17.0.1

OpenStack Integration (GlassFish)

May 2017

Trademarks	LINUX is a registered trademark of Linus Torvalds. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Open Service Catalog Manager is a registered trademark of FUJITSU LIMITED. Oracle, GlassFish, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates. Apache Ant, Ant, and Apache are trademarks of The Apache Software Foundation. UNIX is a registered trademark of the Open Group in the United States and in other countries. VMware vSphere is a registered trademark of VMware in the United States and in other countries. Other company names and product names are trademarks or registered trademarks of their respective owners.

High Risk Activity

The Customer acknowledges and agrees that the Product is designed, developed and manufactured as contemplated for general use, including without limitation, general office use, personal use, household use, and ordinary industrial use, but is not designed, developed and manufactured as contemplated for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (hereinafter "High Safety Required Use"), including without limitation, nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system. The Customer shall not use the Product without securing the sufficient safety required for the High Safety Required Use. In addition, FUJITSU (or other affiliate's name) shall not be liable against the Customer and/or any third party for any claims or damages arising in connection with the High Safety Required Use of the Product.

Export Restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Contents

About this Manual 5

1 Introduction 8

1.1 Components Involved in the OpenStack Integration 8

1.2 Usage Scenarios 9

2 Installing the OpenStack Integration Software 10

2.1 Prerequisites and Preparation 10

2.1.1 OSCM and OpenStack 10

2.1.2 Hardware and Operating Systems 10

2.1.3 Java and Ant 10

2.1.4 Application Server 10

2.1.5 Relational Database 11

2.1.6 Mail Server 12

2.2 Installation 12

2.2.1 Preparing the Software and Setup Utilities 12

2.2.2 Configuring the OpenStack Integration 13

2.2.3 Setting up the Database 15

2.2.4 Setting up the Application Server Resources 16

2.2.5 Exchanging Certificates 17

2.3 Installing the OpenStack Service Controller in an Existing APP Environment 18

2.4 Update Installation 19

3 Creating and Publishing Services 22

3.1 Prerequisites and Preparation 22

3.2 Creating Technical Services 22

3.3 Creating and Publishing Marketable Services 23

4 Using OpenStack Services in OSCM 24

4.1 Subscribing to Services 24

4.2 Executing Service Operations 24

4.3 Terminating Subscriptions 25

5 Administrating the OpenStack Integration 26

5.1 Controlling the Provisioning Process 26
5.2 Handling Problems in the Provisioning Process 26
5.3 Handling Communication Problems Between APP and OSCM 27
5.4 Backup and Recovery 28
5.5 Updating Configuration Settings 29
5.6 Adapting the Log Configuration 30
6 Uninstallation 32
Appendix A Configuration Settings 33
A.1 GlassFish Configuration Settings 33
A.2 Database Configuration Settings 35
A.3 APP Configuration Settings 36
A.4 Controller Configuration Settings 38
Appendix B Service Parameters and Operations 41
Glossary 45

About this Manual

This manual describes the integration of OpenStack, an open-source cloud operating system, with Open Service Catalog Manager (OSCM).

This manual is structured as follows:

Chapter	Description
Introduction on page 8	Provides an overview of the OSCM OpenStack integration, the components involved, and the supported usage scenarios.
Installing the OpenStack Integration Software on page 10	Describes how to prepare and carry out the installation of the OpenStack integration software.
Creating and Publishing Services on page 22	Describes how to create and publish services for OpenStack in OSCM.
Using OpenStack Services in OSCM on page 24	Describes how to provision and deprovision virtual systems in OpenStack through services in OSCM.
Administrating the OpenStack Integration on page 26	Describes administration tasks related to the OSCM OpenStack integration software.
Uninstallation on page 32	Describes how to uninstall the OSCM OpenStack integration software.

The descriptions of services and usage scenarios also apply to FUJITSU Cloud Service K5, a cloud platform based on OpenStack and integrated with OSCM.

Readers of this Manual

This manual is intended for operators who want to offer virtual systems controlled by OpenStack through services on a marketplace provided by OSCM. It assumes that you have access to an existing OSCM installation and an OpenStack Web server. In addition, you should have basic knowledge of OpenStack and you should be familiar with the concepts and administration of OSCM.

Notational Conventions

This manual uses the following notational conventions:

Add	The names of graphical user interface elements like menu options are shown in boldface.
`init`	System names, for example command names and text that is entered from the keyboard, are shown in Courier font.
`<variable>`	Variables for which values must be entered are enclosed in angle brackets.
`[option]`	Optional items, for example optional command parameters, are enclosed in square brackets.
`one | two`	Alternative entries are separated by a vertical bar.

`{one | two}`	Mandatory entries with alternatives are enclosed in curly brackets.

Abbreviations

This manual uses the following abbreviations:

APP	Asynchronous Provisioning Platform
DBMS	Database Management System
IaaS	Infrastructure as a Service
IdP	SAML Identity Provider
OSCM	Open Service Catalog Manager
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 OSCM is available:

- *Overview:* A PDF manual introducing OSCM. It is written for everybody interested in OSCM and does not require any special knowledge.
- *Online Help:* Online help pages describing how to work with the administration portal of OSCM. The online help is intended for and available to everybody working with the administration portal.
- *Installation Guide (GlassFish):* A PDF manual describing how to install and uninstall OSCM. It is intended for operators who set up and maintain OSCM in their environment.
- *Operator's Guide:* A PDF manual for operators describing how to administrate and maintain OSCM.
- *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 OSCM.
- *Supplier's Guide:* A PDF manual for suppliers describing how to define and manage service offerings for applications that have been integrated with OSCM.
- *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 OSCM.
- *Developer's Guide:* A PDF manual for application developers describing the public Web services and application programming interfaces of OSCM and how to integrate applications and external systems with OSCM.

- *ServerView Resource Orchestrator Integration (GlassFish):* A PDF manual for operators describing how to offer and use virtual platforms and servers controlled by FUJITSU ServerView Resource Orchestrator through services in OSCM.
- *Amazon Web Services Integration (GlassFish):* 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 OSCM.
- *OpenStack Integration (GlassFish):* A PDF manual for operators describing how to offer and use virtual systems controlled by OpenStack through services in OSCM.
- *VMware vSphere Integration (GlassFish):* A PDF manual for operators describing how to offer and use virtual machines provisioned on a VMware vSphere server through services in OSCM.
- Javadoc and YAML documentation for the public Web services and application programming interfaces of OSCM and additional resources and utilities for application developers.

1 Introduction

Open Service Catalog Manager (OSCM) 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.

OpenStack is an open-source cloud operating system that controls large pools of processing, storage, and networking resources throughout a data center. It is managed by the OpenStack Foundation.

OpenStack allows users to deploy virtual systems which handle different tasks for managing a cloud environment. It makes horizontal scaling easy, which means that tasks which benefit from running concurrently can easily serve users by spinning up more resources. For example, a mobile application which needs to communicate with a remote server might be able to divide the work of communicating with each user across many different resources, all communicating with one another but scaling quickly and easily as the application gains more users.

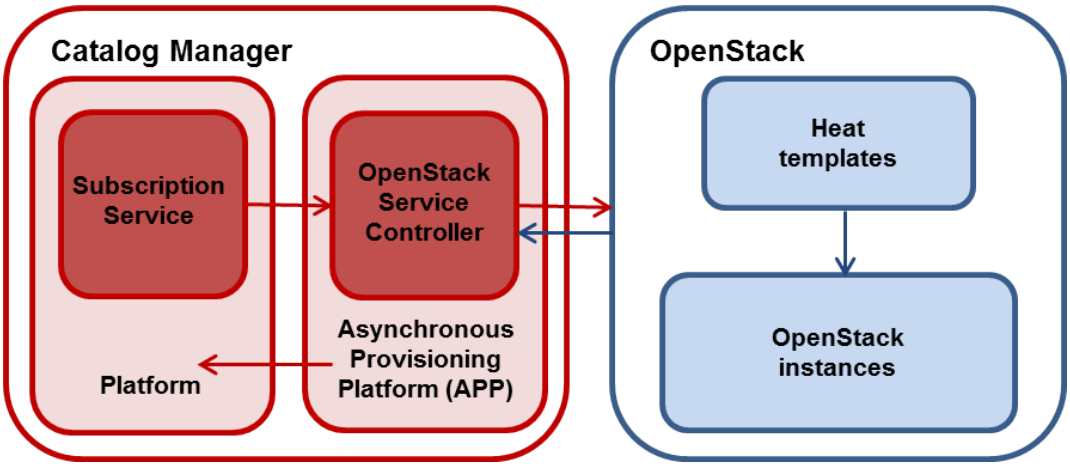
The integration of OpenStack with OSCM provides for an Infrastructure as a Service (IaaS) solution that leverages the features of both products: Through services, which are published on a marketplace in OSCM, users can request and use virtual systems in OpenStack. The usage costs can be calculated and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

The OpenStack integration package provided with OSCM includes all components required for connecting an existing OSCM installation with OpenStack. This manual describes how to deploy this package and how to create and use services for OpenStack on a OSCM marketplace.

