V17.7.0


# QuickStart Guide

June 2018

| | |
|---|---|
| Trademarks | LINUX is a registered trademark of Linus Torvalds.<br>Open Service Catalog Manager is a registered trademark of FUJITSU LIMITED.<br>The OpenStack Word Mark and OpenStack logo are registered trademarks/service marks or trademarks/ service marks of the OpenStack Foundation in the United States and other countries.<br>Apache Tomcat, Tomcat, and Apache are trademarks of The Apache Software Foundation.<br>Java is a registered trademark of Oracle and/or its affiliates.<br>UNIX is a registered trademark of the Open Group in the United States and in other countries.<br>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

This manual describes how to get started with Open Service Catalog Manager (OSCM) .

## Readers of this Manual

This manual is directed to operators who want to quickly set up a basic installation of OSCM with Docker and Docker Compose. For more detailed information on configuration and usage, refer to the official *OSCM documentation*. It assumes that you are familiar with the following:

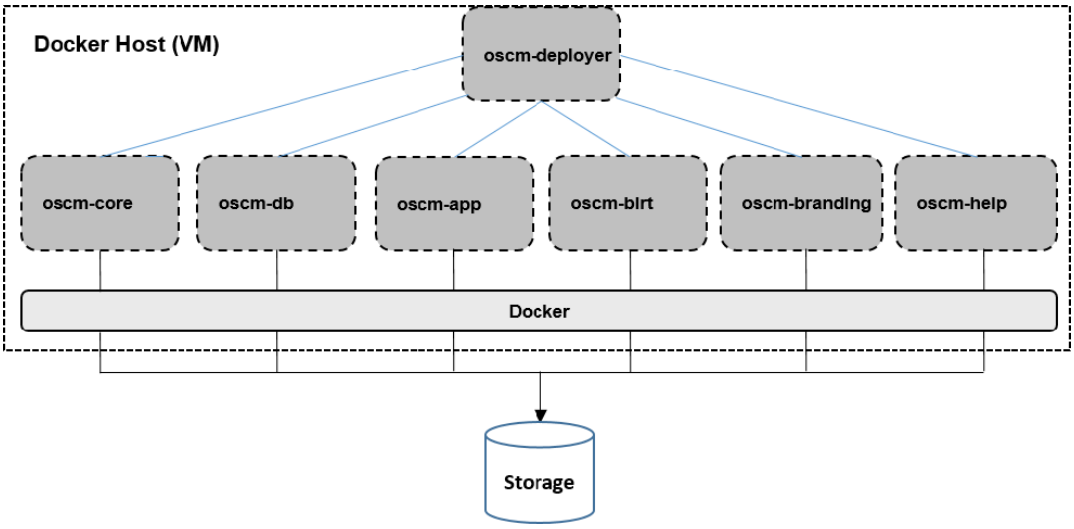
- Administration of the operating systems in use, including the adaption and execution of batch files or shell scripts.
- Java EE technology, particularly as to the deployment on application servers.
- Container technology, particularly Docker and Docker Compose.
- OSCM concepts as explained in the Overview manual.

# 1 OSCM Containers

OSCM is provided in Docker containers and deployed in a container environment such as OpenStack or a Kubernetes cluster providing for optimum performance, scalability, failover, and non-stop operation. The applications integrated with OSCM and their data may be hosted in the same Virtual Machine (VM) as OSCM or in different locations.

The `oscm-deployer` container is used for the configuration and deployment of the following OSCM containers:

- `oscm-core`: The OSCM core application.
- `oscm-birt`: The report engine that OSCM uses for generating reports.
- `oscm-branding`: A static Web server providing an empty directory structure for customizing the layout and branding of OSCM marketplaces.
- `oscm-help`: A static Web server providing the online help for the OSCM administration portal and marketplaces.
- `oscm-app`: The Asynchronous Provisioning Platform (APP) together with an OpenStack, an Amazon Web Services (AWS), and a VMware service controller.
- `oscm-db`: Database SQL server providing the database schema required for running OSCM and APP.
- `oscm-proxy`: A static Web server providing reverse proxy functionality.



