs. Specify this input only if the attribute input is set to launchPermission.

groups

The groups to which you want to grant or revoke access (according to Amazon, the only currently supported value is all; other values may be supported in the future). Specify this input only if the attribute input is set to launchPermission.

productCode

The product code to associate with your AMI. Specify this input only if the attribute input is set to productCode. According to Amazon, each AMI can have a single product code associated with it. The user must be the owner of the AMI to associate a product code with it.

delimiter

The delimiter for the elements of the inputs that are lists.

Results

The operation returns the following results:

requestId

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

returnResult

Returns true if the operation completes successfully and false if it does not.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- If the attribute input is set to productCode, the productCode input cannot be empty.
- If the attribute input is set to launchPermission, the operationType input and at least one of the userIds or groups inputs should not be empty.
- An Amazon DevPay account is needed in order to be able to change the productCode.
- **Important:** Once set, you cannot change or remove the productCode.

Register Image

The **Register Image** operation registers an AMI with Amazon EC2. Images must be registered before they can be launched. You can register an image more than once, and have multiple AMI IDs for the same image at the same time.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

imageLocation

The full path to your AMI manifest in Amazon S3 storage (for example, /mybucket/myimage.manifest.xml).

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The unique ID of the newly registered machine image (an AMI ID).



Like all Amazon EC2 operations, the results might not be visible instantly.

Remove Image

The **Remove Image** operation deregisters the specified Amazon Machine Image (AMI). Once deregistered, the AMI cannot be used to launch new instances. The image is not deleted from Amazon S3.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`amild`

The ID of the AMI you want to deregister.

Results

The flow returns the following results:

`requestId`

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

`returnResult`

Returns true if the operation completes successfully.



Like all Amazon EC2 operations, the results might not be visible instantly.

Reset Image Attribute

The **Reset Image Attribute** operation resets an attribute of an Amazon Machine Image (AMI) to its default value.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`amild`

The ID of the AMI you want to modify.

`attribute`

The attribute to reset. Currently, the only valid value is `launchPermission`.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true if the operation completes successfully and false if it does not.



Like all Amazon EC2 operations, the results might not be visible instantly.

Instances

Describe Instances

The Describe Instances operation returns a list of your Amazon EC2 instances.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`instanceIds`

A list of the instance IDs to describe. If you leave this input empty, all instances are described. The instances are returned in the order that the IDs are given in this input. If you leave this input empty, the instance IDs are returned in the order the instances are stored by Amazon EC2; this is the same order in which the inputs can be seen in the user interface provided by Amazon.

`filterName1`

The first filter name for the operation. The valid values are:

- `architecture`
- `availability-zone`
- `block-device-mapping.attach-time`
- `block-device-mapping.delete-on-termination`
- `block-device-mapping.device-name`
- `block-device-mapping.status`
- `block-device-mapping.volume-id`
- `client-token`
- `dns-name`
- `group-id`

- image-id
- instance-id
- instance-lifecycle
- instance-state-code
- instance-state-name
- instance-type
- ip-address
- kernel-id
- key-name
- launch-index
- launch-time
- monitoring-state
- owner-id
- placement-group-name
- platform
- private-dns-name
- private-ip-address
- product-code
- ramdisk-id
- reason
- requester-id
- reservation-id
- root-device-name
- root-device-type
- spot-instance-request-id
- state-reason-code
- state-reason-message
- subnet-id
- tag-key
- tag-value
- virtualization-type
- vpc-id
- tag:<tag-name> where <tag-name> is the name of a tag that an instance may have. See the notes below for an example of filtering the instances based on a specific tag/value combination.

`filterValues1`

A first value/values list for the corresponding filter name.

`filterName[number]`

To add additional filters, you must add extra inputs using the following naming convention: `filterName[number]` where you replace `[Number]` with increasing values (for example, `filterName2`, `filterName2`).

`filterValues[number]`

To add additional values, you must add extra inputs using the following naming convention: `filterValues[number]` where you replace `[Number]` with increasing values (for example, `filterValues2`, `filterValues3`).

`delimiter`

The delimiter used for the list of filter values, the `instanceIds` list and for the list of security groups in the response string. The default value is a comma(,).

`rowDelimiter`

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

`colDelimiter`

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

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Your EC2 instances in a table with the following columns: Instance Id, AMI ID, Machine type, Status, Public DNS, Key pair name, Ramdisk ID, Platform, Kernel ID, Monitoring, Root Device Name, Root Device Type, State Transition Reason, Availability Zone, Security groups, Subnet Id, Vpc Id, Private Ip Address, and Block Device Mapping. Column fields are separated by the `colDelimiter` input. If you have no instances, an empty string is returned.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- To list the resources assigned tag `Purpose=X` or `Purpose=Y`, specify:
`filterName1=tag:Purpose`
`filterValues1=X,Y`
- You can find more details on the filters that can be applied on this operation in the `DescribeInstances` section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.
- If you assign a value of `block-device-mapping.attach-time` or `launch-time` to a `filterName` input, the timestamps supplied to the corresponding `filterValues` must have the local timezone and must use the format `yyyy/MM/dd HH:mm:ss`. In this case, you cannot use

wildcards for filterValues.

Get Instance Details

The **Get Instance Details** operation returns details about a specific Amazon EC2 instance.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

instanceIdentifier

The ID of the instance you want details about. If you specify a nonexistent ID, the operation fails.

delimiter

The delimiter used for the results that are lists. The default value is a comma (,).

Results

The operation returns the following results:

returnResult

The instanceId of the specified instance.

imageId

The ID of the Amazon Machine Image (AMI) used for this instance.

kernelId

The operating system kernel associated with the instance (optional).

platform

The operating system platform (optional).

publicDns

The public hostname of the instance (contactable from outside the Amazon EC2 network).

privateDns

The private hostname of the instance (can only be used inside the Amazon EC2 network).

launchTime

The time the instance was launched.

status

The status of the instance (for example, running, pending, or terminated).

transitionReason

The reason the instance entered this state.

availabilityZone

The availability zone in which the instance is located.

machineType

The type of the instance (for example, m1.small or c1.xlarge).

ownerId

The account number of the AMI owner.

ramdiskId

The RAM disk associated with the image (optional).

keyPairName

The key pair name if this instance was launched with an associated key pair.

launchIndex

The order in which the instance was launched when launching more than one instance in one reservation.

monitoring

The monitoring status.

reservationId

The reservation ID used to launch the instance.

productCodes

The list of product codes.

groupSet

A list of the security groups to which the instance belongs, separated by the delimiter input.

requestId

The ID of the request sent to Amazon.

subnetId

The ID of the subnet.

vpcId

The VPC ID.

privateIpAddress

The private IP address that is assigned to the instance.

blockDeviceMapping

The block device mapping set.

placementGroup

The placement group to which the instance belongs.

tags

The tags assigned to the resource. Each tag is present in the format tagKey=tagValue in the returned list (for example, Owner=DbAdmin,Stack=Production).



- Like all Amazon EC2 operations, the results might not be visible instantly.
- Instances that are terminated or starting up may not have a public DNS name or other attributes.

Get System Log

The **Get System Log** operation retrieves the console output (the system log) of a running instance. Instance console output is buffered and posted shortly after instance boot, reboot, and termination. Amazon EC2 preserves the most recent 64 KB of output which is available for at least one hour after the most recent post.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

instanceId

The ID of the instance you want to query.

Results

The operation returns the following results:

requestId

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

returnResult

The console output.

updateTime

The time the output was last updated.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- This operation may also work on instances which were terminated a few days before. Use the Get Instance Details operation to verify that the queried instance is still alive.

