UBIQUBE

Quickstart Guide

MSactivator

Table of Contents

This documentation covers how you can quickly get started using MSactivator™ by installing a Docker based mini lab and start testing the product.

Alternatively, if you want to test MSactivator™ without having to go though the installation process, you can register for a free access to our hosted trial setup

Prerequisites

The following prerequisites are required for a successful use of MSactivator™:

1. A recent version of Git.
2. A working docker and docker compose setup
    - Docker
    - Docker Compose

Setup requirements

Internet Resources

Ensure you have access from HOST machine(s) to:

- https://github.com
- https://hub.docker.com

Install the MSactivator™ behind a proxy

Often the MSactivator™ has to be installed on a host that sits behind a proxy. This requires some specific configuration on the host, on docker and also on the MSactivator™ containers themselves:

Proxy setting on the host OS where the proxy needs to be specified and used. This is needed to download all the necessary docker packages as well as docker compose on the machine. Those settings are different depending on the OS that might be used.

Once installed, docker does not inherit the proxy settings from the host machine. Proxy settings need to be specified in dockers configuration file to download the necessary MSactivator™ images.

However, along with that, the "no proxy" option should be specified for the internal container to containers communication. Falling to do so will result in all the traffic routed the proxy.

The easiest way to do that is to edit the file docker-compose.yml and add the following to the services `msa-dev`, `msa-api` and `msa-sms`:

```
environment:
    http_proxy: "<PROXY URL>"
    https_proxy: "<PROXY URL>"
    no_proxy: "localhost,127.0.0.1,linux_me_2,linux_me,msa-cerebro,camunda,msa-
alarm,db,msa-sms,msa-ai-ml,msa-ui,msa-dev,msa-bud,msa-kibana,msa-es,msa-front,msa-api"
```

If your proxy also rewrites the SSL certificates you will face errors such as `fatal: unable to access 'https://github.com/openmsa/Workflows.git/': Peer's Certificate issuer is not recognized.` when installing the MSactivator™. To solve this you also need to add the following to the `msa_dev` and `msa_api` environment.

```
    GIT_SSL_NO_VERIFY: "true"
```

Hardware Resources

on a PC (for lab experimentation)

Minimum RAM memory to allocate to Docker:

- On a 8GB RAM PC: allocate 4GB, 2 CPU, 50 GB of disk space.
- On a 16GB RAM PC: you can allocate 6-8GB, 2+ CPU, 50 GB of disk space.

with less that 4GB of memory, the MSactivator™ may not function properly: some container such as the API container may not be able to start causing issues when connecting to the UI.

allocating more CPU might be tempting but before you do that you need to make sure that your system has enough memory. For instance, if you allocate 4 CPU to Docker, you need to allocate at least 8GB of memory.

if you are using Docker on MacOS, this guide will help you with this configuration.

if you are using Docker on Windows, this guide will help you with this configuration.

on a production environment

- minimum 16GB, 4CPU, 200GB

Docker optional custom configuration

The custom configuration below may not be needed depending on your host computer and you intended use of the MSactivator™. If you are just running the product for evaluation purpose or training, don't worry too much about these at first.

In a production environment, these custom configuration may be mandatory to allow your setup to support a higher workload.

Docker for Linux (or Linux VM)

execute as root

```
sysctl -w vm.max_map_count=262144
touch /etc/sysctl.d/50-msa.conf
echo 'vm.max_map_count = 262144' > /etc/sysctl.d/50-msa.conf
sysctl -p /etc/sysctl.d/50-msa.conf
```

Docker for Mac

From the command line, run:

```
screen ~/Library/Containers/com.docker.docker/Data/vms/0/tty
```

Press enter and use `sysctl` to configure vm.max_map_count:

```
sysctl -w vm.max_map_count=262144
```

To exit the screen session, type Ctrl a d.

Docker for Windows

```
docker-machine create -d virtualbox  \
    --virtualbox-cpu-count=2 \
    --virtualbox-memory=8192 \
    --virtualbox-disk-size=50000 \
    default
```

In the docker VM, do as for Linux host above:

```
sudo sysctl -w vm.max_map_count=262144
sudo tee -a /etc/sysctl.conf <<< "vm.max_map_count=262144"
```

Mini lab creation

Step 1: creation

Create the mini lab: clone the quickstart git repository from Github, download and run the MSactivator™ from DockerHub.

1. `git clone https://github.com/ubiqube/quickstart.git`
2. `cd quickstart`
3. `./scripts/install.sh` (or `./scripts/install_win.sh` for Windows users)

If you are already running the MSactivator™ and wish to upgrade it to a new version check the documentation below

The architecture of the mini lab is detailed in the architecture overview documentation in the admin guide.

Step 2: install the trial license

Contact your sales representative to request for a trial license

Open https://localhost/ and connect with username `ncroot` and password `ubiqube`.

Browse to "Settings" in the left menu and upload the free trial license you received.

use localhost if you are using your PC as the docker host, otherwise use the IP address assigned to the docker host.

Step 3: provision the mini lab with tenants, managed entities, etc.

The mini lab comes with 2 Linux container (linux_me / 172.20.0.101 and linux_me_2 / 172.20.0.102) for experimentation.