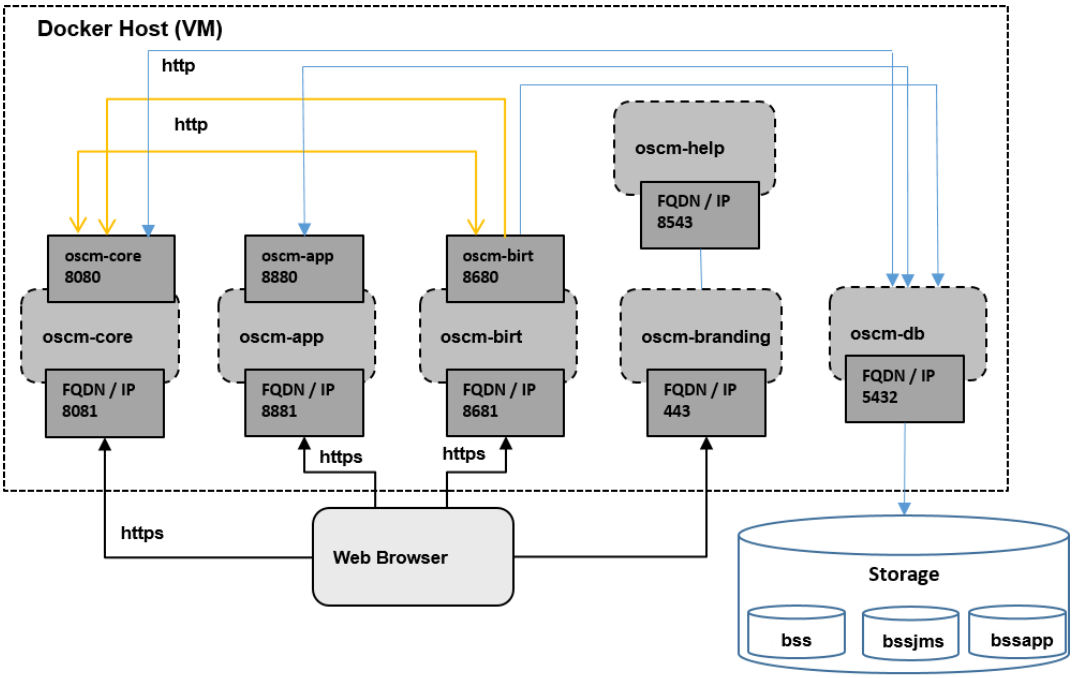

OSCM and APP store their data in PostgreSQL databases. For each database, a volume for persistent storage is mounted during the deployment process. The databases write to these volumes so that data is preserved in case of database updates. By default, data is persisted in the following directory on the machine hosting the VM where OSCM is deployed (the Docker host):

`<docker>/data/oscm-db/data:/var/lib/postgresql/data`

Hereafter, the `<docker>` directory is referred to as `/docker`.

## Container Communication

The following figure provides an overview of the container communication on a Docker host (a VM):




The host-internal communication always relies on the HTTP protocol, whereas calls from outside of the host are always secured via HTTPS. The platform operator is responsible for opening the indicated ports, which can either be addressed using the FQDN or the floating IP address of the respective container.

To look inside a container:

1. List all containers (even stopped ones) to show the container names:

```
docker ps -a
```

2. Log in to a container using the following command:

```
docker exec -it <container name> /bin/bash
```

For example:

```
docker exec -it oscm-core /bin/bash
```

# 2 Prequisites and Preparation

For getting started with the setup of OSCM, a Linux system is required with the following software installed:

1. Docker
2. Docker Compose

This system is hereafter referred to a the *Docker host*.

The following system resources are recommended for the Docker host:

1. 2 CPU cores
2. 8 GB of RAM
3. 20 GB of disk space

| | |
|---|---|
| **Note:** | This is a minimum configuration for testing purposes only. It is not suitable for production use. |

# 3 Installation

The installation of OSCM consists of the following main steps:

1. Preparing the installation directory on the Docker host.
2. Preparing configuration files.
3. Preparing Docker Compose files and starting OSCM.

## 3.1 Preparing the Installation Directory

On the Docker host, you need to create a directory where the various data can be stored, such as persistent database data or configuration files. Hereafter, this directory is referred to as `/docker`.