FUJITSU Cloud Service K5 is a cloud platform based on OpenStack and integrated with OSCM. The descriptions of services and usage scenarios also apply to this platform.

1.1 Components Involved in the OpenStack Integration

The following picture provides an overview of the main components involved in the integration of OSCM and OpenStack:



In OSCM, customer subscriptions are managed by means of the **Subscription service**. When a customer creates or terminates a subscription for a virtual system in OpenStack, the

Subscription service asynchronously triggers the corresponding actions in OpenStack through the **Asynchronous Provisioning Platform (APP)** and the **OpenStack service controller**: Virtual systems are created or deleted in OpenStack.

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 OSCM in asynchronous mode. The actual communication with the applications is carried out by service controllers. APP and the OpenStack service controller are the main components that make up the OpenStack integration software.

Heat, the orchestration component provided by OpenStack, offers template-based mechanisms for describing cloud applications. Through both an OpenStack ReST API and a CloudFormation-compatible Query API, Heat provides compatibility with existing template formats. The integration of OpenStack with OSCM supports the AWS CloudFormation Template and the Heat Orchestration Template.

A template enables application developers to describe and automate the deployment of infrastructure, services, and applications. Collections of resources (e.g. networks, servers, or storage) can be deployed from a single template. The template serves as an orchestration document that details everything needed to carry out the orchestration. Once instantiated, the resources are also referred to as stacks.

Each APP installation supports one OpenStack service controller. This limitation can be overcome by installing APP several times to different application server domains. The need for more than one service controller may arise because multiple OpenStack accounts or technology provider organizations have to be used.

1.2 Usage Scenarios

The OSCM OpenStack integration supports the following usage scenarios:

- **Provisioning of a virtual system**: When a customer subscribes to a corresponding service on a OSCM marketplace, the service controller triggers OpenStack to create an OpenStack instance based on a specific template.
- **Suspending and resuming a virtual system**: A customer can explicitly suspend and resume an OpenStack instance by executing a service operation at the corresponding subscription.
- **Starting and stopping servers**: A customer can explicitly start and stop all the servers in an OpenStack instance by executing a service operation at the corresponding subscription.
- **Deletion of a virtual system**: When a customer terminates a subscription for an OpenStack instance, the service controller triggers OpenStack to delete the instance. The subscription is terminated in OSCM independent of whether the deletion is successful in OpenStack.

In OpenStack, the virtual systems created for OSCM subscriptions are managed in the same way as other virtual systems. They can be viewed and monitored with the available OpenStack tools.

Modifying a subscription and thereby triggering modifications of the virtual system in OpenStack is not supported. For more details on the supported usage scenarios, refer to *Using OpenStack Services in OSCM* on page 24.

2 Installing the OpenStack Integration Software

The following sections describe how to install and configure the OpenStack integration software as well as the preparations you need to take beforehand.

Installing the OpenStack integration software consists of installing APP and registering the OpenStack service controller.

If you already have a working APP installation in your environment, proceed as described in *Installing the OpenStack Service Controller in an Existing APP Environment* on page 18.

2.1 Prerequisites and Preparation

The following sections describe the prerequisites that must be fulfilled and the preparations you need to take before installing and deploying the OpenStack integration software.

2.1.1 OSCM and OpenStack

- You must have access to a fully functional OSCM installation. You can install the OpenStack integration software in the same environment or on a different server.
- You must have access to OSCM as an administrator and as a technology manager of an organization that has at least the technology provider role.
- You must have access to a fully functional OpenStack installation.

  The OpenStack integration requires the following API versions:

  Orchestration API (Heat) v1

  Compute API (Nova) v2.1

  Identity API (Keystone) v3

  For secure connections, the OpenStack integration supports the Transport Layer Security (TLS) protocol, v1.2.

2.1.2 Hardware and Operating Systems

The OpenStack integration software as a Java application does not rely on specific hardware or operating systems. It can be deployed on any platform supported by the application server and the database management system.

2.1.3 Java and Ant

The OpenStack integration software requires a Java Development Kit (JDK), version 8, 64 bit. Deployment with JDK 8, Update 80 has been tested and is recommended.

In order to be able to execute the installation scripts, you need to install the Apache Ant 1.8 (or higher) open source software. In the subsequent sections, `<ANT_HOME>` is the installation directory of Apache Ant.

2.1.4 Application Server

The OpenStack integration software must be deployed on an application server compatible with Java EE version 7. The following application server is supported:

Oracle GlassFish Server, version 4.1.1.

You can deploy the OpenStack integration software on the application server you use for OSCM. Alternatively, you can use a separate application server installation.

Be aware that when operating OSCM in SAML_SP mode, every Web service client must run in a separate domain of the application server.

Note: Before installing GlassFish, make sure that the `JAVA_HOME` environment variable points to a Java Development Kit (JDK), version 8, 64 bit.

Proceed as follows:

1. Install the application server as described in its documentation, and configure it as required by your environment.

Note: Make sure that the path of the GlassFish installation directory does not contain blanks.

2. After you have configured GlassFish, make a backup copy of the GlassFish installation.
3. Make sure that GlassFish is running in a JDK 8 environment. Also, make sure that no other applications (e.g. Tomcat) are running on your GlassFish ports.

The installation of the OpenStack integration software creates the `app-domain` domain in your application server. If required, you can change the domain name in the `glassfish.properties` file before starting the installation.

In the subsequent sections, `<GLASSFISH_HOME>` is the installation directory of GlassFish.

2.1.5 Relational Database

The OpenStack integration software stores its data in a relational database. The following database management system (DBMS) is supported:

PostgreSQL, version 9.1.12.

Install the DBMS as described in its documentation.

If required, you can use a separate machine for the OpenStack integration database.

Setup and Configuration

Edit the file

`<postgres_dir>/data/postgresql.conf`

as follows (`<postgres_dir>` is the PostgreSQL installation directory):

1. Set the `max_prepared_transactions` property value to `50`.
2. Set the `max_connections` property value to `210`.

   This property determines the maximum number of concurrent connections to the database server.

   Note the following: This setting is used in combination with the JDBC pool size settings for the domains on your application server. If you change the JDBC pool size, you might need to adapt the `max_connections` setting. Refer to the OSCM *Operator's Guide*, section *Tuning Performance*, for details.

3. Set the `listen_addresses` property value:

Specify the IP addresses of all application servers on which the database server is to listen for connections from client applications. If you use the entry `'*'`, which corresponds to all available IP addresses, you must be aware of possible security holes.

4. Save the file.

If you use a server name in all configuration files instead of `localhost` during installation, edit the file

`<postgres_dir>/data/pg_hba.conf`

as follows (`<postgres_dir>` is the PostgreSQL installation directory):

1. Add the IP address of the application server that is to host the OpenStack integration software.

   For example:

   ```
   host all all 123.123.12.1/32 md5
   ```

   Also add the application server's IPv6 address.

   For example:

   ```
   host all all fe80::cdfb:b6ed:9b38:cf17/128 md5
   ```

   There are authentication methods other than `md5`. For details, refer to the PostgreSQL documentation.

2. Save the file.

Restart your PostgreSQL server for the changes to take effect.

2.1.6 Mail Server

To inform users about relevant issues, the OpenStack integration software requires a mail server in its environment. You can use any mail server that supports SMTP.

The settings for addressing the mail server are defined in the `glassfish.properties` file of the OpenStack integration package.

2.2 Installation

The installation of the OpenStack integration software consists of the following main steps:

1. *Preparing the Software and Setup Utilities* on page 12
2. *Configuring the OpenStack Integration* on page 13
3. *Setting up the Database* on page 15
4. *Setting up the Application Server Resources* on page 16

2.2.1 Preparing the Software and Setup Utilities

The OpenStack integration software and setup utilities are provided in the OpenStack integration package, `oscm-openstack-install-pack.zip`. The contents of the package need to be made available in your environment as follows:

Extract the contents of the `oscm-openstack-install-pack.zip` package to a separate temporary directory on the system from where you want to install and deploy the OpenStack integration software.

In the following sections, this directory is referred to as `<install_pack_dir>`.

