FUJITSU

FUJITSU Software Enterprise Service Catalog Manager V17.7

Amazon Web Services Integration

June 2018

Contents

About this Manual

This manual describes the integration of the Amazon Elastic Compute Cloud Web service (Amazon EC2), a major component of Amazon Web Services (AWS), with) FUJITSU Software Enterprise Service Catalog Manager, hereafter referred to as ESCM.

This manual is structured as follows:

Chapter	Description
Introduction on page 7	Provides an overview of the ESCM AWS integration, the components involved, and the supported usage scenarios.
Creating and Publishing Services on page 10	Describes how to create and publish services for AWS in ESCM.
Using AWS Services in ESCM on page 12	Describes how to provision and deprovision virtual servers in AWS through services in ESCM.
Administrating the AWS Integration on page 14	Describes administration tasks related to the ESCM AWS integration.
Controller Configuration Settings on page 17	Describes the configuration settings for the AWS integration software.
Service Parameters and Operations on page 19	Describes the technical service parameters and service operations which are supported by the AWS service controller.

Readers of this Manual

This manual is intended for operators who want to offer virtual servers controlled by AWS through services on a marketplace provided by ESCM. It assumes that you have access to an existing ESCM installation and that you have an AWS account. In addition, you should have basic knowledge of Amazon EC2 and you should be familiar with the concepts and administration of ESCM.

Notational Conventions

This manual uses the following notational conventions:

Add	Names of graphical user interface elements.
`init`	System names, for example command names and text that is entered from the keyboard.
`<variable>`	Variables for which values must be entered.
`[option]`	Optional items, for example optional command parameters.
`one | two`	Alternative entries.
`{one | two}`	Mandatory entries with alternatives.

Abbreviations

This manual uses the following abbreviations:

Amazon EC2	Amazon Elastic Compute Cloud
AMI	Amazon Machine Image
APP	Asynchronous Provisioning Platform
AWS	Amazon Web Services
ESCM	Enterprise Service Catalog Manager
DBMS	Database Management System
IaaS	Infrastructure as a Service
IdP	SAML Identity Provider
SAML	Security Assertion Markup Language
STS	Security Token Service
WSDL	Web Services Description Language
WSIT	Web Services Interoperability Technologies

Available Documentation

The following documentation on ESCM is available:

- *Overview:* A PDF manual introducing ESCM. It is written for everybody interested in ESCM and does not require any special knowledge.
- *Operator's Guide:* A PDF manual for operators describing how to administrate and maintain ESCM.
- *Technology Provider's Guide:* A PDF manual for technology providers describing how to prepare applications for usage in a SaaS model and how to integrate them with ESCM.
- *Supplier's Guide:* A PDF manual for suppliers describing how to define and manage service offerings for applications that have been integrated with ESCM.
- *Reseller's Guide:* A PDF manual for resellers describing how to prepare, offer, and sell services defined by suppliers.
- *Broker's Guide:* A PDF manual for brokers describing how to support suppliers in establishing relationships to customers by offering their services on a marketplace.
- *Marketplace Owner's Guide:* A PDF manual for marketplace owners describing how to administrate and customize marketplaces in ESCM.
- *OpenStack Integration:* A PDF manual for operators describing how to offer and use virtual systems controlled by OpenStack through services in ESCM.
- *Amazon Web Services Integration:* A PDF manual for operators describing how to offer and use virtual servers controlled by the Amazon Elastic Compute Cloud Web service through services in ESCM.
- *VMware vSphere Integration:* A PDF manual for operators describing how to offer and use virtual machines provisioned on a VMware vSphere server through services in ESCM.

- *Online Help:* Online help pages describing how to work with the administration portal of ESCM. The online help is intended for and available to everybody working with the administration portal.

1 Introduction

Enterprise Service Catalog Manager (ESCM) is a set of services which provide all business-related functions and features required for turning on-premise applications and tools into 'as a Service' (aaS) offerings and using them in the Cloud. This includes ready-to-use account and subscription management, online service provisioning, billing and payment services, and reporting facilities.