Modify Instance Attribute

The **Modify Instance Attribute** operation modifies an attribute of an instance. The instance must be in a stopped state.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

instanceId

The ID of the instance.

attribute

Specifies the attribute to modify. The valid values are `instanceType`, `kernel`, `ramdisk`, `userData`, `disableApiTermination`, `instanceInitiatedShutdownBehavior`, and `blockDeviceMapping`.

value

The value of the attribute to be modified.

Results

The operation returns the following results:

requestId

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

returnResult

Returns true if successful. Otherwise, it returns an error.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- If the value `blockDeviceMapping` is assigned to the attribute input, then the value input should have one of the following formats:
 - `DeviceName=VolumeId>DeleteOnTermination`
Examples: `/dev/sda1=vol-c2acb5ab:false`, `/dev/sdc=vol-959fe1fe:true`
 - `DeviceName=VolumeId`
If the **DeleteOnTermination** flag is not explicitly specified, then the value `true` is used by default.
Example: `/dev/sda1=vol-959fe1fe`
- By default, EBS volumes created and attached to an instance at instantiation are destroyed when the EC2 instance is terminated. This operation can be used to change this behavior by passing the `blockDeviceMapping` value to the attribute input and setting the `false` value to the `DeleteOnTermination` flag for the device and volume specified through the value input.
- If the value `blockDeviceMapping` is assigned to the attribute input but the instance does not have a block device with the `DeviceName` and `VolumeId` specified through the value input, then the operation succeeds but no changes are made to the instance.

Monitor Instance

The **Monitor Instance** operation enables Amazon CloudWatch monitoring for a running instance.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`instanceIdentifier`

The ID of the instance you want to monitor.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The state of the monitoring after the execution of this operation (for example, pending).



Like all Amazon EC2 operations, the results might not be visible instantly.

Reset Instance Attribute

The **Reset Instance Attribute** operation resets an attribute of an instance to its default value. The instance must be in a stopped state.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`instanceId`

The ID of the instance.

`attribute`

Specifies the attribute to modify. The valid values are kernel and ramdisk.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true if successful. Otherwise, it returns an error.



Like all Amazon EC2 operations, the results might not be visible instantly.

Restart Instance

The **Restart Instance** operation requests a reboot of one instance. The operation succeeds if the instance is valid and belongs to you. Requests to reboot terminated instances are ignored.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

instanceIdentifier

The ID of the instance you want to reboot.

Results

The operation returns the following results:

requestId

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

returnResult

Returns true if the instance is being restarted. This may take some time and the instance's status may remain running while it is being restarted.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- We recommend that you verify (for instance, using the Get Instance Details operation) that the instance you want to restart exists and that it is in the running state. Otherwise, the operation's returnResult may be true but nothing may have happened.

Run Instances

The **Run Instances** operation launches a specified number of instances of an Amazon Machine Image (AMI) for which you have permissions. If Amazon EC2 cannot launch the minimum number AMIs you request, no instances are launched. If there is insufficient capacity to launch the maximum number of AMIs you request, Amazon EC2 launches the minimum number specified for each AMI and allocates the remaining available instances using round robin.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

amiId

The unique ID of the Amazon Machine Image you want to launch.

minCount

The minimum number of instances you want to launch from the specified AMI.

maxCount

The maximum number of instances you want to launch from the specified AMI.

kernelId

The ID of the kernel with which to launch the instance (for example, aki-12345678).

instanceType

The instance type. The valid values are m1.small, m1.large, m1.xlarge, c1.medium, and c1.xlarge. If the amiId input specifies a 64 bit image, then this input should be provided with one of the following values: m1.large, m1.xlarge, or c1.xlarge. If the amiId input specifies a 32 bit image, then this input can be left empty or given one of the following values: m1.small or c1.medium. In this case, if the input is left empty, a small instance is created by default.

placementZone

The placement constraints (availability zones) for launching the instances. By default, Amazon selects an availability zone automatically (for example, us-east-1a, us-east-1b).

monitoring

Specifies whether the instance should be monitored after its launch. The valid values are true or enabled if the instance should be monitored after launch and false or disabled if not. The default is false. If you do not specify a value, the instance is not monitored after starting.

ramdiskId

The ID of the RAM disk with which to launch the instance, required by some kernels (for example, ari-12345678).

keyName

The name of one of your existent key pairs, used to access your instance. Launching public images without a key pair ID leaves them inaccessible.

securityGroups

The name of the security group or groups of which your instances will be part. If you do not specify a security group, the default group is used.

placementGroup

The name of the placement group your instances will be part of.

delimiter

The delimiter for returned lists of IDs and the securityGroups input. The default is a comma (,).

subnetId

The Amazon VPC subnet ID within which to launch the instance(s) for Amazon Virtual Private Cloud.

privateIpAddress

The IP address from the VPC subnet that is assigned to the instance (for example, 10.0.0.25). If you specify an IP address, you can launch a single instance. By default, Amazon VPC selects an IP address from the subnet.

`disableApiTermination`

Specifies whether the instance can be terminated using the APIs. You must modify this attribute before you can terminate any "locked" instances from the APIs. The valid values are true and false. A value of false means the instance is unlocked. The valid values are true and false.

`instanceInitiatedShutdownBehavior`

Specifies whether the instance's Amazon EBS volumes are stopped or terminated when a shutdown is issued from with the instance. The valid values are stop and terminate.

`deviceName1`

The device name (for example, /dev/sdh).

`deviceName[number]`

To define more than one block device mapping, you should add extra inputs using the following naming convention: `deviceName[number]` where you replace `[number]` with an increasing number for each combination (for example, `deviceName2`, `deviceName3`).

`virtualName1`

The virtual device name (for example, `ephemeral0`).

`virtualName[number]`

To define more than one block device mapping, you must add extra `virtualName` inputs using the naming convention `virtualName[number]` where you replace `[number]` with an increasing number for each combination (for example, `virtualName2`, `virtualName3`).

`volumeSize1`

The size of the volume in GBs.

`volumeSize[number]`

To define more than one block device mapping, you must add extra `volumeSize` inputs using the naming convention `volumeSize[number]` where you replace `[number]` with an increasing number for each combination (for example, `virtualSize2`, `virtualSize3`).

`snapshotId1`

The ID of the snapshot (for example, `snap-c02745a8`).

`snapshotId[number]`

To define more than one block device mapping, you must add extra `snapshotId` inputs using the naming convention `snapshotId[number]` where you replace `[number]` with an increasing number for each combination (for example, `snapshotId2`, `snapshotId3`).

`deleteOnTermination1`

Specifies whether the Amazon EBS volume is deleted on instance termination. The valid values are true and false. The default value is true.

`deleteOnTermination[number]`

To define more than one block device mapping, you must add extra `deleteOnTermination` inputs using the naming convention `deleteOnTermination[number]` where you replace `[number]` with an increasing number for each combination (for example, `deleteOnTerminated2`, `deleteOnTerminated3`).

`clientToken`

A unique identifier you provide to ensure idempotency of the request. If the operation timesout or connection errors occur, you can rerun the operation and be sure you haven't launched more instances than you intended. The token must be a unique, case-sensitive string of up to 64 ASCII characters. If you repeat the request with the same client token, the same response is returned for each repeated request. The client token is valid for at least 24 hours after the termination of the instance. You should not reuse a client token for another call later on. You can use the same client token for the same request across different regions (for example, this is a token).

Results

The operation returns the following results:

`returnResult`

A list containing the IDs of your newly started instances separated by the delimiter input value.

`reservationID`

The reservation ID used to launch the instances.