On the Docker host, execute the following command:

```
mkdir /docker
```

## 3.2 Preparing Configuration Files

A deployment container is available which prepares configuration file templates where you can specify settings such as mail server, login information, etc.

Use `-v` to mount the `/docker` directory to `/target` in the container:

```
docker run --name deployer1 --rm -v /docker:/
    target servicecatalog/oscm-deployer
```

This command creates two configuration files in the `/docker` directory:

1. `.env`: Configuration settings for Docker, such as images and the base data directory (default: `/docker`)
2. `var.env`: Configuration settings for OSCM and APP, such as mail server, database and other settings.

Edit both files and adjust the configuration settings to your environment. The initial, mandatory settings are stored in the `bss` and `bssapp` databases. They are described in detail in *Initial Configuration Settings* on page 12.

## 3.3 Preparing Docker Compose Files and Starting OSCM

A second deployment container is available which you can run to do the following:

1. Create the necessary Docker Compose files.
2. Create the necessary subdirectories.
3. Initialize the application databases.
4. Start the application containers.

Execute the following command on your Docker host:

```
docker run --name deployer2 --rm -v /docker:/target
    -v /var/run/docker.sock:/var/run/docker.sock
    -e INITDB=true -e STARTUP=true servicecatalog/oscm-deployer
```

# 4 Integrating Certificates for Trusted Communication

Certificates are required for OSCM to allow for trusted communication between OSCM and the Asynchronous Provisioning Platform (APP), or an application underlying a technical service . The OSCM deployer has already created a respective directory structure and a suitable Docker Compose configuration. In this way, default certificates have been inserted into the respective containers after deployment, thus communication between OSCM and APP is secured.

It is however possible to use custom SSL keypairs for the application listeners. They may be self-signed or official. Privacy Enhanced Mail (PEM) format is mandatory. This is a container format that may include just the public certificate, or may include an entire certificate chain including public key, private key, and root certificates. It is only necessary to place the respective certificate and/or key files in PEM format into the appropriate directories.

## 4.1 Importing SSL Key pairs

If you want to use your own SSL key pairs that your application is to use, replace the default key pair by your PEM files in the following directories on your Docker host:

Private key: `/docker/config/<container-name>/ssl/privkey`

Public certificate: `/docker/config/<container-name>/ssl/cert`

Intermediates / chain (optional): `/docker/config/<container-name>/ssl/chain`

Replace `/docker` with the base directory on the Docker host that you use for persisting data, and `<container-name>` with the name of the respective OSCM container, for example, `oscm-core` or `oscm-app`.

The custom certificates must also be placed into the following trusted directory so that a trusted relationship between the containers is established:

`/docker/config/certs`

The `/docker/config/certs` directory is shared by all containers. By default, if you use your own SSL key pairs, you must also place all the public certificate files here.

For example, if you have a custom SSL keypair for the `oscm-core` container, you need to place the private key into the `/docker/config/oscm-core/ssl/privkey` directory, and the public certificate into the `/docker/config/oscm-core/ssl/cert` directory. Additionally, you need to place the public certificate into the `/docker/config/certs` directory on your Docker host. In this case, a restart of the `oscm-core` and `oscm-app` containers is required.

## 4.2 Importing Trusted SSL Certificates

If you want your application to trust certain, possibly self-signed SSL certificates, put them in PEM format in the following directory on your Docker host:

`/docker/config/certs`

Replace `/docker` with the base directory on the Docker host that you use for persisting data.

For example, if you want to use the VMware service controller, you need to export the vSphere certificate in PEM format, and copy it to the `/docker/config/certs` directory. Since the VMware service controller is running in the `oscm-app` container, a restart of this container is required.

# 5 Usage

## 5.1 Accessing the OSCM Administration Portal

After having run the second deployment container, OSCM will take several minutes to start up. The less CPU power you have, the longer it will take. Once everything has started, you may access the OSCM administration portal in your Web browser using the FQDN or IP address you specified earlier in the `var.env` file.

Access the OSCM administration portal in a Web browser using an URL in the following format:

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

`<hostname.fqdn>` is the name and the fully qualified domain name of the machine where OSCM has been deployed. `<port>` is the port to address the machine (default: `8081`), `oscm-portal` is the default context root of OSCM and cannot be changed.