Amazon Web Services (AWS) is a collection of remote computing services that together make up a Cloud computing platform offered by Amazon. Amazon Elastic Compute Cloud (Amazon EC2) is one of the central Web services of AWS. It provides computing capacities in the Cloud and allows you to quickly scale these capacities as your computing requirements change.

When integrating applications with ESCM, the instance provisioning can be done in two provisioning modes: synchronous or asynchronous mode.

Asynchronous provisioning is required if provisioning operations take a long time because long-running processes or manual steps are involved. This is the case, for example, when provisioning virtual machines on a virtual machine server. ESCM supports the integration of such applications with its asynchronous provisioning platform (APP). This is a framework which provides a provisioning service as well as functions, data persistence, and notification features which are always required for integrating applications in asynchronous mode.

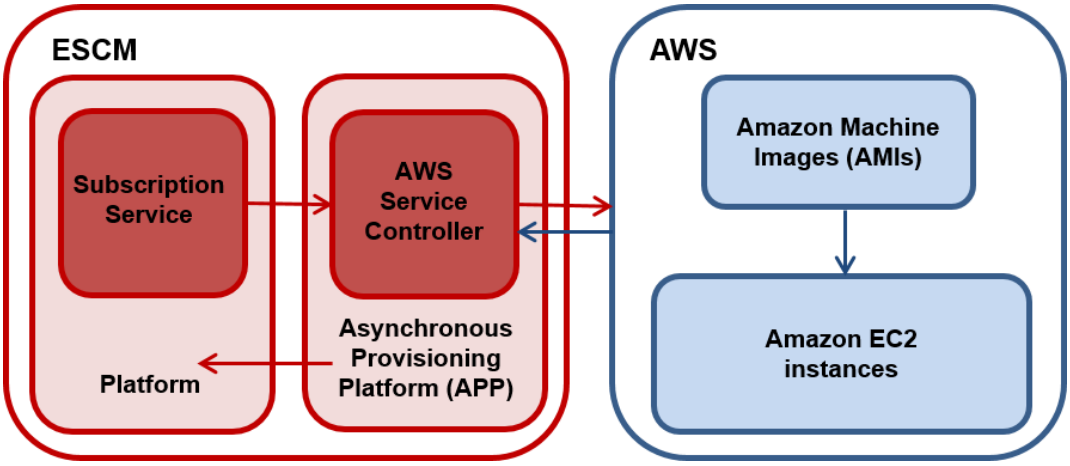
The integration of AWS with ESCM provides for an Infrastructure as a Service (IaaS) solution that leverages the features of both products: Through services, which are published on a marketplace in ESCM, users can request and use virtual servers in Amazon EC2. The usage costs can be calculated and charged by means of the ESCM billing and payment services.

When deploying the `oscm-app` container provided with ESCM, an AWS service controller is preconfigured and registered with the asynchronous provisioning platform (APP). This service controller can be used for integrating AWS services with ESCM.

The AWS service controller includes all components required for connecting an existing ESCM installation with AWS. This manual describes how to create and use services for Amazon EC2 on an ESCM marketplace.

1.1 Components Involved in the AWS Integration

The following picture provides an overview of the main components involved in the integration of ESCM and AWS:



In ESCM, customer subscriptions are managed by means of the **Subscription service**. When a customer creates or terminates a subscription for an Amazon EC2 instance in AWS, the Subscription service asynchronously triggers the corresponding actions in AWS through the **Asynchronous Provisioning Platform (APP)** and the **AWS service controller**: Virtual servers are created or deleted in AWS.

APP is a framework which provides a provisioning service, an operation service, as well as functions, data persistence, and notification features which are required for integrating applications with ESCM in asynchronous mode. The actual communication with the applications is carried out by service controllers.

Amazon EC2 allows customers to provision and use virtual servers on which to run their applications. Each virtual server is based on an Amazon Machine Image (AMI). An AMI serves as the basic unit of deployment for services delivered with Amazon EC2. AWS customers can either request pre-configured AMIs or they can create their own images. They can provision their images with a variety of operating systems and load them with custom application environments.