`requestId`

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

`requesterId`

The ID of the requester who sent this request to Amazon.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- The block device mappings exposed to the instance are made up of a virtual name (`virtualName`) and a device name (`deviceName`). Virtual names include `none` and `ephemeral[0-3]`. Virtual names are used together with device names (for example, `/dev/sdb=none`, `/dev/sdc=ephemeral0`).
- A set of `deviceName`, `virtualName`, `volumesize`, `snapshotId`, and `deleteOnTermination` values represent a block device mapping item. Each item specifies whether the device is mapped to a virtual device (for instance, a local instance store), an Amazon EBS volume, or to nothing. For a device mapped to a virtual device, the block device mapping specifies the name of the store (for instance, `ephemeral0`). For a device mapped to an Amazon EBS volume (which is available only for Amazon EBS-backed AMIs), the block device mapping specifies the following items: `snapshotId` to use for the volume, `volumeSize` (in GB), and `deleteOnTermination` flag (whether to delete the volume on instance termination).
- If you do not specify a value for the `minCount` and `maxCount` inputs, one instance is launched.

- The architecture of your AMI, RAM disk, and kernel must match. Even then, there is no guarantee that a given combination will function correctly or boot successfully if they weren't designed to be used together.
- Instances usually enter the pending state after this operation. They do not have DNS names and other attributes for at least a few minutes.
- If any of the AMIs has a product code attached for which the user has not subscribed, the operation fails.

Run Instances From Template

The **Run Instances From Template** flow launches a specified number of instances with the same configuration as a given one. If Amazon EC2 cannot launch the minimum number of instances you request, no instances are launched. If there is insufficient capacity to launch the maximum number of instances you request, Amazon EC2 launches the minimum number specified and allocates the remaining available instances using round robin.

Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*

`instanceIdentifier`

The ID of the template instance. The following configuration characteristics of the instance are used when launching the new instances: AMI ID, availability zone, machine type, kernel ID, ramdisk ID, monitoring, key pair name, and security groups.

`minCount`

The minimum number of instances you want to launch.

`maxCount`

The maximum number of instances you want to launch.

Results

The flow returns the following results:

`returnResult`

A list containing the IDs of your newly started instances.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- If you do not specify a value for the minCount and maxCount inputs, one instance is launched.
- Instances usually enter the pending state after this flow. They do not have DNS names and other attributes for at least a few minutes.
- If any of the AMIs has a product code attached for which the user has not subscribed, the operation fails.

Start Instance

The **Start Instance** operation starts an instance that uses an Amazon EBS volume as its root device. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, the root partition Amazon EBS volume remains, continuing to persist the data, and you are charged for Amazon EBS volume usage. You can restart the instance at any time.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`instanceId`

The instance ID.

Results

The operation returns the following results:

`returnResult`

The result of the request.

`requestId`

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

`name`

The state of the instance.



Like all Amazon EC2 operations, the results might not be visible instantly.

Stop Instance

The **Stop Instance** operation stops an instance that uses an Amazon EBS volume as its root device. Instances that use Amazon EBS volumes as their root devices can be quickly stopped and started. When an instance is stopped, the compute resources are released and you are not billed for hourly instance usage. However, your root partition Amazon EBS volume remains, continuing to persist your data, and you are charged for Amazon EBS volume usage. You can restart your instance at any time.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`instanceIdentifier`

The ID of the instance you want to stop.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The state of the instance after the operation (for example, shutting-down).

`previousState`

The state the instance was in before you stopped it (for example, running).



Like all Amazon EC2 operations, the results might not be visible instantly.

Terminate Instance

The **Terminate Instance** operation shuts down an instance. If you terminate an instance more than once, each call succeeds. Terminated instances remain visible (for instance, when using the **Describe Instances** operation) after termination for approximately one hour.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

`instanceIdentifier`

The ID of the instance you want to terminate.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The state of the instance after the operation (for example, shutting-down).

`previousState`

The state the instance was in before you stopped it (for example, running).



Like all Amazon EC2 operations, the results might not be visible instantly.

Unmonitor Instance

The **Unmonitor Instance** operation disables Amazon CloudWatch monitoring for a running instance.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`instanceIdentifier`

The ID of the instance for which you want to disable monitoring.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The state of the monitoring after the execution of this operation (for example, disabling).



Like all Amazon EC2 operations, the results might not be visible instantly.

Key Pairs

Allocate New Key Pair

The **Allocate New Key Pair** operation creates a new 2048-bit RSA key pair with the specified name. The private key is returned as an unencrypted PEM encoded PKCS#8 private key. If a key with the specified name already exists, Amazon EC2 returns an error.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`keyName`

The name of the new key pair.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

The key material of your RSA private key.

keyFingerprint

An SHA-1 digest of the DER encoded private key.



Like all Amazon EC2 operations, the results might not be visible instantly.

Delete Key Pair

The **Delete Key Pair** operation deletes the specified key pair by removing the public key from Amazon EC2. You must own the key pair. The operation succeeds if you call it more than once on a given key and even if the key does not exist. We recommend that you check that the key you want to delete exists by using the Enumerate Key Pairs operation.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

keyName

The name of the key pair you want to delete.

Results

The operation returns the following results:

requestId

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

returnResult

Returns true if the operation succeeds.



Like all Amazon EC2 operations, the results might not be visible instantly.

Enumerate Key Pairs

The **Enumerate Key Pairs** operation lists the key pairs that are available to you.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

rowDelimiter

The row delimiter in the results table. The default value is the line separator for the current operating system.

colDelimiter

The column delimiter in the results table. The default value is a comma (,).

`filterName1`

The first filter name for the operation. The valid values are fingerprint and key-name.

`filterValues1`

A first value or values list for the corresponding filter name.

`filterName[number]`

To add additional filters, you must add extra `filterName` inputs using the convention `filterName[number]` where you replace `[number]` with increasing numbers (for example, `filterName2`, `filterName3`).

`filterValues[number]`

To add additional values, you must add extra `filterValues` inputs using the naming convention `filterValues[number]` where you replace `[number]` with increasing numbers (for example, `filterValues2`, `filterValues3`).

`delimiter`

The delimiter used for the list of volume IDs and the list of filter values. The default is a comma (,).

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Your key pairs in a table with the following columns: Key name and Fingerprint.

Example:

```
testkey2,8c:0a:1a:c8:a5:07:cb:3d:0e:60:3f:00:7d:a1:e9:c4:4f:28:e4:7b
minFedora,8b:79:f0:6d:6d:4c:73:80:c0:1a:9e:e3:81:f3:cc:3f:33:20:cf:89
```



- Like all Amazon EC2 operations, the results might not be visible instantly.
- In case no key pair exists, the operation succeeds but `returnResult` is empty.
- You can find more details on the filters that can be applied on this operation in the `DescribeKeyPairs` section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.

Import Key Pair

The **Import Key Pair** operation imports the public key from an RSA key pair that you created with a third-party tool to your EC2 account. Only the public key is transmitted to Amazon.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

keyName

A name for the key pair. Make sure there is no other key with the same name.

publicKeyMaterial

The path (including file name) to the public key file.

Results

The operation returns the following results:

requestId

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

returnResult

The key pair name that was provided.

keyFingerprint

An MD5 public key fingerprint.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- The supported formats are the following:
 - OpenSSH public key format.
 - Base64 encoded DER format.
 - SSH public key file format as specified in RFC4716.
 - DSA keys are not supported.
- The supported lengths are:
 - 1024
 - 2048

Placement Groups

Create Placement Group

The **Create Placement Group** operation creates a new placement group.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

groupName

The name of the new placement group.

`strategy`

The placement group strategy. Currently, the only supported value is cluster.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true when the operation is successful, and false if it fails.



Like all Amazon EC2 operations, the results might not be visible instantly.

Delete Placement Group

The **Delete Placement Group** operation deletes a placement group.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`groupName`

The name of the placement group.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true when the operation is successful, and false if it fails.



Like all Amazon EC2 operations, the results might not be visible instantly.