You are prompted for the user ID and password. The initial credentials are as follows:

User ID: `administrator`

Password: `admin123`

It is recommended that you change the initial password in the OSCM administration portal (**Change Password** page in the **Account** menu).

After login, the operator functionality is available in the **Operation** menu.

## 5.2 Enable Login to APP and Service Controllers

In order to be able to login to the Asynchronous Provisioning Platform (APP) and its service controllers, some settings have to be made in the administration portal:

1. Choose **Manage organization** in the **Operation** menu.
2. Enter `PLATFORM_OPERATOR` in the **Organization ID** field.
3. Enable the following organization roles: **Supplier** and **Technology provider**
4. Fill in the mandatory fields (red asterisks)
5. Click **Save**
6. Go to the **Account** menu and choose **Manage users**
7. Click on `administrator`
8. Enter your Email address
9. Enable all user roles.
10. Click **Save**
11. Logout of the administration portal and login again to enable the changes.

Now you are able to login to APP:

`http://<hostname.fqdn>:8880/oscm-app/`

User name: `administrator`

Password: `admin123`

You can also login to the instance status interface of the service controllers:

`http://<hostname.fqdn>:8880/oscm-app-<controller-name>/`

User name: `administrator`

Password: `admin123`

`<controller-name>` can be `openstack`, `aws`, or `vmware`.

# Appendix A: Initial Configuration Settings

### .env File

The configuration settings that must be set in the `.env` file are by default:

- Which Docker images to use and from which registry they are to be fetched:
  - `IMAGE_DB`: image for the `oscm-db` container
  - `IMAGE_CORE`: image for the `oscm-core` container
  - `IMAGE_APP`: image for the `oscm-app` container
  - `IMAGE_BIRT`: image for the `oscm-birt` container
  - `IMAGE_BRANDING`: image for the `oscm-branding` container
  - `IMAGE_INITDB`: image for the temporary, stateless `oscm-initdb` container
  - `IMAGE_PROXY`: image for the `oscm-proxy` container
  - `IMAGE_HELP`: image for the `oscm-help` container
- The base directory where data is persisted on your Docker host. The default directory is `/docker` as defined with the `DOCKER_PATH` configuration setting.

### var.env File

The configuration settings that must be set in the `var.env` file described in detail below.

### Settings for Connecting to the Databases

- `DB_PORT_CORE`

  The port of the PostgreSQL database (`bss`) where the configuration settings for the `oscm-core` container are stored.

  Must be set to `5432`.

- `DB_PORT_JMS`

  The port of the PostgreSQL database (`bssjms`) where the JMS messages are stored.

  Must be set to `5432`.

- `DB_PORT_APP`

  The port of the PostgreSQL database (`bssapp`) where the configuration settings for the `oscm-app` container are stored.

  Must be set to `5432`.

- `DB_PWD_CORE`

  Specify the password of the user to connect to the `bss` database.

  Default: `bssuser`

- `DB_PWD_APP`

  Specify the password of the user to connect to the `bssapp` database.

  Default: `bssappuser`

- `DB_SUPERPWD`

  Specify the password of the PostgreSQL database superuser.

  Default: `postgres`

### Settings for Connecting to the Mail Server

- `SMTP_HOST`

  The host name or IP address of your mail server used for notifications by OSCM.

- `SMTP_PORT`

  The port used by your mail server.

- `SMTP_FROM`

  The email address to be used for emails sent by OSCM.

- `SMTP_USER`

  If your mail server requires authentication, the name of the user allowed to access the mail server. If no authentication is required, set to `none`.

- `SMTP_PWD`

  If your mail server requires authentication, the password of the user allowed to access the mail server. If no authentication is required, set to `none`.

- `SMTP_AUTH`

  Defines whether your mail server requires authentication. Can be set to `true` or `false`.

- `SMTP_TLS`

  Defines if TLS (Transport Layer Security) is to be used for mail server communication. Can be set to `true` or `false`.

- `APP_ADMIN_MAIL_ADDRESS`

  The email address used for emails sent by the Asynchronous Provisioning Platform (APP). For example, this is the email address of the administrator of the technology provider organization responsible for the OpenStack service controller.