After extraction, the following directories are available:

- **databases/app_db**

  Configuration files for setting up the database used by the OpenStack integration software.

- **doc**

  The *OpenStack Integration* guide (this manual).

- **domains/app_domain**
  - Configuration file (`glassfish.properties`) for setting up the application server resources for the domain to which the OpenStack integration software is to be deployed.
  - `applications`: The APP application (`oscm-app.ear`) and the OpenStack service controller (`oscm-app-openstack.ear`).

- **install**

  XML files that support you in setting up the database and the application server resources for APP.

- **samples**:

  Templates and corresponding technical service samples for OpenStack and FUJITSU Cloud Service K5.

2.2.2 Configuring the OpenStack Integration

The OpenStack integration software and setup utilities require a number of settings. These settings are provided in the following subdirectories and files of `<install_pack_dir>`:

- **databases/app_db**
  - `db.properties`: Settings for the database setup and access.
  - `configsettings.properties`: Configuration settings for APP.

    The initial installation stores these settings in the `bssapp` database, where you can change them later, if required. An update installation only adds new settings to the database, but does not overwrite existing ones. In the case that mandatory settings are missing, an error is thrown, and you need to add these settings manually before executing the installation scripts again.

  - `configsettings_controller.properties`: Configuration settings for the OpenStack service controller.

    The initial installation stores these settings in the `bssapp` database. You can change them later using a graphical user interface.

    The `configsettings_controller.properties` file specifies the organization ID and user credentials for accessing OSCM as well as the required credentials for accessing the tenant for your organization in OpenStack. For security reasons, it is recommended that you delete the file as soon as you have successfully installed and configured the OpenStack integration software.

- **domains/app_domain**

  The configuration settings for setting up the application server domain to which APP is deployed are provided in the following file:

  `glassfish.properties`

Additional configuration files contained in other subdirectories are used internally and must not be changed.

For details on the configuration settings, refer to *Configuration Settings* on page 33. For details on updating the configuration settings, refer to *Updating Configuration Settings* on page 29.

You need to adapt the settings in the files above to your environment. In particular, server names, ports, paths, user IDs, and passwords require adaptation.

Proceed as follows to view and adjust the configuration settings:

1. Open each of the configuration files listed above with an editor.
2. Check the settings in each file and adapt them to your environment.
3. Save the files to their original location in `<install_pack_dir>/<subdirectory>`. For future reference, it is a good idea to create a backup of the files.

Observe the following configuration issues:

- The specified ports are suggestions and work with the default settings used in the files.
- If you install everything on the local system, use either the server name or `localhost` in all configuration files for all URLs that need to be resolved by APP.

  Do not mix the specification of server names and `localhost`.

  The `APP_BASE_URL` setting in the `configsettings.properties` file for the `app-domain` domain must be resolved by clients. They always require that the server name be specified.

  Specify the `APP_BASE_URL` setting as follows:

  ```
  APP_BASE_URL=http://<host>:<port>/oscm-app
  ```

  If you have changed the `glassfish.domain.portbase` setting in the `glassfish.properties` file, you must change the port here accordingly.

Configuration for SAML_SP authentication mode:

If the OSCM installation you want to work with is configured for SAML_SP authentication mode, Web service calls to it are secured and authenticated by a Security Token Service (STS). This is a Web service that issues security tokens as defined in the WS-Security/WS-Trust specification. The STS is usually provided by the Identity Provider (IdP) system in use (for example Active Directory Federation Service, Cloudminder, or OpenAM).

To use an STS for Web service calls, you must perform the following steps before installing the OpenStack integration software:

1. From the IdP or OSCM operator, obtain a metadata exchange file in WSDL format generated with and for the IdP system in use. The metadata includes namespace information required for connecting to the STS.
2. Save the metadata exchange file to the following file, overwriting the existing empty file:

   `<install_pack_dir>/domains/app_domain/wsit/STSService.xml`
3. Open the `STSService.xml` file and retrieve the value of `targetNamespace`, for example:

   `http://schemas.microsoft.com/ws/2008/06/identity/securitytokenservice`
4. Open the following file:

   `<install_pack_dir>/domains/app_domain/wsit/wsit-client.xml`
5. Replace the placeholder in the `namespace` tag of the `wsit-client.xml` file with the `targetNamespace` value copied from the `STSService.xml` file.
6. Close and save the `wsit-client.xml` file to its original location.

During the installation process, an `OSCM-wsit.jar` file is created containing the `STSService.xml` file as well as the `wsit-client.xml` file; the `.jar` file is then copied to `<install_pack_dir>/domains/app_domain/lib`.

7. Make sure that you enter correct values for the SAML_SP authentication mode in the `configsettings.properties` file in `<install_pack_dir>/databases/app_db`:
   - `BSS_AUTH_MODE=SAML_SP`
   - `BSS_STS_WEBSERVICE_URL=https://<server>:<port>/{SERVICE}/STS`
   - `BSS_STS_WEBSERVICE_WSDL_URL= https://<server>:<port>/oscm/v1.9/{SERVICE}/STS?wsdl`
   - `APP_KEYSTORE_PASSWORD=changeit`
   - `APP_TRUSTSTORE_PASSWORD=changeit`

2.2.3 Setting up the Database

The OpenStack integration software requires and stores its data in the `bssapp` PostgreSQL database.

The database is created by executing installation scripts. It is initialized with the appropriate schema and settings.

Proceed as follows:

1. Make sure that the database server is running.
2. Open the command prompt (Windows) or a terminal session (UNIX/Linux).
3. Set the following environment variable for your current session:

   `DB_INTERPRETER`: The absolute path and name of the `psql` executable of PostgreSQL. The executable is usually located in the `bin` subdirectory of the PostgreSQL installation directory.

   Example (Unix/Linux):

   ```
   export DB_INTERPRETER="/opt/PostgreSQL/9.1/bin/psql"
   ```

   Example (Windows):

   ```
   set DB_INTERPRETER="C:\Program Files\PostgreSQL\9.1\bin\psql"
   ```

4. Create the database by executing the `build-db.xml` file in `<install_pack_dir>/install` as follows:

   ```
   <ANT_HOME>/bin/ant -f build-db.xml initDB
   ```

   If you set an ID or password other than `postgres` for the PostgreSQL user account (`postgres`) when installing the database management system, you have to specify the ID or password with the call to the `build-db.xml` file as follows:

   ```
   <ANT_HOME>/bin/ant -f build-db.xml initDB
      -Ddb.admin.user=<user ID> -Ddb.admin.pwd=<password>
   ```

Note: It may be required to enclose the `-Ddb.admin.user=<user ID>` and `-Ddb.admin.pwd=<password>` in double or single quotes depending on the operating system.

If the setup of the database fails with errors, proceed as follows:

1. Check and correct the configuration files.
2. Execute the `build-db.xml` file in `<install_pack_dir>/install` as follows:

```
<ANT_HOME>/bin/ant -f build-db.xml DROP.dbsAndUsers
```

3. Repeat the setup.

2.2.4 Setting up the Application Server Resources

The OpenStack integration software requires specific settings and resources in the application server, such as mail settings or a data source.

Proceed as follows to create the resources and make the required settings in the application server:

1. Open the command prompt (Windows) or a terminal session (UNIX/Linux).
2. Execute the `build-glassfish.xml` file in `<install_pack_dir>/install` as follows:

```
<ANT_HOME>/bin/ant -f build-glassfish.xml SETUP
```

   This has the following results:

   - The `app-domain` domain is created and started.
   - The settings and resources for APP are created in the application server.
   - APP (`oscm-app.ear`) is deployed to the `app-domain` domain.
   - A key file for encryption and decryption is generated in the location you specified in the `APP_KEY_PATH` setting in the `configsettings.properties` file. By default, this file is named `key` and located in `<GLASSFISH_HOME>/glassfish/domains/app-domain/config`.
   - The OpenStack service controller (`oscm-app-openstack.ear`) is deployed to the `app-domain` domain.

3. Depending on your environment, it may be required to define a proxy server for the `app-domain` domain in the **JVM Options** of the application server. The OpenStack service controller can address the corresponding system via the proxy server.

   To define a proxy server, specify the following JVM options:

   - `-Dhttps.proxyHost`
   - `-Dhttps.proxyPort`

   If authentication is required, specify the following additional settings:

   - `-Dhttps.proxyUser`
   - `-Dhttps.proxyPassword`

   For all direct communication, you need to bypass the proxy server. Specify the hosts which are to be addressed directly and not through the proxy server in the following setting:

   - `-Dhttp.nonProxyHosts`

   For example, APP must not use the configured proxy for Web service calls to OSCM:

   `-Dhttp.nonProxyHosts=localhost|127.0.0.1|myServer*`

   where `myServer` is the host on which OSCM is running.