Describe Placement Groups

The **Describe Placement Groups** operation returns a list of your Amazon EC2 placement options.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

groupName

The name of the placement groups to describe. If you leave the value for this input blank, all of your owned placement groups are described.

filterName1

The first filter name for the operation. The valid values are group-name, state, and strategy.

filterName[number]

To add additional filters, you must add extra filterName inputs using the convention filterName[number] where you replace [number] with increasing numbers (for example, filterName2, filterName3).

filterValues1

The first value or list of values for the corresponding filter name.

filterValues[number]

To add additional values, you must add extra filterValues inputs using the convention filterValues[number] where you replace [number] with increasing numbers (for example, filterValues2, filterValues3).

delimiter

The delimiter used for the list of placement group names and the lists of filter values. The default is a comma (,).

rowDelimiter

The delimiter to use between the rows in the results table. The default is the line separator of the current operating system.

colDelimiter

The delimiter to use to separate the fields that describe a placement group in the result string. The default is a semicolon (;).

Results

The operation returns the following results:

requestId

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

returnResult

A table listing the placement groups, one in each row. The table has the following columns: group name, strategy, and state.



Like all Amazon EC2 operations, the results might not be visible instantly.

Security groups

Allow Access To Security Group

The **Allow Access To Security Group** operation adds new permissions to the specified security group. You can specify the new permissions by providing the IP protocol, start and end of port range, and the CIDR range or source group.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`groupName`

The name of the group to modify.

`ipProtocol1`

The IP protocol used. The valid values are tcp, udp, and icmp.

`ipProtocol[number]`

If you need more than one combination of ipProtocol, fromPort, and toPort when adding access permissions, you must add extra ipProtocol inputs (one for each combination) following the naming convention ipProtocol[number] where you replace [number] with increasing numbers (for example, ipProtocol2, ipProtocol3).

`fromPort1`

The start of the port range for the TCP and UDP protocols, and the type number for ICMP. For ICMP, -1 indicates a wildcard (any ICMP type number).

`fromPort[number]`

If you need more than one combination of ipProtocol, fromPort, and toPort when adding access permissions, you must add extra fromPort inputs (one for each combination) following the naming convention fromPort[number] where you replace [number] with increasing numbers (for example, fromPort2, fromPort3).

`toPort1`

The end of the port range for the TCP and UDP protocols, and the type number for ICMP. For ICMP, -1 indicates a wildcard (any ICMP type number).

`toPort[number]`

If you need more than one combination of ipProtocol, fromPort, and toPort when adding access permissions, you must add extra toPort inputs (one for each combination) following the naming convention toPort[number] where you replace [number] with increasing numbers (for example, toPort2, toPort3).

`sourceSecurityGroupNames1`

A list containing the names of the source security groups. This is required when giving access to one or more source security groups.

sourceSecurityGroupNames[number]

If you need more than one list of source security groups when adding access permissions, you must add extra sourceSecurityGroupNames inputs (one for each combination) following the naming convention sourceSecurityGroupNames[number] where you replace [number] with increasing numbers (for example, sourceSecurityGroupNames2, sourceSecurityGroupNames3).

sourceSecurityGroupOwnerIds1

A list containing the user IDs of each of the source security group's owners. This is required when giving access to one or more source security groups.

sourceSecurityGroupOwnerIds[number]

If you need more than one list of source security groups when adding access permissions, you must add extra sourceSecurityGroupOwnerIds inputs (one for each combination) following the naming convention sourceSecurityGroupOwnerIds[number] where you replace [number] with increasing numbers (for example, sourceSecurityGroupOwnerIds2, sourceSecurityGroupOwnerIds3).

cidrIps1

A list containing the source IPs in CIDR format. This is required when giving access to one or more IP address ranges.

cidrIps[number]

If you need more than one list of CIDR ranges when adding access permissions, you must add extra cidrIps inputs (one for each combination) following the naming convention cidrIps[number] where you replace [number] with increasing numbers (for example, cidrIps2, cidrIps3).

delimiter

The delimiter used to separate the source security group names and source security group owner ID, or CIDR IPs. The default value is a comma (,).

Results

The operation returns the following results:

requestId

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

returnResult

Returns true when the operation is successful and false when it is not.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- If a permission is specified through the source group and the ipProtocol, fromPort, and toPort inputs are not specified, then all the following permissions are added:
 - tcp protocol, from port 0, to port 65535, specified source group
 - udp protocol, from port 0, to port 65535, specified source group
 - icmp protocol, from port -1, to port -1, specified source group
- If you add a combination of access parameters that already exists, the operation fails.

- In case the new permissions are specified by a source group name, any machine from `sourceSecurityGroupName` can connect to any machine from `groupName`, using any of the TCP, UDP or ICMP protocols.

Create Security Group

The **Create Security Group** operation creates a new security group.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`groupName`

The name of the new group. This name must be unique per account.

`groupDescription`

The description of the group. This is only informational, and can contain alphanumeric characters, spaces, dashes, and underscores.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true when the operation is successful or false if it is not.



Like all Amazon EC2 operations, the results might not be visible instantly.

Delete Security Group

The **Delete Security Group** operation deletes a security group.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`groupName`

The name of the security group you want to delete.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true when the operation is successful or false if it is not.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- If you attempt to delete a security group which used to exist but has been recently deleted, the operation might go to Success. This may happen because Amazon remembers a deleted security group for a certain amount of time:
- If you attempt to delete a security group that contains instances, a fault is returned.
- If you attempt to delete a security group that is referenced by another security group, a fault is returned. For example, if Security Group B has a rule that allows access from Security Group A, Security Group A cannot be deleted until you remove the rule.

Describe Security Groups

The **Describe Security Groups** operation describes the security groups that you own. By default, this operation returns information about all of your security groups, but you can specify a list of group names to restrict the results to only those specified.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

`groupNames`

The names of the groups that you want to describe. If you leave this input blank, all of your owned groups are described.

`rowDelimiter`

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

`colDelimiter`

The delimiter used between columns in the output table. The default is a comma (,).

`filterName1`

The first filter name for the operation. The valid values are:

- description
- group-name
- ip-permission.cidr
- ip-permission.from-port
- ip-permission.group-name
- ip-permission.protocol
- ip-permission.to-port
- ip-permission.user-id

- owner-id

filterValues1

A first value or values list for the corresponding filter name.

filterName[number]

To add additional filters, you must add extra filterName inputs using the convention filterName[number] where you replace [number] with increasing numbers (for example, filterName2, filterName3).

filterValues[number]

To add additional values, you must add extra filterValues inputs using the naming convention filterValues[number] where you replace [number] with increasing numbers (for example, filterValues2, filterValues3).

delimiter

The delimiter used for the list of group names. The default is a comma (,).

Results

The operation returns the following results:

requestId

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

returnResult

A table listing the security groups, one in each row. The table has the following columns: group name, group owner id, and group description.

groupNames

A list of the group names of your security groups.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- You can find more details on the filters that can be applied on this operation in the DescribeSecurityGroups section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.

Get Security Group Details

The **Get Security Group Details** operation returns information about a specific security group that you own.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

groupName

The name of the security group you want to describe.

rowDelimiter

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

colDelimiter

The delimiter used between columns in the output table. The default is a comma (,).

Results

The operation returns the following results:

requestId

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

returnResult

The name of the group.

groupName

The name of the group.

ownerId

The ID of the owner of the group.

groupDescription

The description of the group.

ipPermissions

A table describing the permissions for this group, one in each row. The table has the following columns:

- ip protocol
- from port The start of the port range for the TCP and UDP protocols, or an ICMP type number (an ICMP type number of -1 indicates a wildcard—any ICMP type number).
- to port The end of the port range for the TCP and UDP protocols, or an ICMP code (-1 indicates a wildcard).
- source security group The name of the source security group.
- user id The user ID of the owner of the source security group.
- source IP in CIDR notation



Like all Amazon EC2 operations, the results might not be visible instantly.