Credentials for the Linux machine are:

- username: `msa`
- password: `ubiqube`

In order to ease your discovery of MSactivator™, we are providing a script that will create the mini lab environment for you:

- 1 tenant - BladeRunner
- 1 subtenant - Tyrell Corporation

- 2 managed entities to manage the Linux container - linux_me
- some microservices and workflows to start configuring the managed entity - users, firewall, etc.

To create the mini lab environment, run the CLI command from where you have cloned the quickstart Github project:

```
docker compose exec msa-dev /usr/bin/create_mini_lab.sh
```

Step 4: change default password of super admin(ncroot)

Browse: https://localhost/ and connect with username `ncroot` and password `ubiqube`.

After Login, Go to Profile tab and Edit Profile. Update new password for `ncroot` and do Logout.

Step 5: start using the MSactivator™

Browse: https://localhost/ and connect with username `ncroot` and updated password in previous step.

Mini lab detailed description

Use case 1: firewall orchestration on Linux

This lab use case will show you in a simple way how you can use MSactivator™ to automate the configuration of iptables-based firewall policy on the Linux containers included in the mini lab.

The development of this use case is detailed in this guide: Firewall Policy Automation

You can use the guide to recreate the use case step by step or you may also directly run the use case with the workflow and microservices that are installed in the mini-lab.

For that, you'll have to select the subtenant named "TYRELL CORPORATION," click on the link "Automation" on the left menu and select the tab "Workflows." The workflow to use is "Simple Firewall (Python)."

Mini lab additional info

Fresh install or upgrade ?

The quickstart repository is maintained on https://github.com/ubiqube/quickstart, the latest version is tagged as `tags/MSA-2.8.10`

If you are upgrading your MSactivator™, the best and easiest option is to remove your Quickstart git repository and use `git clone` to get the new version.

In case you have updated the MSactivator™ with `docker compose up -d` or `docker compose pull`, you might experience cache issues (for instance, your changes may not be reflected on the UI). To solve that, you can clean your browser cache, or use a browser private session.

Upgrade version N to N+1

Starting from it's version 2.2.0, the quickstart project provides a script `upgrade.sh` for taking care of possible upgrade actions such as recreate some volumes, execute some database specific updates, update the libraries,...

Let's say that you are running the version MSA-2.8.9, to upgrade to the version MSA-2.8.10 you need to follow these steps:

1. `cd quickstart`
2. `git checkout master`
3. `git pull`
4. `./scripts/install.sh`

when running the upgrade script, it will ask you if you want your local libraries to be updated with the latest version from the community. If you answer 'y', the update will be done automatically. You can also do the update later manually on the container `msa_dev`.

```
# ./scripts/install.sh --help
usage: install.sh [--mini-lab|-m] [--force|-f] [--cleanup|-c] [--remove-orphans|-ro]
this script installs and upgrades a {product_name}

-m: mini lab creation. Create a demo platform around a Linux ME
-f: force the upgrade without asking for user confirmation. Permit also to reapply the
upgrade and to auto merge files from OpenMSA
-c: cleanup unused images after upgrade to save disk space. This option clean all
unused images, not only MSA quickstart ones
-ro: remove containers for services not defined in the compose file. Use it if some
containers use same network as MSA
```

```
-mano : apply mano containers
```

Note for Windows user

If your original version is 2.1, plz refer to upgrade script first provided with 2.2.0 GA. In any cases, perform those commands

1. `cd quickstart`
2. `git checkout master`
3. `git pull`
4. `./scripts/install_win.sh`

NOTE : you may face an issue at the end of the script with a message "Wait Kibana to be ready". This is a known issue, the workaround is to execute `docker compose up -d` from another CLI.

Install a specific version

For each release of the MSactivator™, there is a tag that you can use if you need to install a specific version of the product.

To install a tagged version, you can checkout the tag and go to the install steps above.

For example, the CLI command below will checkout the quickstart for the release MSA-2.8.10.

```
git clone -b MSA-2.8.10 https://github.com/ubiqube/quickstart.git
```

The releases and tags are available here

install_libraries script

The script `install_libraries.sh` is installed in the container msa_dev.

This script is designed to populate the libraries for a fresh install or to update your libraries with the latest version from Github.

```
$ docker compose exec msa-dev /usr/bin/install_libraries.sh --help
usage: install_libraries.sh all|ms|wf|da|py|mano|quickstart [--lic] [-y]

this script installs some libraries available @github.com/openmsa

Commands:
all:          install everything: workflows, microservices and adapters
ms:           install the microservices from https://github.com/openmsa/Microservices
wf:           install the workflows from https://github.com/openmsa/Workflows
da:           install the adapters from https://github.com/openmsa/Adapters
mano:         install/update the python-sdk from https://github.com/openmsa/etsi-mano
py:           install/update the python-sdk from https://github.com/openmsa/python-sdk
quickstart:   install/update the local quickstart from
```

```
https://github.com/ubiqube/quickstart

Options:
--lic:     force license installation
-y:           answer yes for all questions
```

In case of calling this script on an existing setup, it will take care of merging the code from the Github master branch into your local development branch. With the option `-y`, an automated merge will be attempted, without the option, the script will ask for user input.