V17.6.0

QuickStart Guide

December 2017 - Draft for review

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 . This is a quick start guide intended to help you start up a basic installation of OSCM with Docker and Docker Compose as quickly as possible. For more advanced configuration and usage, refer to the individual Docker containers' documentation on DockerHub. ### insert link to fest ### on DockerHub.

Readers of this Manual

This manual is directed to operators who deploy, configure, use, and setup OSCM on a Docker host. 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 Introduction

For initial testing, 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*.

For testing purposes, the following system resources are recommended:

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

Note that this minimum configuration is not suitable for production use.

2 Setup

This chapter describes the installation and configuration of OSCM.

2.1 Preparing the Installation Directory

On the Docker host, you need to create a directory which holds various data, such as persistent database data or configuration data. In this document, this directory is referred to as `/docker`:

On the Docker host, execute the following command:

```
mkdir /docker
```

2.2 Preparing Configuration Files

A deployment container is available which can be run to prepare configuration file templates.

Use `-v` to mount the directory you created earlier 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, such as images and 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.

2.3 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
```

3 Usage

3.1 Login to the Administration Portal

OSCM will take a few 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.

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.

3.2 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`

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`

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`

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 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.