Revoke Access To Security Group

The **Revoke Access To Security Group** operation removes permissions from the specified security group. You can specify the permissions by giving the IP protocol, start and end of port range, and the CIDR range or source group.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

groupName

The name of the group to modify.

ipProtocol1

The IP protocol used. The valid values are tcp, udp, and icmp.

ipProtocol[number]

If you need additional combinations of ipProtocol, fromPort, and toPort to remove access permissions, you must add extra ipProtocol inputs for each combination following the naming convention ipProtocol[number], where you replace [number] with increasing numbers (for example, ipProtocol2, ipProtocol3).

fromPort1

The start of the port range for the TCP and UDP protocols, and the type number for ICMP. If you do not specify a value for ICMP, port 0 is used by default. For ICMP, -1 indicates a wildcard (any ICMP type number).

fromPort[number]

If you need additional combinations of ipProtocol, fromPort, and toPort to remove access permissions, you must add extra fromPort inputs for each combination following the naming convention fromPort[number] where you replace [number] with increasing numbers (for example, fromPort2, fromPort3).

toPort1

The end of the port range for the TCP and UDP protocols, and the type number for ICMP. If you do not specify a value for ICMP, port 0 is used by default. For ICMP, -1 indicates a wildcard (any ICMP type number).

toPort[number]

If you need additional combinations of ipProtocol, fromPort, and toPort to remove access permissions, you must add extra toPort inputs for each combination following the naming convention toPort[number] where you replace [number] with increasing numbers (for example, toPort2, toPort3).

sourceSecurityGroupNames1

A list containing the names of the source security groups. This is required when revoking access to one or more source security groups.

`sourceSecurityGroupNames[number]`

If you need more than one list of source security groups when removing access permissions, you must add extra `sourceSecurityGroupNames` inputs (one for each list) following the naming convention `sourceSecurityGroupNames[number]` where you replace `[number]` with increasing numbers (for example, `sourceSecurityGroupNames2`, `sourceSecurityGroupNames3`).

`sourceSecurityGroupOwnerIds1`

A list containing the user IDs of each of the owner of the source security group's owners. This is required when revoking access to one or more source security groups.

`sourceSecurityGroupOwnerIds[number]`

If you need more than one list of source security groups when removing access permissions, you must add extra `sourceSecurityGroupOwnerIds` inputs (one for each list) following the naming convention `sourceSecurityGroupOwnerIds[number]` where you replace `[number]` with increasing numbers (for example, `sourceSecurityGroupOwnerIds2`, `sourceSecurityGroupOwnerIds3`).

`cidrIps1`

A list containing the source IPs in CIDR format. This is required when revoking access to one or more IP address ranges.

`cidrIps[number]`

If you need more than one list of CIDR ranges when removing access permissions, you must add extra `cidrIps` inputs (one for each list) following the naming convention `cidrIps[number]` where you replace `[number]` with increasing numbers (for example, `cidrIps2`, `cidrIps3`).

`delimiter`

The delimiter used to separate the source security group names and source security group owner IDs, or CIDR IPs. The default value is a comma (,).

Results

The operation returns the following results:

`requestId`

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

`returnResult`

Returns true when the operation is successful or false if it is not.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- This operation goes to success even in situations when no permission is revoked or the permission does not exist. The only situation when a permission is actually revoked is when the inputs match exactly the inputs used to create the permission (for example, the inputs given to the Allow Access To Security Group operation).
- If a permission is specified through the source group, and the `IpProtocol`, `fromPort` and `toPort` are not assigned, the operation attempts to remove all the following permissions:
 - tcp protocol, from port 0, to port 65535, specified source group
 - udp protocol, from port 0, to port 65535, specified source group

- icmp protocol, from port -1, to port -1, specified source group

Spot Instances

Cancel Spot Instance Requests

The **Cancel Spot Instance Requests** operation cancels one or more spot instance requests.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

`spotInstanceRequestIds`

A list of spot instance request IDs separated by the delimiter input value.

`rowDelimiter`

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

`colDelimiter`

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

`delimiter`

The delimiter used to separate the spot instance request IDs in the `spotInstanceRequestIds` input.

Results

The operation returns the following results:

`requestId`

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

`returnResult`

A table containing descriptions of the instance requests. The table has the following columns: spot instance request ID and state. The column fields are separated by the `colDelimiter` input value.



Like all Amazon EC2 operations, the results might not be visible instantly.

Create Spot Datafeed Subscription

The **Create Spot Datafeed Subscription** operation creates the data feed subscription for spot instances, allowing you to view spot instance usage logs. You can create one data feed per account.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

storageBucket

The Amazon S3 bucket in which to store the spot instance data feed. This must be a valid bucket associated with your account, and you must have FULL_CONTROL over the bucket.

prefix

The prefix that is prepended to data feed file names. The spot instance data feed file name uses the following format (with the date and hour in UTC):

```
{Bucket}.s3.amazonaws.com/{Optional Prefix}{Distribution ID}.{YYYY}-{MM}-{DD}-{HH}.{Unique ID}.gz
```

Results

The operation returns the following results:

returnResult

The state of the spot instance request. The valid values are open, closed, canceled, and failed.

requestId

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

ownerId

The AWS account ID of the account.

bucket

The Amazon S3 bucket where the spot instance data feed is located.

prefix

The prefix that is prepended to data feed file names.

state

The state of the spot instance request. The valid values are open, closed, canceled, and failed.

code

The reason code for the spot instance state change.

message

The message for the spot instance state change.



Like all Amazon EC2 operations, the results might not be visible instantly.

Delete Spot Datafeed Subscription

The **Delete Spot Datafeed Subscription** operation deletes the data feed for spot instances.

All of the operation's inputs are described in [Common Inputs in the Integration](#)

Results

The operation returns the following results:

`returnResult`

The state of the execution; true if the operation executes successfully, otherwise an error message.

`requestId`

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



Like all Amazon EC2 operations, the results might not be visible instantly.

Describe Spot Datafeed Subscription

The **Describe Spot Datafeed Subscription** operation describes the data feed for spot instances.

Inputs

All of the operation's inputs are described in *Common Inputs in the Integration*

Results

The operation returns the following results:

`returnResult`

The state of the spot instance request. The valid values are open, closed, canceled, and failed.

`requestId`

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

`ownerId`

The AWS account ID of the account.

`bucket`

The Amazon S3 bucket where the spot instance data feed is located.

`prefix`

The prefix that is prepended to data feed file names.

`state`

The state of the spot instance request. The valid values are open, closed, canceled, and failed.

`code`

The reason code for the spot instance state change.

message

The message for the spot instance state change.



Like all Amazon EC2 operations, the results might not be visible instantly.

Describe Spot Instance Requests

The **Describe Spot Instance Requests** operation describes spot instance requests.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`spotInstanceRequestIds`

A list of spot instance request IDs separated by the delimiter input value. If you do not specify any list values, all of your requests are returned.