In case several service controllers are to run in the same APP domain, and only one of them is to communicate via a proxy server, you need to exclude the target systems of the other service controllers, for example, as follows:

```
-Dhttps.proxyHost=proxy.intern.myserver.com
-Dhttps.proxyPort=8081
-Dhttp.nonProxyHosts=myServer.com|localhost|127.0.0.1|
http://10.140.18.112*|http://myServer.com:8880/templates/*|
https://ror-demo.myServer.com:8014/cfmgapi/endpoint*
```

After having configured the proxy server, restart the `app-domain`.

If the setup of the application server domain fails with errors, proceed as follows:

1. Stop the `app-domain` domain.
2. Delete the `app-domain` domain.
3. Repeat the setup.

2.2.5 Exchanging Certificates

For secure communication of the OpenStack integration software with OSCM, you need to exchange the corresponding certificates.

You need to:

- Import the certificate of OSCM into the truststore of the `app-domain` application server domain of the OpenStack integration software.
- Export the certificate of the `app-domain` domain and import it into the `bes-domain` application server domain of OSCM.

Proceed as follows:

1. Obtain a `.crt` file with the certificate of OSCM from the OSCM operator.

   The `.crt` file can be created, for example, by executing the following command at the command prompt (Windows) or in a terminal session (UNIX/Linux) on the application server where OSCM is deployed:

   ```
   <AppServerJRE>/bin/keytool -export -rfc -alias s1as
      -file ctmgbss.crt -storepass <password> -keystore
      <GLASSFISH_HOME>/glassfish/domains/bes-domain/config/keystore.jks
   ```

2. Import the certificate of OSCM into the truststore of the `app-domain` application server domain.

   To import the OSCM certificate from the `.crt` file you created, you can use, for example, the following command at the command prompt (Windows) or in a terminal session (UNIX/Linux) on the application server:

   ```
   <AppServerJRE>/bin/keytool -import -trustcacerts -alias <alias>
      -file <filename>.crt -storepass <password> -keystore
      <GLASSFISH_HOME>/glassfish/domains/app-domain/config/cacerts.jks
   ```

3. Create a `.crt` file with the certificate of the `app-domain` domain in which you have deployed the OpenStack integration software.

   The `.crt` file can be created, for example, by executing the following command at the command prompt (Windows) or in a terminal session (UNIX/Linux) on the application server:

   ```
   <AppServerJRE>/bin/keytool -export -rfc -alias s1as
   ```

```
   -file ctmgapp.crt -storepass <password> -keystore
   <GLASSFISH_HOME>/glassfish/domains/app-domain/config/keystore.jks
```

4. Import the certificate of the `app-domain` domain into the `bes-domain` application server domain of OSCM.

   To do this, you can use, for example, the following command at the command prompt (Windows) or in a terminal session (UNIX/Linux) on the application server:

```
<AppServerJRE>/bin/keytool -import -trustcacerts -alias ctmgapp
   -file ctmgapp.crt -storepass <password> -keystore
   <GLASSFISH_HOME>/glassfish/domains/bes-domain/config/cacerts.jks
```

5. If the OpenStack integration software and OSCM are configured for SAML_SP authentication mode, obtain the relevant certificates from the IdP system and import them into the truststore of the `app-domain` domain.

   For example, when using Microsoft Active Directory as the IdP, you need to obtain and import the service communications and token-signing certificates.

6. Stop and restart the `app-domain` and the `bes-domain` domains for the certificates to become effective.

2.3 Installing the OpenStack Service Controller in an Existing APP Environment

If you already have a working APP installation in your environment, you can use it for the OpenStack integration and simply register the OpenStack service controller in it. Proceed as follows:

1. Check the prerequisites described in *OSCM and OpenStack* on page 10.
2. Deploy the OpenStack service controller to the `app-domain` domain. To do this, you use the GlassFish administration console, for example:

   `http://127.0.0.1:8848/`

   The `oscm-app-openstack.ear` file of the service controller is located in `<install_pack_dir>/domains/app_domain/applications`

3. Register the OpenStack service controller as follows in APP:
   1. In a Web browser, access the base URL of APP, for example: `http://127.0.0.1:8880/oscm-app`
   2. Log in with the ID and password of the user and organization defined in the `configsettings.properties` file of APP (`BSS_USER_ID` and `BSS_USER_PWD`).
   3. Specify the controller ID (`ess.openstack`) and the technology provider organization responsible for the OpenStack service controller.
   4. Click **Save Configuration** to save the settings.
4. Configure the OpenStack service controller following the instructions in *Updating Configuration Settings* on page 29.

2.4 Update Installation

Before updating your installation of the OpenStack integration software, read the *Release Notes* of the new release. They contain information on compatibility issues, changes and enhancements, and known restrictions.

The platform operator and technology managers must make sure that the following rules are observed: The OSCM version must be higher or equal to the APP version. The APP version must be equal to the controller version.

Example: If you want to use the OpenStack service controller included in the V17.0 release, you must upgrade OSCM and APP to V17.0 first.

Preparing the Update

Before you start with the update installation, carry out the following steps:

1. Make sure that all provisioning operations are complete. Follow the steps as described in *Handling Problems in the Provisioning Process* on page 26.
2. Set the following environment variable for your current session:

   `DB_INTERPRETER`: The absolute path and name of the `psql` executable of PostgreSQL. The executable is usually located in the `bin` subdirectory of the PostgreSQL installation directory.

   Example:

   ```
   export DB_INTERPRETER="/opt/PostgreSQL/9.1/bin/psql"
   ```

3. Create a backup of the key file required for the encryption and decryption of service parameters with data type `PWD` or custom attributes marked for encryption. By default, the file is named `key` and located in the following directory:

   `<GLASSFISH_HOME>/glassfish/domains/app-domain/config`

Note: For the update installation, make sure that the default path is specified in the `configsettings.properties` file (`APP_KEY_PATH="./key"`) and that the file is named `key`. The installation script expects this default and does not look up the database.

Updating the Database

Proceed with updating the database as follows:

1. Check whether the file

   `postgresql-9.1-903.jdbc4.jar`

   is contained in the following directories of the application server:

   - `lib` directory of the `app-domain` domain
   - `<GLASSFISH_HOME>/mq/lib/ext`

   If it is not, copy the file from the `<install_pack_dir>/install/lib` directory to the location where it is missing.
2. Create a backup of the `bssapp` database using the standard PostgreSQL commands. The database backup must be compatible with PostgreSQL 9.1.12.
3. Update the following configuration files so that the settings match your current installation:
   - `db.properties`
   - `configsettings.properties`

- `configsettings_controller.properties`

Particularly take care of settings which have been introduced or changed with the new release to which you are upgrading.

4. Update the schema and configuration settings of the `bssapp` database by executing the `build-db.xml` file in `<install_pack_dir>/install` as follows:

```
<ANT_HOME>/bin/ant -f build-db.xml updateDatabase
```

Note: Make sure that Ant runs in a Java 8 runtime environment when calling the `build-db.xml` file.

Updating the Application Server

After you have executed the preparation steps:

1. Copy the key file of which you have created a backup copy into the following directory of the extracted installation package:

   `oscm-app-install-pack\domains\app_domain\installer`

   Make sure that the key file is named `key`. The installation script will then copy the key file to is default location in the updated version of the application server:

   `<GLASSFISH_HOME>/glassfish/domains/app-domain/config`

Note: For the update installation, make sure that the default path is specified in the `configsettings.properties` file (`APP_KEY_PATH="./key"`) and that the file is named `key`. The installation script expects this default and does not look up the database.

2. Redeploy or deploy the applications that make up the OpenStack integration software in the `app-domain` domain:
   a. `oscm-app`
   b. `oscm-app-openstack`
3. Exchange the following certificates again:
   - Import the certificates of OSCM and OpenStack into the truststore of the `app-domain` application server domain of the OpenStack integration software.
   - Export the certificate of the `app-domain` and import it into the `bes-domain` application server domain of OSCM.
4. Restart the `app-domain` domain.