When APP is deployed, an AWS service controller is also deployed, registered, and initialized.

1.2 Usage Scenarios

The ESCM AWS integration supports the following usage scenarios:

- **Provisioning of a virtual server**: When a customer subscribes to a corresponding service on an ESCM marketplace, the service controller triggers AWS to create an Amazon EC2 instance based on a specific AMI.
- **Automatic execution of user-specific configuration data**: When a customer subscribes to a corresponding service on an ESCM marketplace, he has the option of passing user-specific configuration data to the Amazon EC2 instance to be provisioned. The data can be used to modify the static information defined in the underlying AMI. Thus, the customer can, for example, perform automated configuration tasks or run scripts.

- **Starting and stopping a virtual server**: A customer can explicitly start and stop an Amazon EC2 instance by executing a service operation at the corresponding subscription.
- **Deletion of a virtual server**: When a customer terminates a subscription for an Amazon EC2 instance, the service controller triggers AWS to delete the instance. The subscription is terminated in ESCM independent of whether the deletion is successful in AWS.

In Amazon EC2, the virtual servers created for ESCM subscriptions are managed in the same way as other virtual servers. They can be viewed and monitored with the available AWS tools.

Modifying a subscription and thereby triggering modifications of the virtual server in AWS is not supported. For more details on the supported usage scenarios, refer to *Using AWS Services in ESCM* on page 12.

2 Creating and Publishing Services

The following sections describe how to create and publish services in ESCM by means of which customers can request and use virtual servers in AWS.

2.1 Prerequisites and Preparation

The following prerequisites must be fulfilled before you can create and publish services in ESCM:

- To create technical services for the AWS integration in ESCM, you must have access to ESCM as a technology manager. You must be a member of the technology provider organization responsible for the AWS service controller as specified in the configuration settings for the installation.
- In AWS, appropriate AMIs for virtual servers must exist, to which the technical services in ESCM can be mapped. The AWS user specified in the configuration settings for the installation must have the necessary credentials to create and configure virtual servers based on these AMIs.
- To create marketable services for the AWS integration in ESCM, you must have access to ESCM as a service manager of an organization with the supplier role. This may be the same organization as the technology provider organization or a different one.
- To publish your marketable services, you must have access to an appropriate marketplace in ESCM in your service manager role.

2.2 Creating Technical Services

The first step in providing ESCM services for AWS is to create one or more technical services.

Proceed as follows:

1. Define one or more technical services in an XML file.

   As a basis, you can use the technical service sample provided on *https://github.com/servicecatalog/oscm/tree/master/oscm-app-aws/resources*.

   In the technical service definition, be sure to specify:

   - The asynchronous provisioning type
   - The direct access type
   - Service parameters which represent the AMIs defined in Amazon EC2. For details on the supported service parameters, refer to *Service Parameters and Operations* on page 19.
2. Log in to the ESCM administration portal with your technology manager account.
3. Import the technical services you created and appoint one or more supplier organizations for them.

For details on these steps, refer to the *Technology Provider's Guide* and to the online help of ESCM.

2.3 Creating and Publishing Marketable Services

As soon as the technical services for the AWS integration exist in ESCM, you can define and publish marketable services based on them. Your cost calculation for the services should include any external costs for operating the virtual servers in Amazon EC2.

Proceed as follows:

1. Log in to the ESCM administration portal with your service manager account.
2. Define one or more marketable services based on the technical services you created for AWS.
3. Define price models for your marketable services.
4. Publish the services to a marketplace.

For details on these steps, refer to the *Supplier's Guide* and to the online help of ESCM.

3 Using AWS Services in ESCM

The following sections describe how users can subscribe to and work with the services you have created for AWS in ESCM. You will find details of the supported usage scenarios outlined in *Usage Scenarios* on page 8.

3.1 Subscribing to Services

Users of customer organizations can subscribe to the services you have created for AWS on the marketplace where you have published them. This results in the provisioning of a virtual server in Amazon EC2, as defined in the underlying technical service.