`rowDelimiter`

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

`colDelimiter`

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

`filterName1`

The first filter name for the operation. The valid values are:

- availability-zone-group
- create-time
- fault-code
- fault-message
- instance-id
- launch-group
- launch.block-device-mapping.delete-on-termination
- launch.block-device-mapping.device-name
- launch.block-device-mapping.snapshot-id
- launch.block-device-mapping.volume-size
- launch.group-id
- launch.image-id
- launch.instance-type
- launch.kernel-id
- launch.key-name
- launch.monitoring-enabled

- launch.ramdisk-id
- product-description
- spot-instance-request-id
- spot-price
- state
- tag-key
- tag-value
- type
- valid-from
- valid-until
- tag:<tag-name> where <tag-name> stands for the name of a tag that a spot instance request may have. See the notes below for an example of filtering the spot instance requests based on a specific tag/value combination.

filterValues1

The first value or values list for the corresponding filter name.

filterName[number]

To add more filters, you must add extra filterName inputs using the naming convention filterName[number] where you replace [Number] with an increasing number (for example, filterName2, filterName3).

filterValues[number]

To add more values, you must add extra filterValues inputs using the naming convention filterValues[number] where you replace [Number] with an increasing number (for example, filterValues2, filterValues3).

delimiter

The delimiter used to separate the spot instance request IDs and items in the lists of filter values. It is also used to separate the security group ID. The default value is a comma (,).

Results

The operation returns the following results:

returnResult

Descriptions of the instances in a table with the following columns:

- spotInstanceRequestId
- spotPrice
- type
- state
- code
- message

- validFrom
- validUntil
- launchGroup
- availabilityZoneGroup
- imageId
- keyName
- group Id(s) (separated by delimiter)
- addressingType
- instanceType
- placement
- kernelId
- ramdiskId
- subnetId
- instanceId
- createTime
- productDescription

The column fields are separated by the **colDelimiter** input value.

requestId

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



- Like all Amazon EC2 operations, the results might not be visible instantly.
- To list the resources assigned tag Purpose=X or Purpose=Y specify:

filterName1=tag:Purpose

filterValues1=X,Y

- You can find more details on the filters that you can apply on this operation in the DescribeSpotInstanceRequests section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.
- If the value assigned to a filterName input is create-time, valid-from, or valid-until, the timestamps supplied to the corresponding filterValues must have the local time zone and must use the following format: yyyy/MM/dd HH:mm:ss. You cannot use wildcards for filterValues.

Describe Spot Price History

The **Describe Spot Price History** operation describes spot price history.

Inputs

All of the operation's inputs except the following are described in *[Common Inputs in the Integration](#)*

startDate

The start date and time of the spot instance price history data. The format of the date can be yyyy/MM/dd HH:mm:ss or yyyy/MM/dd. For the last format, 00:00:00 is the time considered.

endDate

The end date and time of the spot instance price history data. The format of the date can be yyyy/MM/dd HH:mm:ss or yyyy/MM/dd. For the last format, 00:00:00 is the time considered.

instanceType

Specifies the instance types to return. Separate the types with the delimiter input value. The available instance types are m1.small, m1.large, m1.xlarge, c1.medium, c1.xlarge, m2.xlarge, m2.2xlarge, m2.4xlarge, and t1.micro.

filterName1

The first filter name for the operation. The valid values are instance-type, product-description, spot-price, and timestamp.

filterValues1

A first value/values list for the corresponding filter name.

filterName[number]

To add more filters, you must add extra inputs using the naming convention filterName[number] where you replace [Number] with increasing numbers (for example, filterName2, filterName3).

filterValues[number]

To add more values, you must add extra inputs using the naming convention filterValues[number] where you replace [Number] with increasing numbers (for example, filterValue2, filterValue3).

delimiter

The delimiter used to separate the instance types and the items in the list of filter values. The default value is a comma (,).

description

The product description. The valid values are Linux/UNIX, SUSE/Linux, and Windows.

rowDelimiter

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

colDelimiter

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

Results

The operation returns the following results:

returnResult

Descriptions in a table with the following columns: Instance Type, Product Description (description of the AMI), Spot Price, and Timestamp.

Column fields are separated by the colDelimiter input value. If the price history is empty, an empty string is returned.

requestId

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



- Like all Amazon EC2 operations, the results might not be visible instantly.
- This operation returns all the prices that are active in the given interval between startDate and endDate and not the prices which were issued in this interval.
- You can find more details on the filters that can be applied on this operation in the DescribeSpotPriceHistory section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.
- If the value assigned to a filterName input is timestamp, then the timestamps supplied to the corresponding filterValues must have the local time zone and must use the following format: yyyy/MM/dd HH:mm:ss. In this case, you cannot use wildcards for filterValues.

Request Spot Instances

The **Request Spot Instances** operation creates a spot instance request.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

spotPrice

Specifies the maximum hourly price for any spot instance launched to fulfill the request. The price is in US dollars.

instanceCount

The maximum number of spot instances to launch. The default is 1.

type

The type of the spot instances to launch. The available types are one-time and persistent. The default is one-time.

validFrom

The start date of the request. The format of the date is "yyyy/MM/dd HH:mm:ss".

validUntil

The end date of the request. The format of the date is "yyyy/MM/dd HH:mm:ss".

launchGroup

Specifies the instance launch group. Launch groups are spot instances that launch together and terminate together.

availabilityZoneGroup

Specifies the availability zone group. If you specify the same Availability Zone group for all spot instance requests, all spot instances are launched in the same Availability Zone.

imageId

The AMI ID.

keyName

Specifies the name of one of your existent key pairs used to access your instance. Launching public images without a key pair ID will leave them inaccessible.

securityGroups

The name of the security group or groups of which your spot instances will be part. If you do not specify a security group or groups, the default group is used.

userData

MIME, Base 64-encoded user data sent to the spot instances at launch time.

instanceType

Specifies the instance type. The available types are m1.small, m1.large, m1.xlarge, c1.medium, c1.xlarge, m2.xlarge, m2.2xlarge, and m2.4xlarge.

kernelId

The ID of the kernel with which to launch the instance(s).

ramdiskId

The ID of the RAM disk with which to launch the instance(s). This is required by some kernels.

subnetId

The Amazon VPC subnet ID within which to launch the instance(s) for Amazon Virtual Private Cloud.

availabilityZone

Specifies the placement constraints (availability zones) for launching the instance(s).

deviceName1

The device name (for instance, /dev/sdh).

virtualName1

The virtual device name (for instance, ami, root, swap, or ephemeralN).

snapshotId1

The ID of the snapshot.

volumeSize1

The size of the volume in GBs. This is required if you are not creating a volume from a snapshot.

deleteOnTermination1

Set this to true if the Amazon EBS volume should be deleted on instance termination; otherwise, set it to false. The default is true.

monitoring

Set this to true if the instance should be monitored after launch; otherwise, set it to false. The default is false. If you do not provide a value, the instance is not monitored after starting.

rowDelimiter

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

colDelimiter

The delimiter separates the fields in a request description in the result string. The default is a comma (,).

delimiter

The delimiter used to separate the security groups. The default is a comma (,).

Results

The operation returns the following results:

returnResult

Descriptions of the instances in a table with the following columns:

- spotInstanceRequestId
- spotPrice
- type
- state
- code
- message
- validFrom
- validUntil
- launchGroup
- availabilityZoneGroup
- imageId
- keyName
- group Id(s) (separated by delimiter)
- instanceType
- placement

- kernelId
- ramdiskId
- subnetId
- instanceId
- createTime
- productDescription

The column fields are separated by the **colDelimiter** input value.

requestId

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



- Like all Amazon EC2 operations, the results might not be visible instantly.
- If this is a one-time request, the request becomes active at the start date and time and remains active until all instances launch, the request expires (the end date is reached), or the request is canceled. If the request is persistent, the request becomes active at this date and time and remains active until it expires or is canceled.
- If the user data (userData) is not correctly encoded, the operation goes to success but the request fails.
- The block device mappings exposed to the instance are made up of a device name (deviceName1) and a virtual name (virtualName1), and an Amazon EBS volume. Virtual names include:
 - ami - The root file system device, as seen by the instance.
 - root - The root file system device, as seen by the kernel.
 - swap - The swap device, as seen by the instance.
 - ephemeralN - The Nth ephemeral store
- The Amazon EBS volume is specified by its snapshot ID (snapshotId1). If you do not specify a snapshot ID, a new EBS volume is created with the given size (volumeSize1).
- If more than one block device mapping is required, you should add extra inputs using the following naming convention: deviceName[number], virtualName[number], snapshotId[number], volumeSize[number], and deleteOnTermination[number] where you replace [number] with increasing numbers such as 2, 3, ... The success of the operation does not guarantee the success of the request. For the request to succeed, the inputs should be valid according to the rules stated by Amazon regarding block device mapping.