Updating the Configuration for SAML_SP Mode

If you are running OSCM and the OpenStack integration software in SAML_SP mode, and if the IdP metadata of your SSO environment have changed, you need to update your WSIT files accordingly:

1. Extract the `OSCM-wsit.jar` file into a separate directory.

   The `.jar` file is located in

   `<GLASSFISH_HOME>/domains/app_domain/lib`

   The `OSCM-wsit.jar` file contains the following files:

   `wsit-client.xml`

`STSService.xml`

2. Adapt the `.xml` files as required by your environment. For details, refer to *Configuring the OpenStack Integration* on page 13.
3. Recreate the `OSCM-wsit.jar` file with the modified contents.
4. Stop the `app-domain` domain.
5. Copy the adapted `OSCM-wsit.jar` to the following directory:
   `<GLASSFISH_HOME>/domains/app_domain/lib`
6. Restart the `app-domain` domain.

Note: If, for some reason, you recreate the `app-domain` domain, you also need to recreate the `OSCM-wsit.jar` in the `<GLASSFISH_HOME>/domains/app_domain/lib` directory.

3 Creating and Publishing Services

The following sections describe how to create and publish services in OSCM by means of which customers can request and use virtual systems in OpenStack.

3.1 Prerequisites and Preparation

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

- To create technical services for the OpenStack integration in OSCM, you must have access to OSCM as a technology manager. You must be a member of the technology provider organization responsible for the OpenStack service controller as specified in the configuration settings for the installation.
- Templates for the virtual systems to be provisioned must exist. They form the basis for the technical services in OSCM. The user specified in the configuration settings for the installation must have the necessary credentials to create and configure virtual systems for your organization based on these templates in OpenStack.

  In FUJITSU Cloud Service K5, the `cpf_systemowner` role is required.
- To create marketable services for the OpenStack integration in OSCM, you must have access to OSCM 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 OSCM in your service manager role.

3.2 Creating Technical Services

The first step in providing OSCM services for OpenStack is to create one or more technical services.

Proceed as follows:

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

   The OpenStack integration package, `oscm-openstack-install-pack.zip`, includes technical services as samples:

   - `samples/TechnicalService_OpenStack.xml` for native OpenStack
   - `samples/TechnicalService_K5.xml` for FUJITSU Cloud Service K5

   The technical services specify sample templates as parameter options. The sample templates are also included in the `oscm-openstack-install-pack.zip` file:

   - `samples/template.json` for a template of type AWS CloudFormation for native OpenStack
   - `samples/template.yaml` for a template of type Heat Orchestration Template for native OpenStack
   - `samples/template_k5.yaml` for a template of type Heat Orchestration Template for FUJITSU Cloud Service K5

   You can use the samples to see how templates are mapped to technical service definitions. Copy the sample files to a location of your choice that can be reached from OSCM by HTTP or HTTPS. For example, you could copy them to the `docroot/templates` directory of the `app-domain` domain in your application server.

   In the technical service definition, be sure to specify:

   - The asynchronous provisioning type

- The direct access type
- Service parameters which correspond to the parameters specified in the template defined in OpenStack. For details on the supported service parameters, refer to *Service Parameters and Operations* on page 41.

Note: Make sure that you do not specify the `baseUrl` attribute in the technical service definition XML file. It specifies an application's remote interface and is not needed for providing OSCM services for OpenStack.

2. Log in to the OSCM 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 OSCM.

3.3 Creating and Publishing Marketable Services

As soon as the technical services for the OpenStack integration exist in OSCM, 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 systems.

Proceed as follows:

1. Log in to the OSCM administration portal with your service manager account.
2. Define one or more marketable services based on the technical services you created for OpenStack.
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 OSCM.

4 Using OpenStack Services in OSCM

The following sections describe how users can subscribe to and work with the services you have created for OpenStack in OSCM. You will find details of the supported usage scenarios outlined in *Usage Scenarios* on page 9.

4.1 Subscribing to Services

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

To enable the provisioning of an OpenStack instance, the customer has to enter the name of the key pair of the virtual system when subscribing to the corresponding service in OSCM. The key pair name and the associated private key are used to securely access the OpenStack instance. For details on creating key pairs, refer to the user documentation of OpenStack.

When subscribing to a service, the customer also has to enter a name for the stack to be instantiated. Before the provisioning operation is started, the name is checked against the OpenStack conventions or a pattern specified in the technical service definition. OpenStack generates a random number that is appended to this stack name to make it unique.

Depending on the service parameters, the technical service either maps to one specific template or the customer can choose from different templates that specify the resource orchestration in OpenStack.

Depending on the parameters defined for the technical service, the customer can additionally choose from different options to configure the resources for the instance to be provisioned, for example, the instance type, the number of CPUs, or the storage size.

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 system provided by OpenStack using the access information indicated in the subscription details on the marketplace in OSCM.

In OpenStack, the virtual systems created for OSCM subscriptions are managed in the same way as other instances. They can be viewed and monitored with the standard OpenStack tools. Changes, however, should not be made as this may cause problems and inconsistencies between OpenStack and OSCM.

4.2 Executing Service Operations

Customers can explicitly start and stop servers and suspend and resume instances in OpenStack from OSCM. To do this, they execute the appropriate service operation from the subscription for the virtual system:

- **Start**: Starts all servers in the OpenStack instance that were stopped.
- **Stop**: Stops all servers in the OpenStack instance that were started.
- **Suspend**: Suspends the OpenStack instance.
- **Resume**: Resumes the OpenStack instance if it was suspended.

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

Customers can view the status of all operations they started in their account information on the OSCM marketplace.

4.3 Terminating Subscriptions

A customer can at any time terminate a subscription for a virtual system in OpenStack.

OpenStack is triggered to delete the virtual system. The subscription is terminated in OSCM independent of whether the deletion is successful in OpenStack.

5 Administrating the OpenStack Integration

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

- *Controlling the Provisioning Process* on page 26
- *Handling Problems in the Provisioning Process* on page 26
- *Handling Communication Problems Between APP and OSCM* on page 27
- *Backup and Recovery* on page 28
- *Updating Configuration Settings* on page 29
- *Adapting the Log Configuration* on page 30

5.1 Controlling the Provisioning Process

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

In the definition of the technical services for OpenStack, 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 OpenStack 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.

5.2 Handling Problems in the Provisioning Process

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

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

After solving the problem, the OpenStack integration components and OSCM 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 OpenStack service controller.
2. Invoke the instance status interface of APP for the OpenStack service controller by opening the following URL in a Web browser:

   `https://<server>:<port>/oscm-app/controller/?cid=ess.openstack`

   For example:

   `https://127.0.0.1:8881/oscm-app/controller/?cid=ess.openstack`

   The Web page shows all subscriptions for OpenStack, including detailed information such as the customer organization, the ID of the related OpenStack 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 OpenStack 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 OpenStack does not respond.
   - `UNLOCK` - to remove the lock for an OpenStack instance in APP.
   - `DELETE` - to terminate the subscription in OSCM and remove the instance in APP, but keep the virtual system in OpenStack for later use. The service manager role is required for this action.
   - `DEPROVISION` - to terminate the subscription in OSCM, remove the instance in APP, and delete the virtual system in OpenStack. The service manager role is required for this action.
   - `ABORT_PENDING` - to abort a pending provisioning operation in OSCM. OSCM is notified to roll back the changes made for the subscription and return it to its previous state. In OpenStack, no actions are carried out.
   - `COMPLETE_PENDING` - to complete a pending provisioning operation in OSCM. OSCM 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.
- The **Run with timer** column indicates whether the timer for the interval at which APP polls the status of instances is running. You can reset the timer, if required. For details on the timer setting, refer to *Configuration Settings* on page 33.

5.3 Handling Communication Problems Between APP and OSCM

When the communication between APP and OSCM is no longer possible, for example, because OSCM is stopped, APP suspends the processing of requests. An internal flag is set in the APP database: `APP_SUSPEND=true`, and an email is sent to the address specified in the `APP_ADMIN_MAIL_ADDRESS` configuration setting.

Contact the OSCM operator to make sure that OSCM is up and running again correctly.

You then have the following possibilities to resume the processing of requests by APP:

1. Click the link provided in the email.
2. Log in to APP.

   APP is restarted instantly. In the APP database, the `APP_SUSPEND` key is set to `false`.

As an alternative, you can proceed as follows:

1. In a Web browser, access the base URL of APP, for example:

`http://127.0.0.1:8880/oscm-app`

2. Log in with the ID and password of the user and organization defined in the `configsettings.properties` file of APP (`BSS_USER_ID` and `BSS_USER_PWD`).

   A message is shown that APP has been suspended due to a communication problem with OSCM.

3. Click **Restart**.

   APP is restarted instantly. In the APP database, the `APP_SUSPEND` key is set to `false`.

5.4 Backup and Recovery

The OpenStack integration software does not offer integrated backup and recovery mechanisms. Use the standard file system, application server, and database mechanisms instead.