To enable the provisioning of a virtual server, the customer has to enter the name of the key pair of the virtual server when subscribing to the corresponding service in ESCM. The key pair name and the associated private key are used to securely access the Amazon EC2 instance. For details on creating key pairs, refer to the user documentation of Amazon Web Services.

In addition, the customer has to enter a name for the virtual server when subscribing to the corresponding service. The technical service may specify a prefix which is preprended to this name, as well as a pattern against which the name is checked before the provisioning operation is started.

Depending on the parameters defined for the technical service, the customer can choose from different options to configure the virtual server to be provisioned.

The provisioning operations are carried out in asynchronous mode. As long as the provisioning is not complete, the status of the subscription is **pending**. The status changes to **ready** as soon as the provisioning has been finished successfully.

As soon as the provisioning is complete, the users assigned to the subscription can access the virtual server provided by AWS using the IP address indicated in the subscription details on the marketplace in ESCM. The users can access the virtual server according to the connection processes specified by Amazon. For details, refer to the user documentation of Amazon Web Services.

3.2 Executing User-Specific Configuration Data

When an instance is provisioned in Amazon EC2, a customer has the option of passing user-specific configuration data to the instance. The data can be used to perform automated configuration tasks or run scripts. Customers can pass two types of user data to Amazon EC2 instances: shell scripts and `cloud-init` directives. They can also pass this data as plain text, as a file, or as base64-encoded text for API calls.

To access user data scripts or `cloud-init` directives, the technology provider must enter the URL pointing to the scripts or directives into the definition of the technical service. The URL must be accessible for APP.

For details on executing user data, refer to the user documentation of Amazon Web Services.

3.3 Executing Service Operations

Customers can explicitly start and stop a virtual server in AWS from ESCM. To do this, they execute the appropriate service operation from the subscription for the virtual server:

- **Start**: Starts the virtual server if it was stopped.
- **Stop**: Stops the virtual server if it was started.

As a prerequisite, the service operations must be defined in the technical service underlying the subscribed service.

3.4 Terminating Subscriptions

A customer can at any time terminate a subscription for a virtual machine in AWS.

AWS is triggered to delete the virtual server. The subscription is terminated in ESCM independent of whether the deletion is successful in AWS. Note, however, that the subscription name cannot be re-used before the deletion has been completed in AWS.

4 Administrating the AWS Integration

The following sections describe administration tasks you may need to perform in your role as an operator of the AWS integration software.

4.1 Controlling the Provisioning Process

The AWS integration provides you with the following feature for controlling the provisioning and deprovisioning of virtual servers:

In the definition of the technical services for AWS, you can specify the `MAIL_FOR_COMPLETION` parameter. This is an address to which emails are to be sent describing manual steps required to complete an operation.

If you specify this parameter, the AWS service controller interrupts the processing of each operation before its completion and waits for a notification about the execution of a manual action. This notification consists in opening the link given in the email.

Omit the `MAIL_FOR_COMPLETION` parameter if you do not want to interrupt the processing.

4.2 Handling Problems in the Provisioning Process

If the provisioning of a virtual server fails on the AWS side or if there are problems in the communication between the participating systems, the corresponding subscription in ESCM remains pending. The AWS service controller informs the technology managers of its responsible technology provider organization by email of any incomplete provisioning or delete operation in AWS.

You can then take the appropriate actions to solve the problem in AWS or in the communication. For example, you could remove an incomplete virtual server, or you could restore a missing connection.

After solving the problem, the AWS integration components and ESCM need to be synchronized accordingly. You do this by triggering a corresponding action in the APP component. Proceed as follows:

1. Work as a technology manager of the technology provider organization responsible for the AWS service controller.
2. Invoke the instance status interface of APP for the service controller of the application by opening it's URL in a Web browser.

   The access URL has the following format:

   `https://<hostname.fqdn>:<port>/oscm-app/controller/?controllerid=ess.aws`

   `<hostname.fqdn>` is the name and the fully qualified domain name of the machine where the `oscm-app` container has been deployed, `<port>` is the port to address the machine (default: `8881`), `oscm-app/controller/?controllerid=ess.aws` is the default context root of the service controller and cannot be changed.