### Settings for Connecting to the Service Controllers

- `CONTROLLER_ID`

  The identifier of the service controller to be configured. Currently, the OpenStack, the AWS, and the VMware service controllers can be configured with the following identifiers: `ess.openstack`, `ess.aws`, or `ess.vmware`, respectively.

- `CONTROLLER_ORG_ID`

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

  Before the first deployment, the setting's value must be set to `PLATFORM_OPERATOR`. The operator can then log in to the service controller and define the technology provider organization that is to be responsible.

- `CONTROLLER_USER_KEY`

  The identifier of the user for accessing the service controller.

  Before the first deployment, the setting's value must be set to `1000`.

- `CONTROLLER_USER_NAME`

  The name of the user for accessing the service controller. The user specified here must have the technology manager role in OSCM and belong to the organization specified in the `CONTROLLER_ORG_ID` setting. It is recommended that the user account is used only for carrying out actions on behalf of the service controller in OSCM.

Before the first deployment, the setting's value must be set to `administrator`.

- `CONTROLLER_USER_PASS`

  The password of the user for accessing the service controller. The password is encrypted when it is stored in the database.

  Before the first deployment, the setting's value must be set to `admin123`.

## Settings for Connecting to the VMware Service Controller

The VMware service controller requires additional settings.

- `DB_USER_VMWARE`

  The name of the user to connect to the `vmware` database. This database stores the vSphere configuration.

- `DB_PWD_VMWARE`

  The password of the user to connect to the `vmware` database. This database stores the vSphere configuration.

- `VCENTER_NAME`

  The name of the vCenter running on the vSphere server.

- `DATACENTER_NAME`

  The name of data center managed by the vCenter specified above.

- `CLUSTER_NAME`

  The name of the cluster where VMs are to be provisioned.

- `LOADBALANCER_NAME`

  The name of the load balancer used by vSphere. Usually, you can use the name of the VM Network in vSphere.

## Additional Settings

- `KEY_SECRET`

  A string which will be used in the `bss` database as a seed for encryption and decryption of service parameters with data type `PWD` and custom attributes marked for encryption. Make sure not to forget this string so that your database is persisted.

**Note:** Be aware that you must use the same key when updating the database. Otherwise, encryption and decryption is no longer possible after an update.

- `HOST_FQDN`

  The fully qualified domain name or the IP address of the host to be used to access OSCM.

- `REPORT_ENGINEURL`

  The URL of the report engine used to generate the OSCM reports. If you do not specify a correct URL template, OSCM will not be able to generate any reports, since the Report Web service cannot be called correctly.

  The setting must be as follows:

```
https://<HOST_FQDN>:8681/birt/frameset?
   __report=\${reportname}.rptdesign&SessionId=\${sessionid}&
   __locale=\${locale}&WSDLURL=\${wsdlurl}&SOAPEndPoint=\
```

```
${soapendpoint}&wsname=Report&wsport=ReportPort
```

Replace <HOST_FQDN> with the value entered for the HOST_FQDN setting. The other values must not be changed.

- TOMEE_DEBUG

  Defines if the application server is to generate log files for OSCM. Can be set to true or false.

## Application Server Settings

- CONTAINER_CALLBACK_THREADS

  The number of threads for constructing and destroying beans. Required for the stateless init-db container.

  Default: 50

- CONTAINER_MAX_SIZE

  The size of the instance pool for the stateless init-db container.

  Default: 50

The following settings are used for the Docker engine environment which needs access to the Docker registry as well as for the OSCM

- PROXY_ENABLED

  Defines whether the proxy is enabled in the container. Can be set to true or false.

- PROXY_HTTP_HOST

  Host of the proxy used for HTTP connections

- PROXY_HTTP_PORT

  Port of the proxy used for HTTP connections

- PROXY_HTTPS_HOST

  Host of the proxy used for HTTPS connections

- PROXY_HTTPS_PORT

  Port of the proxy used for HTTPS connections

- PROXY_NOPROXY

  Pipe-separated list of hosts for which the proxy is to be bypassed. By default, this list already contains localhost, 127.0.0.1, and the floating IP address of the OSCM instance.