Backup

It is recommended that you create a regular backup of the following data according to the general guidelines of your organization:

- Database (`bssapp`). The frequency of database backups depends on the amount of changes and the availability of time slots with low load. PostgreSQL supports database backups without previous shutdown. For details, refer to the PostgreSQL documentation.

  Make sure to also make a backup of the file containing the key required for encryption and decryption of service parameters with data type `PWD` or custom attributes marked for encryption. By default, this file is located in:

  `<GLASSFISH_HOME>/glassfish/domains/app-domain/config/key`

  The location of this file can be changed using the `APP_KEY_PATH` configuration setting.

- Certificates contained in the truststore of the `app-domain` domain (`cacerts.jks` file).
- Configuration files.

Note:	When preparing for an update installation of the current OpenStack integration software, always create a backup of the data mentioned above.

Recovery

If you need to recover your OpenStack integration installation, the recommended procedure is as follows:

1. Restore the `bssapp` database from the backup using the relevant PostgreSQL commands.
2. Make sure that the file containing the key required for encryption and decryption of service parameters with data type `PWD` or custom attributes marked for encryption exists in the location specified in the `APP_KEY_PATH` configuration setting. By default, this file is to be located in:

   `<GLASSFISH_HOME>/glassfish/domains/app-domain/config/key`

   Its default name is `key`.

   If the file is missing, copy it from your backup to the correct location.

3. Stop the `app-domain` domain of the application server.
4. Restore the certificate truststore of the `app-domain` domain (`cacerts.jks` file) from the backup.
5. Start the `app-domain` domain.

5.5 Updating Configuration Settings

The OpenStack integration software and setup utilities require a number of settings. In the installation, you adapted the settings, in particular server names, ports, paths, and user IDs, to your environment.

The configuration settings are provided in the following subdirectories and files of `<install_pack_dir>`:

- **`databases/app_db`**
  - `db.properties`: Settings for the database setup and access.
  - `configsettings.properties`: Configuration settings for APP.

    The initial installation stores these settings in the `bssapp` database, where you can change them later, if required. An update installation overwrites the settings. If you don't want existing settings to be overwritten, delete them from the properties file. In case that mandatory settings are missing in the properties file and not yet stored in the database, an exception will occur.
  - `configsettings_controller.properties`: Configuration settings for the OpenStack service controller.

    The initial installation stores these settings in the `bssapp` database. You can change them later using a graphical user interface.
- **`domains/app_domain`**

  The configuration settings for setting up the application server domain to which APP is deployed are provided in the following file:

  `glassfish.properties`

For details on the configuration settings, refer to *Configuration Settings* on page 33.

If you need to change the settings, proceed as described in the following sections.

To update the configuration settings for database access:

1. Log in to the administration console of the application server.
2. Adapt the settings as required.

To update the configuration settings for the application server:

1. Open the `glassfish.properties` file located in `<install_pack_dir>/domains/app_domain` with an editor.
2. Check the settings in the file and adapt them to your environment if required.
3. Save the file to its original location in `<install_pack_dir>/domains/app_domain`.
4. Update the settings and resources in the application server by executing the `build-glassfish.xml` file in `<install_pack_dir>/install` as follows:

```
<ANT_HOME>/bin/ant -f build-glassfish.xml
   SETUP.configureDomains
```

To update the configuration settings for APP:

1. Edit the content of the `configsettings.properties` file as required.
2. Execute the `build-db.xml` file in `<install_pack_dir>/install` as follows:

```
<ANT_HOME>/bin/ant -f build-db.xml
```

```
UPDATE.configSettings
```

To update the configuration settings for the OpenStack service controller:

1. In a Web browser, access the URL of the OpenStack service controller, for example: `http://127.0.0.1:8880/oscm-app-openstack`.
2. Log in with the ID and password of the user specified in the configuration settings for the OpenStack service controller (`BSS_USER_ID` and `BSS_USER_PWD`) or as another technology manager registered for the same organization.
3. Enter the required settings.
4. Save the settings.

If you want to change the technology provider organization responsible for the OpenStack service controller, you can use the Web interface of APP:

1. In a Web browser, access the base URL of APP, for example: `http://127.0.0.1:8880/oscm-app`
2. Log in with the ID and password of the user specified for `BSS_USER_KEY` in the configuration settings for APP or as another administrator of the same organization.
3. Specify the technology provider organization for the OpenStack service controller, `ess.openstack`.
4. Save the settings.
5. Make sure that the configuration settings for the OpenStack service controller are updated.

   Any technology manager registered for the technology provider organization you specified can log in to the graphical user interface for updating the controller configuration settings (see above). At least the ID and password of the user to be used for accessing OSCM must be changed in the controller configuration settings.

5.6 Adapting the Log Configuration

The OpenStack integration software records information and problems such as connection issues in the following log files on the application server:

- `<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/logs/app-openstack.log`: Log of the OpenStack service controller
- `<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/logs/app-core.log`: Log of the APP component

The logging is based on `log4j`. The default log level is `INFO`, which may not be sufficient depending on the circumstances. In such a case, you will need to adapt the log level in the configuration files. The following configuration files are of relevance:

- `<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/config/log4j.ess.openstack.properties`: Log configuration of the OpenStack service controller
- `<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/config/log4j.app.core.properties`: Log configuration of the APP component

Proceed as follows to adapt the log level:

1. Open the relevant configuration file.
2. Find the string `log4j.logger.org.oscm.app` in the configuration file.
3. Change the log level as desired to one of the following:
   - `ERROR` - designates error events that might still allow the application to continue running.

- `WARN` - designates potentially harmful situations.
- `INFO` - designates informational messages that highlight the progress of the application at coarse-grained level.
- `DEBUG` - designates fine-grained informational events that are most useful to debug an application.

Example:

```
log4j.logger.org.oscm.app=INFO
```

Every 60 seconds, the OpenStack integration software checks for changes in the log configuration. There is no need to restart the application.

6 Uninstallation

If you want to uninstall the OpenStack integration software, take the following preparations:

- Back up resources and data you would like to keep. For details, refer to *Backup and Recovery* on page 28.
- In OSCM, delete the marketable services and technical services related to OpenStack.

To uninstall the OpenStack integration software:

1. Stop the `app-domain` domain in the application server.
2. Delete the `app-domain` domain.
3. Delete the `bssapp` database in the database management system.
4. Uninstall the database management system and the application server if you no longer need them for other purposes.

For details on how to proceed, refer to the documentation of the database management system and the application server.

Appendix A: Configuration Settings

The configuration settings for the OpenStack integration software are provided in the following files in subfolders of the directory to which you extracted the `oscm-openstack-install-pack.zip` file (`<install_pack_dir>`):

- `domains/app_domain/glassfish.properties`
- `databases/app_db/db.properties`
- `databases/app_db/configsettings.properties`
- `databases/app_db/configsettings_controller.properties`

This appendix describes the settings in detail.

A.1 GlassFish Configuration Settings

The `glassfish.properties` file located in `<install_pack_dir>/domains/app_domain` contains the configuration settings for the GlassFish application server. The settings are required for configuring the domain where APP is deployed.

Below you find a detailed description of the settings.

GLASSFISH_HOME

The absolute path and name of the GlassFish installation directory.

JDBC_DRIVER_JAR_NAME

The name of the PostgreSQL JDBC driver jar file as available after installation.

Example: `postgresql-9.1-903.jdbc4.jar`

MAIL_HOST

The host name or IP address of your mail server.

MAIL_RESPONSE_ADDRESS

The email address used by the server as the sender of emails.

Example: `saas@yourcompany.com`

MAIL_PORT

The port of your mail server.

Default: `25`

MAIL_USE_AUTHENTICATION

Optional. Defines whether mails can be sent only to users authenticated against the SMTP mail system.

Allowed values: `true`, `false`

Default: `false`

MAIL_USER

Mandatory if `MAIL_USE_AUTHENTICATION=true`. Specifies the name of the user to be used for authentication against the SMTP mail system.

MAIL_PWD

Mandatory if `MAIL_USE_AUTHENTICATION=true`. Specifies the password of the user to be used for authentication against the SMTP mail system.

MAIL_TIMEOUT

Optional. The time interval in milliseconds for sending email messages, i.e. until a socket I/O timeout occurs.

Allowed values: Any value between `0` and `4924967296`

Default: `30000`

MAIL_CONNECTIONTIMEOUT

Optional. The time interval in milliseconds for establishing the SMTP connection, i.e. until a socket connection timeout occurs.