   The Web page shows all subscriptions for the application, including detailed information such as the customer organization, the ID of the related application instance, and the provisioning status.
3. Find the subscription for which you solved the problem in the most recent provisioning or delete operation.

4. In the **Action** column, select the action for the AWS integration components to execute next. Possible actions are the following:
   - `RESUME` - to resume the processing of a provisioning operation in APP which was suspended.
   - `SUSPEND` - to suspend the processing of a provisioning operation in APP, for example when AWS does not respond.
   - `UNLOCK` - to remove the lock for an AWS instance in APP.
   - `DELETE` - to terminate the subscription in ESCM and remove the instance in APP, but keep the virtual server in AWS for later use. The service manager role is required for this action.
   - `DEPROVISION` - to terminate the subscription in ESCM, remove the instance in APP, and delete the virtual server in AWS. The service manager role is required for this action.
   - `ABORT_PENDING` - to abort a pending provisioning operation in ESCM. ESCM is notified to roll back the changes made for the subscription and return it to its previous state. In AWS, no actions are carried out.
   - `COMPLETE_PENDING` - to complete a pending provisioning operation in ESCM. ESCM is notified to complete the changes for the subscription and set the subscription status to **ready** (or **suspended** if it was suspended before). This is possible only if the operations of the service controller are already completed.
5. Click **Execute** to invoke the selected action.

The instance status interface provides the following additional functionality that is useful for problem-solving purposes:

- You can display service instance details for each subscription by clicking the corresponding entry in the table. This displays all subscription-related information that is stored in the `bssapp` database.

4.3 Updating Service Controller Settings in the Database

During deployment, several configuration settings are written to the `bssapp` database. This configuration is used for the initial setup of the AWS service controller and its registration with APP. It is up to the platform operator for taking care that the initial settings are correct.

A technology provider can define service parameters in the technical service definition. If such a parameter has the same ID as a controller configuration setting stored in the APP database, it overrules the configuration setting in the database when the marketable service based on such a technical service is subscribed to. By default the values in the controller configuration settings are used. Refer to the *Technology Provider's Guide* for details on defining technical services.

In addition, a supplier can define custom attributes for subscriptions and for customers. If such an attribute has the same ID as a controller configuration setting stored in the APP database as well as a corresponding technical service parameter, it overrules the technical service parameter as well as the configuration setting in the database when the marketable service based on such a technical service is subscribed to.

The controller configuration settings are evaluated as follows:

1. Configuration setting as stored in the APP database.
2. Technical service parameter. If defined, it overrules 1.
3. Custom attribute for customer. If defined, it overrules 1. and 2.
4. Custom attribute for subscription. If defined, it overrules 1. and 2. and 3.

To update the controller settings in the APP database:

1. Invoke the instance status interface of APP for the AWS service controller of the application by opening it's URL in a Web browser.

   The access URL has the following format:

   `https://<hostname.fqdn>:<port>/oscm-app-aws`

   `<hostname.fqdn>` is the name and the fully qualified domain name of the machine where the `oscm-app` container has been deployed, `<port>` is the port to address the machine (default: `8881`), `oscm-app-aws` is the default context root of the service controller and cannot be changed.

2. Log in with the ID and password of the user specified in the configuration settings for the AWS service controller by the platform operator in the `BSS_USER_ID` and `BSS_USER_PWD` configuration settings, or as another technology manager registered for the same organization.
3. The following settings can be changed:
   - **User ID**: The identifier of the user responsible for the AWS service controller.
   - **User Key**: The user key for accessing ESCM. You receive this key with the confirmation email for your user account. The user must have the technology manager role in ESCM and belong to the technology provider organization responsible for the service controller.

     It is recommended that the user account is used only for carrying out actions on behalf of the service controller in ESCM.

   - **Password**: The password of the user for accessing ESCM.
   - As the technology provider who is responsible for the service controller, you need to have a target where the Amazon EC2 instances will be created, and to recover the so-called access keys. Therefore, you need an AWS account. For details about creating your AWS account, refer to the user documentation of Amazon Web Services.