Windows

Bundle Windows Instance

The **Bundle Windows Instance** operation bundles the Windows instance. The bundling process creates a new image from a running instance and stores the AMI data in S3. Once it

is bundled, you must register the image using the Register Image operation prior to launching it.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

instanceId

The ID of the instance you want to bundle.

storageBucket

The Amazon S3 bucket in which to store the AMI. You must specify a bucket that you already own. If you specify a bucket that belongs to someone else, Amazon EC2 returns an error.

storagePrefix

A prefix that specifies the beginning of the file name of the AMI. Your image is stored in the bucket specified by the storageBucket input under the name <storagePrefix>.manifest.xml.

s3AccessKeyId

The ID of the secret access key associated with your S3 AWS account, if it is different from the EC2 account. If you do not specify the s3AccessKeyId input, the operation uses the EC2 access key ID.

s3AccessKey

The secret access key associated with your S3 AWS account, if it is different from the EC2 account. If you do not specify the s3AccessKey input, the operation uses the EC2 access key.

Results

The operation returns the following results:

requestId

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

returnResult

The Bundle ID by that identifies the bundle task.

status

The state of the bundle task (for example, pending).

startTime

The time the bundle task started.



- Like all Amazon EC2 operations, the results might not be visible instantly.
- You cannot start another bundle task on an instance until the running task is terminated. Also, if the path you specified for the image already exists, any images under that name may be overwritten.
- The s3AccessKeyId and s3AccessKey inputs work together. Therefore, you must specify

values for both of them or leave them both.

- The operation succeeds for an invalid `s3AccessKey` creating a bundle task, but during the bundling process, the task fails with the following error message:
"SignatureDoesNotMatch(403)- The request signature we calculated does not match the signature you provided. Check your key and signing method."

Cancel Bundle Task

The **Cancel Bundle Task** operation aborts a bundle task.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`bundleId`

The ID of the bundle task. If you specify a nonexistent ID, the operation fails. If you specify the ID of an already terminated bundle task, the operation may succeed but nothing will happen.

Results

The operation returns the following results:

`returnResult`

The instance ID of the instance that was being bundled.

`startTime`

The time the task was launched.

`updateTime`

The time the status of the task was last updated.

`status`

The status of the task (for example, canceling or failed).

`progress`

The progress of the task (0% - 100%).

`bucket`

The S3 bucket where the image would have been placed.

`prefix`

The name under which the image would have been placed in the bucket.

`requestId`

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



- Like all Amazon EC2 operations, the results might not be visible instantly.

Enumerate Bundle Tasks

The **Enumerate Bundle Tasks** operation returns a list of your bundle tasks in a table.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

rowDelimiter

The row delimiter in the results table. The default value is the line separator for the current operating system.

colDelimiter

The column delimiter in the results table. The default value is a comma (,).

filterName1

The first filter name for the operation. The valid values are:

- bundle-id
- error-code
- error-message
- instance-id
- progress
- s3-aws-access-key-id
- s3-bucket
- s3-prefix
- start-time
- state
- update-time

filterValues1

A first value or values list for the corresponding filter name.

filterName[number]

To add additional filters, you must add extra filterName inputs using the convention filterName[number] where you replace [number] with increasing numbers (for example, filterName2, filterName3).

filterValues[number]

To add additional values, you must add extra filterValues inputs using the naming convention filterValues[number] where you replace [number] with increasing numbers (for example, filterValues2, filterValues3).

delimiter

The delimiter used for the list of filter values. The default is a comma (,).

Results

The operation returns the following results:

requestId

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

returnResult

Your bundle tasks in a table having the following columns: Bundle ID, Instance ID, Status, Start time, and Progress. The column fields are separated by colDelimiter and the rows by rowDelimiter.

Example:

```
bun-49789d20,i-d63cd5be complete,2009-08-25T10:04:55.000Z,null
```

```
bun-4f789d26,i-d63cd5be failed,2009-08-25T10:58:57.000Z,null
```



- Like all Amazon EC2 operations, the results might not be visible instantly.
- Terminated bundle tasks appear in this list for an undetermined interval of time after they are terminated.
- More details on the filters that can be applied on this operation can be found in the DescribeBundleTasks section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.
- If you assign a value of start-time or update-time to a filterName input, the timestamps supplied to the corresponding filterValues must have the local timezone and must use the format yyyy/MM/dd HH:mm:ss. In this case, you cannot use wildcards for filterValues.

Get Bundle Task Details

The **Get Bundle Task Details** operation returns details about a specified bundle task.

Inputs

All of the operation's inputs except the following are described in *Common Inputs in the Integration*

bundleId

The ID of the bundle task. If you specify a nonexistent ID, the operation fails.

Results

The operation returns the following results:

returnResult

The instance ID of the instance that is being bundled.

`startTime`

The time the task was launched.

`updateTime`

The time the status of the task was last updated.

`status`

The status of the task (for example, running, complete, or failed).

`progress`

The progress of the task (0% - 100%).

`bucket`

The S3 bucket where the image will be placed.

`prefix`

The name under which the image will be placed in the bucket.

`requestId`

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



- Like all Amazon EC2 operations, the results might not be visible instantly.
- Terminated bundle tasks can be queried for an undetermined interval of time after they are terminated.

EC2 Dashboard

The **EC2 Dashboard** flow displays the EC2 dashboard information: the number of instances running or stopped, elastic IPs, EBS volumes, EBS snapshots, key pairs, and security groups.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`notifyMethod`

The method of notification. The valid values are Write to File, Email, Display, and None.

Results

The flow returns the following results:

`returnResult`

The dashboard information.

Tags

Create Tags

The **Create Tags** operation adds or overwrites one or more tags for the specified resource or resources. If one of the specified resources already has a tag with a key specified in one of the `tagKey` inputs, that tag's value is overwritten with the value of the corresponding `tagValue` input. Otherwise, a new tag with the specified `tagKey` and `tagValue` is created for the resource.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`resourceId1`

The ID of a resource to tag (for example, i-78248215).

`resourceId[number]`

To tag more than one resource, you must add extra `resourceId` inputs for each resource using the naming convention `resourceId[number]` where you replace `[number]` with increasing numbers (for example, `resourceId2`, `resourceId3`).

`tagKey1`

The key for a tag. Tag keys are case sensitive and must be unique per resource (for example, `Owner`).

`tagValue1`

The value for a tag. If you do not specify a value for this input, the corresponding tag will not have a value. Tag values are also case sensitive (for example, `DbAdmin`).

`tagKey[number]`

To create more than one tag, you must add extra `tagKey` inputs (there must be a pair of `tagKey` and `tagValue` inputs for each tag) using the naming convention `tagKey[number]` where you replace `[number]` with increasing numbers (for example, `tagKey2`, `tagKey3`).

`tagValue[number]`

To create more than one tag, you must add extra `tagValue` inputs (there must be a pair of `tagKey` and `tagValue` inputs for each tag) using the naming convention `tagValue[number]` where you replace `[number]` with increasing numbers (for example, `tagValue2`, `tagValue3`).

Results

The operation returns the following results:

`returnResult`

Returns true if the operation is successful or false if it fails.