Allowed values: Any value between `0` and `4924967296`

Default: `30000`

glassfish.domain.portbase

Mandatory. The base number for all ports used by the domain of the APP application.

Example: `8800`

glassfish.domain.portadmin

The administration port of the domain used for APP.

Example: `8848`

glassfish.domain.name

The name of the domain where APP is deployed.

Example: `app-domain`

glassfish.domain.admin.user

The user name of the APP domain administrator.

Default: `admin`

glassfish.domain.admin.pwd

The password of the APP domain administrator.

Default: `adminadmin`

glassfish.domain.admin.master.pwd

Mandatory. The master password required for accessing the keystore and truststore files of the application server domain.

Default: `changeit`

glassfish.domain.WS_PORT_SECURE

The port used for a secure HTTP listener for Web service connections of the application server.

In some operational environments it is required to use separate certificates for the SOAP communication between OSCM and APP and for Web browser access, i.e. separate certificates for internal and external communication must be provided.

Default: `8082`

glassfish.domain.stop.waitSeconds

Mandatory. The time in seconds the application server waits until a stop domain operation is executed.

Default: `60`

glassfish.domain.start.maxWaitSeconds

Mandatory. The maximum time in seconds the application server waits until it checks whether a domain is started.

Default: `600`

A.2 Database Configuration Settings

The `db.properties` file located in `<install_pack_dir>/databases/app_db` contains the configuration settings for database access. This configuration is used for the initial setup and schema updates.

db.driver.class

The Java class of the JDBC driver.

Default: `org.postgresql.Driver`

db.host

The database host.

Default: `localhost`

db.port

The database port.

Default: `5432`

db.name

The name of the database.

Default: `bssapp`

db.user

The name of the user to connect to the database.

Default: `bssuser`

db.pwd

The password of the user to connect to the database.

Default: `bssuser`

db.type

The type of the database.

Default: `postgresql`

A.3 APP Configuration Settings

The `configsettings.properties` file located in `<install_pack_dir>/databases/app_db` contains the configuration settings for APP.

APP_BASE_URL

`APP_BASE_URL=http://<server>:<port>/oscm-app`

The URL used to access APP.

APP_TIMER_INTERVAL

`APP_TIMER_INTERVAL=15000`

The interval (in milliseconds) at which APP polls the status of instances. If you increase the value, provisioning takes longer. If you decrease it, more load is put on the system. We strongly recommend that you do not set a value of more than 180000 milliseconds (3 minutes), although the maximum value is much higher (922337203685477580).

If you do not specify this setting at all, the default value used is 15000.

APP_MAIL_RESOURCE

`APP_MAIL_RESOURCE=mail/APPMail`

The JNDI name of the GlassFish mail resource used to send emails.

The resource `mail/APPMail` is created during setup with the parameters defined in the `glassfish.properties` file. This setting needs to be changed only if you want to use a different mail resource.

APP_ADMIN_MAIL_ADDRESS

`APP_ADMIN_MAIL_ADDRESS=admin@example.com`

The email address to which email notifications are sent.

APP_KEY_PATH

The path to the file containing the key required for encryption and decryption of service parameters with data type `PWD` or custom attributes marked for encryption.

Upon the start of APP, it is checked whether a key file exists in the location specified in this setting. If this not the case, the key file is automatically generated and stored in the location specified here. If nothing is specified, the file will be generated into the default location and named `key`.

Default:

`<GLASSFISH_HOME>/glassfish/domains/app-domain/config`

Note:	Be aware that the key file must not be deleted. Otherwise, encryption and decryption is no longer possible. It is recommended to create a backup of this file once generated.

APP_KEYSTORE_PASSWORD

`APP_KEYSTORE_PASSWORD=changeit`

The password required to access the keystore of the domain used for APP in the application server.

APP_TRUSTSTORE

`APP_TRUSTSTORE=./cacerts.jks`

The path and file name of the truststore of the application server domain used for APP. The certificate of OSCM is stored in this truststore.

APP_TRUSTSTORE_BSS_ALIAS

`APP_TRUSTSTORE_PASSWORD=bes-s1as`

The alias of the certificate of OSCM as stored in the truststore of the application server domain used for APP.

APP_TRUSTSTORE_PASSWORD

`APP_TRUSTSTORE_PASSWORD=changeit`

The password required to access the truststore of the application server domain used for APP.

BSS_AUTH_MODE

`BSS_AUTH_MODE=INTERNAL`

Specifies whether OSCM is used for user authentication or whether it acts as a SAML service provider and allows for single sign-on. The setting must be identical to the `AUTH_MODE` setting in OSCM.

Possible values: `INTERNAL` (OSCM user authentication is used) or `SAML_SP` (OSCM acts as a SAML service provider).

Contact the OSCM platform operator for details on which value to set.

BSS_USER_KEY

`BSS_USER_KEY=<userKey>`

The user key for accessing OSCM.

Replace `<userKey>` with the user key which you receive with the confirmation email for your user account.

The user specified here must have the administrator role for your organization in OSCM. The user account is used for carrying out actions on behalf of APP in OSCM. In addition, the user is allowed to register service controllers in APP.

BSS_USER_ID

`BSS_USER_ID=<userId>`

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

Replace `<userId>` with the user ID.

BSS_USER_PWD

`BSS_USER_PWD=_crypt:<password>`

The password of the user specified in `BSS_USER_KEY` for accessing OSCM.

Replace `<password>` with the plain text password. The password is encrypted when it is stored in the database.

BSS_WEBSERVICE_URL

`BSS_WEBSERVICE_URL=https://<server>:<port>/{SERVICE}/BASIC`

Mandatory when `BSS_AUTH_MODE` is set to `INTERNAL` and basic authentication is used. The endpoint of the OSCM Web services to be used. The `{SERVICE}` placeholder must not be replaced.

BSS_WEBSERVICE_WSDL_URL

`BSS_WEBSERVICE_WSDL_URL=https://<server>:<port>/oscm/v1.9/{SERVICE}/BASIC?wsdl`

Mandatory when `BSS_AUTH_MODE` is set to `INTERNAL` and basic authentication is used. The URL specifying the version of the OSCM Web services to be used. The `{SERVICE}` placeholder must not be replaced.

BSS_STS_WEBSERVICE_URL

`BSS_STS_WEBSERVICE_URL=https://<server>:<port>/{SERVICE}/STS`

Mandatory when `BSS_AUTH_MODE` is set to `SAML_SP` and security token based authentication is used. The endpoint of the OSCM Web services to be used. The `{SERVICE}` placeholder must not be replaced.

BSS_STS_WEBSERVICE_WSDL_URL

`BSS_STS_WEBSERVICE_WSDL_URL=https://<server>:<port>/oscm/v1.9/{SERVICE}/STS?wsdl`

Mandatory when `BSS_AUTH_MODE` is set to `SAML_SP`. The URL specifying the version of the OSCM Web services to be used. The `{SERVICE}` placeholder must not be replaced.

A.4 Controller Configuration Settings

The `configsettings_controller.properties` file located in `<install_pack_dir>/databases/app_db` contains the configuration settings for the OpenStack service controller. This configuration is used for the initial setup and stored in the APP database.

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.

CONTROLLER_ID

`CONTROLLER_ID=ess.openstack`

The identifier of the service controller.

BSS_ORGANIZATION_ID

`BSS_ORGANIZATION_ID=<organizationID>`

The ID of the organization in OSCM responsible for the service controller. The organization must have the technology provider role.

BSS_USER_ID

`BSS_USER_ID=<userId>`

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

Replace `<userId>` with the user ID.

BSS_USER_KEY

`BSS_USER_KEY=<userKey>`

The user key for accessing OSCM.

Replace `<userKey>` with the user key which you receive with the confirmation email for your user account.

The user specified here must have the technology manager role in OSCM and belong to the organization specified in the `BSS_ORGANIZATION_ID` setting.

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

BSS_USER_PWD

`BSS_USER_PWD=_crypt:<password>`

The password of the user specified in `BSS_USER_KEY` for accessing OSCM.

Replace `<password>` with the plain text password. The password is encrypted when it is stored in the database.

API_USER_NAME

`API_USER_NAME=<userName>`

The user name to be used to access the tenant for your organization in OpenStack. Once authenticated, this user is authorized to access the Heat API.

The user must have the necessary credentials to create and configure virtual systems for the tenant. In FUJITSU Cloud Service K5, the `cpf_systemowner` role is required.

API_USER_PWD

`API_USER_PWD=_crypt:<password>`

The password of the user for accessing the tenant in OpenStack.

Replace `<password>` with the plain text password which is valid for the user given in `API_USER_NAME`. The password is encrypted when it is stored in the database.