     An AWS account consists of two keys, the **access key ID** and the **secret access key**. These keys are used to authenticate API calls to Amazon EC2, and must be saved in the APP database.

     In the controller-specific settings, you have to specify these two keys.
     - **Access Key ID**: The access key ID which is part of the AWS account.
     - **Secret Access Key**: The secret access key which is part of the AWS account.

4. Save the settings.

Appendix A: Controller Configuration Settings

This appendix describes the controller configuration settings as stored in the `bssapp` database. For details on how to update them, refer to *Updating Service Controller Settings in the Database* on page 15.

CONTROLLER_ID

`CONTROLLER_ID=ess.aws`

The identifier of the service controller. This setting cannot be changed.

BSS_ORGANIZATION_ID

`BSS_ORGANIZATION_ID=<organizationID>`

The ID of the organization in ESCM which is responsible for the service controller. The organization must have the technology provider role. It is created and initially assigned to the AWS service controller by the platform operator.

BSS_USER_ID

`BSS_USER_ID=<userId>`

The identifier of the user specified in `BSS_USER_KEY` for accessing ESCM.

BSS_USER_KEY

`BSS_USER_KEY=<userKey>`

The user key for accessing ESCM.

The user key for accessing ESCM. You receive this key with the confirmation email for your user account. The user must have the technology manager role in ESCM and belong to the technology provider organization responsible for the service controller.

It is recommended that the user account is used only for carrying out actions on behalf of the service controller in ESCM.

BSS_USER_PWD

`BSS_USER_PWD=_crypt:<password>`

The password of the user for accessing ESCM.

ACCESS_KEY_ID_PWD

`ACCESS_KEY_ID_PWD=_crypt:<accessKeyID>`

The identifier of the access key for the AWS account.

A technology provider who is responsible for creating technical services for appropriate AMIs needs to have an AWS account to create Amazon EC2 instances. For details about creating an AWS account, refer to the user documentation of Amazon Web Services.

Together with the secret access key, the access key ID is used to authenticate API calls to Amazon EC2.

SECRET_KEY_PWD

`SECRET_KEY_PWD=_crypt:<secretAccessKey>`

The secret access key for the AWS account.

Together with the access key ID, the secret access key is used to authenticate API calls to Amazon EC2.

Appendix B: Service Parameters and Operations

The following sections describe the technical service parameters and service operations which are supported by the AWS service controller.

You find a sample service on GitHub:
https://github.com/servicecatalog/oscm/tree/master/oscm-app-aws/resources.

Service Parameters

The AWS service controller supports the parameters below.

Note: All parameters defined in the technical service definition must be one-time parameters, since the modification of parameters is not supported. Be sure to set their `modificationType` to `ONE_TIME`.

APP_CONTROLLER_ID

Mandatory. The ID of the service controller as defined in its implementation. The ID is set during the installation of the AWS integration software.

Default (must not be changed): `ess.aws`

DISK_SIZE

Optional. The maximum disk size of the virtual server to be instantiated. The value is a number specifying the size in GB. If not specified, the default size for the selected type of server is used.

Users should not be able to enter a value for this parameter. This means the parameter should not be configurable for customers or, in case you specify more than one disk size, have fixed parameter options for selection.

Example: `3`

IMAGE_NAME

Mandatory. Name of the AMI which is the basis for virtual servers.

Users should not be able to enter a value for this parameter. This means the parameter should not be configurable for customers or, in case you specify more than one image, have fixed parameter options for selection.

Example: `ami-0d77397e`

INSTANCENAME

Mandatory. The name of the virtual server to be instantiated. This name must be specified by customers when they subscribe to a corresponding service. The string given in `INSTANCENAME_PREFIX` is prepended to the name. The name including the prefix must match the pattern given in `INSTANCENAME_PATTERN`.

As the instance name is stored as a tag on the Amazon EC2 instance, the maximum length is 255 Unicode characters.

