V17.6.0

QuickStart Guide

January 2018

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 .

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:

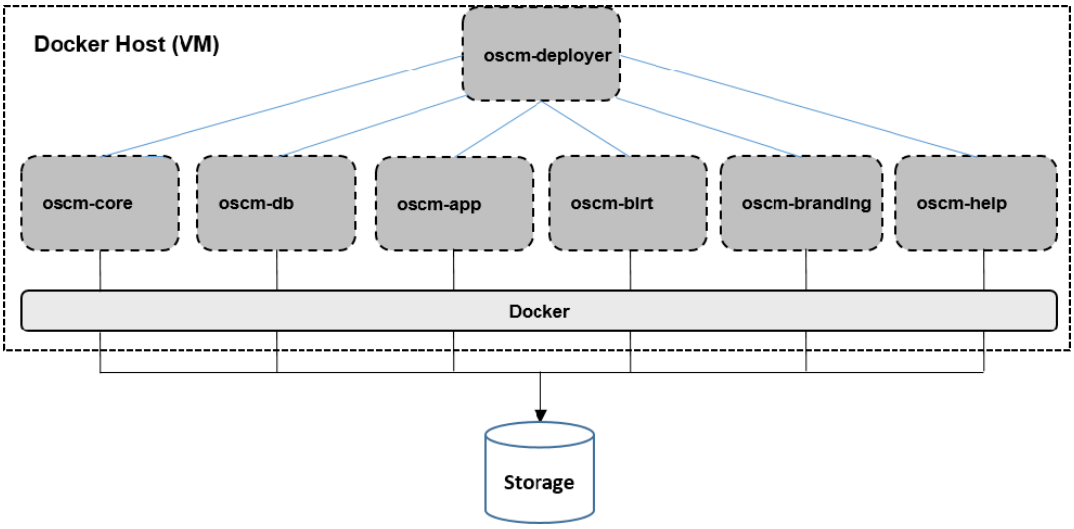
- Administration of the operating systems in use, including the adaption and execution of batch files or shell scripts.
- Java EE technology, particularly as to the deployment on application servers.
- Container technology, particularly Docker and Docker Compose.
- OSCM concepts as explained in the *Overview*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



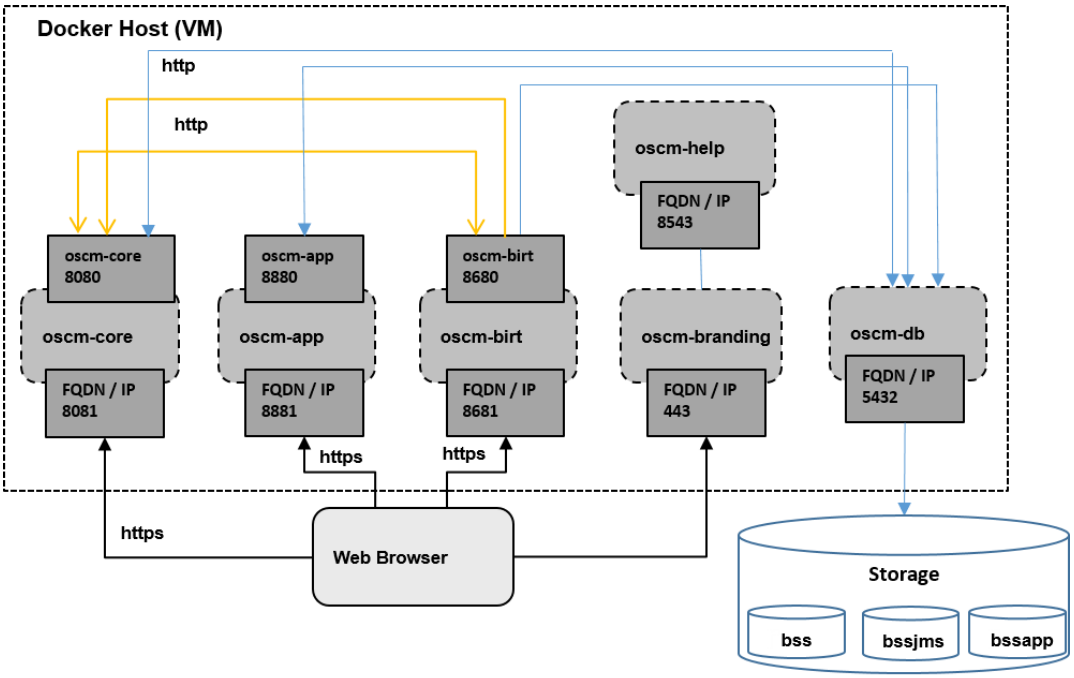
OSCM and APP store their data in PostgreSQL databases. For each database, a volume for persistent storage is mounted during the deployment process. The databases write to these volumes so that data is preserved in case of database updates. By default, data is persisted in the following directory on the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

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

where `<docker>` is the base directory where Docker is installed and where data is persisted. Hereafter, this 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 Prerequisites and Preparation

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

1. Docker
2. Docker Compose

This system is hereafter referred to as 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 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 for Docker: where are the images to be retrieved, where is the base data directory.
2. `var.env`: Configuration for ESCM, such as mail server, database and other settings. Refer to the *Operator's Guide* for details on the configuration settings.

Edit both files and adjust the configuration settings to your environment.

3.3 Preparing Docker Compose Files and Starting ESCM

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 Usage

4.1 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 ESCM 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.

4.2 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 the APP:

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

User name: `administrator`

Password: `admin123`

You can also login to the OpenStack service controller:

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

User name: `administrator`

Password: `admin123`

5 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.

5.1 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`

Certificate: `/docker/config/<CONTAINER_NAME>/ssl/cert`

Intermediates / chain (optional): `/docker/config/<CONTAINER_NAME>/ssl/chain`

Replace `/docker` with the directory where Docker is installed, 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`

5.2 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 directory where Docker is installed.

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.