KEYSTONE_API_URL

```
KEYSTONE_API_URL=https://<keystone endpoint>/<version>/auth
```

The URL of the Keystone API for authenticating the user specified in `API_USER_NAME`. Keystone is the identity service used by OpenStack.

Replace `<keystone endpoint>` with the URL leading to the Keystone API. Replace `<version>` with the API version to use, currently `v3`. HTTP and HTTPS are supported.

The URL of the Keystone API can also be specified as a parameter in technical service definitions. The technical service parameter overrides the setting you specify here.

TEMPLATE_BASE_URL

```
TEMPLATE_BASE_URL=https://<app>/docroot/templates/
```

The URL leading to the templates that are mapped to technical service definitions. The file names of the templates to be used are specified when customers subscribe to a corresponding service on a OSCM marketplace.

Replace `<app>` with the URL used to access the `app-domain` domain in your application server, or specify another URL that can be reached from OSCM by HTTP or HTTPS.

The template URL can also be specified as a parameter in technical service definitions. The technical service parameter overrides the setting you specify here.

TENANT_ID

```
TENANT_ID=<tenant>
```

The identifier of the tenant for your organization in OpenStack. The tenant is specified in each request to the Heat API.

The tenant ID can also be specified as a parameter in technical service definitions. The technical service parameter overrides the setting you specify here.

DOMAIN_NAME

```
DOMAIN_NAME=<domain>
```

Optional. The name of the Keystone domain to use. A domain is a container for projects, users, and groups in the OpenStack environment. If not specified, the default Keystone domain is used (name: `default`).

The domain name can also be specified as a parameter in technical service definitions. The technical service parameter overrides the setting you specify here.

READY_TIMEOUT

```
READY_TIMEOUT=300000
```

Optional. The number of milliseconds to wait for the completion of an operation to start or stop the servers in a stack. If the operation is not completed within the given time, it is aborted with a timeout error. The administrators of the customer organization owning the subscription from which the operation was started are informed by email. After the problem has been sorted out, the start or stop operation can be repeated by the customer.

If the setting is not specified or the given value is not a number, it is ignored, and start and stop operations never time out.

Appendix B: Service Parameters and Operations

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

Service Parameters

The OpenStack 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.

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

ACCESS_INFO_PATTERN

Mandatory. The access information to be output in the subscription details on the marketplace as soon as the provisioning is complete. This information must give all the details the customer needs to access a provisioned instance, e.g. an IP address and a key pair name.

The information must correspond to the output parameters specified in the template file. If the values do not match, the subscription is rejected.

Customers should not be able to enter a value for this parameter, i.e. it should not be configurable.

Example: `Key pair name: {KP_Out}; IP: {IP_Out}`

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`

STACK_NAME

Mandatory. The name of the virtual system to be instantiated. This name must be specified by customers when they subscribe to a corresponding service.

The name is restricted to 30 characters. It must start with a letter and only contain the following characters: Letters `A-Z` and `a-z`, numbers `0-9`, hyphen `-`, underscore `_`, period `.`

OpenStack generates a random number that is appended to the name to make it unique.

Example: `MySystem`

STACK_NAME_PATTERN

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

Stack names must comply with both, the pattern specified here and the general naming conventions (see the description of `STACK_NAME`). If no pattern is given, the general conventions apply.

Customers should not be able to enter a value for this parameter, i.e. it should not be configurable.

Example: `(host_[0-9]*){1,30}`

TEMPLATE_BASE_URL

Optional. The URL leading to the template file or files specified in the `TEMPLATE_NAME` parameter. Specify a URL that can be reached from OSCM by HTTP or HTTPS.

If not specified, the URL is obtained from the controller configuration settings.

Customers should not be able to enter a value for this parameter, i.e. it should not be configurable.

Example: `https://myserver:8880/oscm-app/docroot/templates/`

TEMPLATE_NAME

Mandatory. The name or relative path and name of the template file which forms the basis for the OpenStack instance to be instantiated. The `TEMPLATE_NAME` is added to the `TEMPLATE_BASE_URL` specified in the technical service definition or in the controller configuration settings.

The template file details everything needed to carry out the resource orchestration in OpenStack. The template must be specified by customers when they subscribe to a corresponding service.

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 template, have fixed parameter options for selection.

Example: `MyTemplate.json`

TENANT_ID

Optional. The identifier of the tenant for your organization in OpenStack. The tenant is specified in each request to the Heat API.

If not specified, the tenant ID is obtained from the controller configuration settings.

Customers should not be able to enter a value for this parameter, i.e. it should not be configurable.

Example: `6f4c1e4cbfef4d5a8a1345882fbca110`

KEYSTONE_API_URL

Optional. The URL of the Keystone API for authenticating the user specified to access the tenant for your organization in OpenStack. Keystone is the identity service used by OpenStack.

Specify the URL according to the following format:

`http[s]://<keystone endpoint>/<version>/auth`

`<keystone endpoint>` is the URL leading to the Keystone API, `<version>` the API version to use, currently `v3`. HTTP and HTTPS are supported.

If not specified, the URL is obtained from the controller configuration settings.

Customers should not be able to enter a value for this parameter, i.e. it should not be configurable.

Example: `https://my.keystone.com:5000/v3/auth`

DOMAIN_NAME

Optional. The name of the Keystone domain to use. A domain is a container for projects, users, and groups in the OpenStack environment.

If not specified, the domain is obtained from the controller configuration settings. If it is not specified there either, the default Keystone domain is used (name: `default`).

Customers should not be able to enter a value for this parameter, i.e. it should not be configurable.

Example: `mydomain`

TP_ImageId

Mandatory. The virtual machine image for the instance to be instantiated. Any valid OpenStack image can be specified.

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 template, have fixed parameter options for selection.

Example: `cedarish` for a cedar image.

TP_InstanceType (OpenStack) / **TP_flavor** (FUJITSU Cloud Service K5)

Mandatory. The flavor for the instance to be instantiated. The flavor defines the compute, memory, and storage capacity. Any valid OpenStack flavor can be specified.

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 flavor, have fixed parameter options for selection.

Example: `m1.small`

TP_KeyName

Mandatory. The key pair name of the instance to be instantiated.

The key pair name must be specified by the customer when subscribing to an OpenStack 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 OpenStack.

Example: `my-key-pair`

TP_*

Optional. Any number of parameters that are mapped from the parameters defined in the template file. The parameters in the template file detail everything needed to carry out the resource orchestration in OpenStack. For each parameter in the template file, there must be a corresponding parameter in the technical service definition.

The parameter names must correspond to the names in the template file. The string `TP_` must be prepended to the name. If the names do not match this pattern, the subscription is rejected.

Service Operations for OpenStack Instances

The OpenStack service controller supports the service operations below for the OpenStack instances.

The `actionURL` for each operation is:

`https://<host>:<port>/OperationService/AsynchronousOperationProxy?wsdl`

`<host>` and `<port>` are the server and port of the `app-domain` domain where the OpenStack service controller is deployed.

SUSPEND_VIRTUAL_SYSTEM

Suspends the OpenStack instance.

RESUME_VIRTUAL_SYSTEM

Resumes the OpenStack instance if it was suspended.

START_VIRTUAL_SYSTEM

Starts all servers in the OpenStack instance that were stopped. For this operation, a timeout value can be configured in the controller configuration settings.

STOP_VIRTUAL_SYSTEM

Stops all servers in the OpenStack instance that were started. For this operation, a timeout value can be configured in the controller configuration settings.

Note: If you provision virtual systems that do not support some of these operations or do not contain servers that can be started and stopped, make sure that you remove the corresponding service operations from the technical service definition.

FUJITSU Cloud Service K5 does not support the `SUSPEND_VIRTUAL_SYSTEM` and `RESUME_VIRTUAL_SYSTEM` operations.

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. OSCM comes with a native billing system, but can also be integrated with external ones.

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 OSCM 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 OSCM, 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 OSCM to provide their services to customers.

Marketplace Owner

An organization which holds a marketplace in OSCM, 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 OSCM.

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 Service Provider (PSP)

A company that offers suppliers or resellers online services for accepting electronic payments by a variety of payment methods including credit card or bank-based payments such as direct debit or bank transfer. Suppliers and resellers can use the services of a PSP for the creation of invoices and payment collection.

Payment Type

A specification of how a customer may pay for the usage of his subscriptions. The operator defines the payment types available in OSCM; 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. OSCM 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 OSCM. 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 OSCM for offering applications provisioned by technology providers to customers.

Technical Service

The representation of an application in OSCM. 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 OSCM.