Example: `MyServer`

INSTANCENAME_PATTERN

Mandatory. A regular expression specifying a pattern for the virtual server instance names entered by the users when they subscribe to a corresponding service. If the names do not match the pattern, the subscription is rejected.

The regular expression must be specified in the technical service definition.

Example: `.*{1,255}`

INSTANCENAME_PREFIX

Optional. A string to be prepended to the virtual server instance names entered by the users when they subscribe to a corresponding service.

The string must be specified in the technical service definition.

Example: `aws`

INSTANCE_TYPE

Mandatory. The type of the virtual server to be instantiated. Any valid Amazon EC2 instance type can be specified. In the sample technical service, the following types are defined:

- **t1.micro**
- **t1.small**
- **t1.medium**

Users should not be able to enter a value for this parameter. This means the parameter should not be configurable for customers or, in case you specify more than one type, have fixed parameter options for selection.

Example: `t1.micro`

KEY_PAIR_NAME

Mandatory. The key pair name of the virtual server to be instantiated.

The key pair name must be specified by the customer when subscribing to an AWS service. To log in to the instance, the customer must enter the key pair name and the associated private key.

For details on creating key pairs, refer to the user documentation of Amazon Web Services.

Example: `my-key-pair`

MAIL_FOR_COMPLETION

Optional. The address to which emails are to be sent that describe manual steps required to complete an operation. If you specify this parameter, the service controller interrupts the processing of each operation before its completion and waits for a notification about the execution of a manual action. Omit this parameter if you do not want to interrupt the processing.

Example: `info@company.com`

PUBLIC_IP

Optional. Specifies whether the virtual server to be instantiated is to be assigned a public IP address. The value can be `true` or `false`. If not specified, the default is `false`.

Example: `true`

REGION

Mandatory. The region where the data center hosting the virtual servers is located. Any valid region can be specified. In the sample technical service, the following regions are defined:

- **us-west-1** (US West (Northern California) Region)
- **us-west-2** (US West (Oregon) Region)
- **us-east-1** (US East (Northern Virginia) Region)

Users should not be able to enter a value for this parameter. This means the parameter should not be configurable for customers or, in case you specify more than one region, have fixed parameter options for selection.

Example: `us-east-1`

SECURITY_GROUP_NAMES

Optional. Comma-separated list of security group names for the virtual server to be instantiated.

A security group acts as a firewall that controls the traffic to an Amazon EC2 instance. An Amazon EC2 instance can be assigned to one or more security groups. For details on security groups, refer to the user documentation of Amazon Web Services.

Users should not be able to enter a value for this parameter. This means the parameter should not be configurable for customers or, in case you specify more than one security group, have fixed parameter options for selection.

Be aware that for the specification of security groups, the specification of the `SUBNET` parameter is mandatory. Otherwise the groups will be ignored.

Example: `MySecurityGroup`

SUBNET

Optional. The subnet to which the virtual server is to be assigned.

Users should not be able to enter a value for this parameter. This means the parameter should not be configurable for customers or, in case you specify more than one subnet, have fixed parameter options for selection.

If you specify security groups, you also need to specify the corresponding subnet. If no subnet parameter is specified, the AWS service controller ignores any specified security groups, and the service instance is created in a default subnet and a default security group is assigned.

Example: `subnet-a77430d0`

USERDATA_URL

Optional. URL to access the user data scripts or `cloud-init` directives for the automatic execution of user-specific configuration data. This URL must be accessible for APP.

Users should not be able to enter a value for this parameter. This means the parameter should not be configurable for customers or, in case you specify more than one URL, have fixed parameter options for selection.

Example: `https://127.0.0.1:8881/cloud-init/LAMP.script` (if the file was created under `<oscm-app>/cloud-init/LAMP.script`)

Service Operations for Virtual Servers

The AWS service controller supports the service operations below for virtual servers.

The `actionURL` for each operation is:

```
https://oscm-app:8881/oscm-app/webservices/oscm-app/oscm-app
         /org.oscm.app.v2_0.service.AsynchronousOperationProxy?wsdl
```

`<oscm-app>` and `<8881>` are the server and port of the container where the AWS integration software is deployed.