`requestId`

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



- Like all Amazon EC2 operations, the results might not be visible instantly.
- Each resource can have a maximum of ten tags.
- Each of the tags you specify using pairs of `tagKey[number]` and `tagValue[number]` input values is added to each of the resources specified through the `resourceId[number]` input values.

Delete Tags

The **Delete Tags** operation deletes a specific set of tags from a specific set of resources.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`resourceId1`

The ID of a resource from which to delete a tag (for example, i-78248215).

`resourceId[number]`

To delete tags from more than one resource, you must add extra `resourceId` inputs for each resource using the naming convention `resourceId[number]` where you replace `[number]` with increasing numbers (for example, `resourceId2`, `resourceId3`).

`tagKey1`

The key for a tag to delete (for example, Owner).

`tagValue1`

The value for a tag to delete. If you do not specify a value for this input, the corresponding `tagKey` input value is deleted only if its value is an empty string. If you assign the value `allvalues` to this input, all tags with the corresponding key are deleted regardless of their values (for example, DbAdmin).

`tagKey[number]`

To delete more than one tag, you must add extra `tagKey` inputs (there must be a pair of `tagKey` and `tagValue` inputs for each tag) using the naming convention `tagKey[number]` where you replace `[number]` with increasing numbers (for example, `tagKey2`, `tagKey3`).

`tagValue[number]`

To delete more than one tag, you must add extra `tagValue` inputs (there must be a pair of `tagKey` and `tagValue` inputs for each tag) using the naming convention `tagValue[number]` where you replace `[number]` with increasing numbers (for example, `tagValue2`, `tagValue3`).

Results

The operation returns the following results:

`returnResult`

Returns true if the operation is successful or false if it fails.

`requestId`

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



- Like all Amazon EC2 operations, the results might not be visible instantly.
- Each of the tags you specify using pairs of `tagKey[number]` and `tagValue[number]` input values is deleted from each of the resources specified through the `resourceId[number]` input values if the specified resources have the tags specified as inputs in the operation.

Enumerate Tags

The **Enumerate Tags** operation retrieves a list of your Amazon tags.

Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#)

`filterName1`

The first filter name for the operation. The valid values are key, value, resource-id, and resource-type (for example, key).

`filterValues1`

A first value or list of values for the corresponding `filterName` input value (for example, Owner,Stack).

`filterName[number]`

To specify more than one filter, you must add extra `filterName` inputs (there must be a pair of `filterName` and `filterValues` inputs for each filter) using the naming convention `filterName[number]` where you replace `[number]` with increasing numbers (for example, `filterName2`, `filter Name3`).

`filterValues[number]`

To specify more than one filter, you must add extra `filterValues` inputs (there must be a pair of `filterName` and `filterValues` inputs for each filter) using the naming convention `filterValues[number]` where you replace `[number]` with increasing numbers (for example, `filterValues1`, `filterValues2`).

`delimiter`

The delimiter used to separate items in the list of `filterValues` input values. The default value is a comma (,).

`rowDelimiter`

The delimiter used to separate the rows in the results table. The default value is the line separator for the current operating system.

`colDelimiter`

The delimiter for the columns in the results table. The default value is a semicolon (;).

Results

The operation returns the following results:

`returnResult`

Your Amazon tags in a table with the columns resource type, resource id, tag key, and tag value. Each row corresponds to a different tag. If a column is empty, it contains a null character.

`requestId`

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

Example:

`instance;i-1c892b71;Owner;DbAdmin`

`instance;i-1c892b71;Stack;Production`

`instance;i-04278169;Stack;Test`



- Like all Amazon EC2 operations, the results might not be visible instantly.
- In the case where no tags exist, the operation goes to success but the `returnResult` is empty.
- If you specify a value of resource-type for a `filterName` input, the valid values for the corresponding `filterValues` input are customer-gateway, dhcp-options, image, instance, snapshot, spot-instances-request, subnet, volume, vpc, vpn-connection, and vpn-gateway.
- You can find more details on the filters that can be applied on this operation in the `DescribeTags` section at <http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>.

4 Troubleshooting

This section includes the following topics:

- [Troubleshooting Overview](#)
- [General Troubleshooting Procedures and Tools](#)
- [Error Messages](#)

Troubleshooting Overview

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 EC2 integration, 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.

Error Messages

This section lists the error messages you may receive while using this integration.

Attribute productCodes can not be reset currently.

AWS was not able to validate the provided access credentials.

Either both proxy port and proxy host must be null or none of them.

Failed to open HTTP connection.

fromPort was invalid. Must be numerical and between 0 and 65535.

If attribute=productCodes, productCode must be set!

Invalid attribute item value '###' for UserGroup item type. Error code: InvalidAMIAttributeItemValue.

Invalid attribute item value '###' for userId item type. Error code: InvalidAMIAttributeItemValue.

Invalid attribute.

Invalid bundle ids: ###. Error code: InvalidBundleID.NotFound.

Invalid id: '###' (expecting "ami-..."). Error code: InvalidAMIID.Malformed.

Invalid id: '###' (expecting "i-..."). Error code: InvalidInstanceID.Malformed.

Invalid manifest path: '###'. Error code: InvalidManifest.

Invalid operationType.

Invalid user id: '###'. Error code: InvalidUserID.Malformed.

Must supply either only user/group pair permission inputs (SourceSecurityGroupName and SourceSecurityGroupOwnerId) or only CIDR IP permission inputs (IpProtocol, FromPort, ToPort, CidrIp).

Not authorized for image: ###. Error code: AuthFailure.

Possible values for input "force" are true or false.

The address '###' does not belong to you. Error code: AuthFailure.

The AMI ID '###' does not exist. Error code: InvalidAMIID.NotFound.

The following input must be a valid integer: size.

The given CIDR IP address is invalid. It must have the following format: [0-255].[0-255].[0-255].[0-32].

The given IP address is invalid. It must have the following format: [0-255].[0-255].[0-255].

The given IpProtocol is invalid. Must be one of 'tcp', 'udp' or 'icmp'.

The instance ID '###' does not exist. Error code: InvalidInstanceID.NotFound.

The instance IDs '###' do not exist. Error code: InvalidInstanceID.NotFound.

The keypair '###' already exists. Error code: InvalidKeyPair.Duplicate.

The permission '###' has already been authorized on the specified group. Error code: InvalidPermission.Duplicate.

The request must contain the parameter KeyName. Error code: MissingParameter.

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

The security group '###' already exists. Error code: InvalidGroup.Duplicate.

The security group '###' does not exist. Error code: InvalidGroup.NotFound.

The security group '###' is reserved. Error code: InvalidGroup.Reserved.

The snapshot '###' does not exist. Error code: InvalidSnapshot.NotFound.

The volume '###' does not exist. Error code: InvalidVolume.NotFound.

The volume size must be an integer between 1 and 1024 inclusive.

The zone '###' does not exist. Error code: InvalidZone.NotFound.

toPort was invalid. Must be numerical and between 0 and 65535.

Value (###) for parameter instance is invalid. Expected: 'i-...'. Error code: InvalidParameterValue.

Value (###) for parameter snapshotId is invalid. Expected: 'snap-...'. Error code: InvalidParameterValue.

Value (###) for parameter volumeId is invalid. Expected: 'vol-...'. Error code: InvalidParameterValue.

5 Security

This section includes the following topics:

- [About Amazon EC2 Security](#)

About Amazon EC2 Security

This section describes how security is handled by the Amazon EC2 integration.

The EC2 integration uses the EC2 Query API. To perform its task, each EC2 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 EC2 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 EC2 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 use 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`.

6 OO Tools

This section includes the following topic:

- [OO Tools You Can Use with the Amazon EC2 Integration](#)

OO Tools You Can Use with the Amazon EC2 Integration

Following are OO tools that you can use with the Amazon EC2 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 Operations Orchestration home folder in /Studio/tools/.