Note: If you provision a virtual server that does not support start and stop operations, make sure that you remove the service operations from the technical service definition.

START_VIRTUAL_SYSTEM

Starts a virtual server in AWS if it was stopped.

STOP_VIRTUAL_SYSTEM

Stops a virtual server in AWS if it was started.

Glossary

Administrator

A privileged user role within an organization with the permission to manage the organization's account and subscriptions as well as its users and their roles. Each organization has at least one administrator.

Application

A software, including procedures and documentation, which performs productive tasks for users.

Billing System

A system responsible for calculating the charges for using a service.

Broker

An organization which supports suppliers in establishing relationships to customers by offering the suppliers' services on a marketplace, as well as a privileged user role within such an organization.

Cloud

A metaphor for the Internet and an abstraction of the underlying infrastructure it conceals.

Cloud Computing

The provisioning of dynamically scalable and often virtualized resources as a service over the Internet on a utility basis.

Customer

An organization which subscribes to one or more marketable services in ESCM in order to use the underlying applications in the Cloud.

Infrastructure as a Service (IaaS)

The delivery of computer infrastructure (typically a platform virtualization environment) as a service.

Marketable Service

A service offering to customers in ESCM, based on a technical service. A marketable service defines prices, conditions, and restrictions for using the underlying application.

Marketplace

A virtual platform for suppliers, brokers, and resellers in ESCM to provide their services to customers.

Marketplace Owner

An organization which holds a marketplace in ESCM, where one or more suppliers, brokers, or resellers can offer their marketable services.

Marketplace Manager

A privileged user role within a marketplace owner organization.

Operator

An organization or person responsible for maintaining and operating ESCM.

Organization

An organization typically represents a company, but it may also stand for a department of a company or a single person. An organization has a unique account and ID, and is assigned one or more of the following roles: technology provider, supplier, customer, broker, reseller, marketplace owner, operator.

Organizational Unit

A set of one or more users within an organization representing, for example, a department in a company, an individual project, a cost center, or a single person. A user may be assigned to one or more organizational units.

OU Administrator

A privileged user role within an organization allowing a user to manage the organizational units for which he has been appointed as an administrator, and to create, modify, and terminate subscriptions for these units.

Payment Type

A specification of how a customer may pay for the usage of his subscriptions. The operator defines the payment types available in ESCM; the supplier or reseller determines which payment types are offered to his customers, for example payment on receipt of invoice, direct debit, or credit card.

Platform as a Service (PaaS)

The delivery of a computing platform and solution stack as a service.

Price Model

A specification for a marketable service defining whether and how much customers subscribing to the service will be charged for the subscription as such, each user assigned to the subscription, specific events, or parameters and their options.

Reseller

An organization which offers services defined by suppliers to customers applying its own terms and conditions, as well as a privileged user role within such an organization.

Role

A collection of authorities that control which actions can be carried out by an organization or user to whom the role is assigned.

Seller

Collective term for supplier, broker, and reseller organizations.

Service

Generally, a discretely defined set of contiguous or autonomous business or technical functionality, for example an infrastructure or Web service. ESCM distinguishes between technical services and marketable services, and uses the term "service" as a synonym for "marketable service".

Service Manager

A privileged user role within a supplier organization.

Standard User

A non-privileged user role within an organization.

Software as a Service (SaaS)

A model of software deployment where a provider licenses an application to customers for use as a service on demand.

Subscription

An agreement registered by a customer for a marketable service in ESCM. By subscribing to a service, the customer is given access to the underlying application under the conditions defined in the marketable service.

Subscription Manager

A privileged user role within an organization with the permission to create and manage his own subscriptions.

Supplier

An organization which defines marketable services in ESCM for offering applications provisioned by technology providers to customers.

Technical Service

The representation of an application in ESCM. A technical service describes parameters and interfaces of the underlying application and is the basis for one or more marketable services.

Technology Manager

A privileged user role within a technology provider organization.

Technology Provider

An organization which provisions applications as technical services in ESCM.