



TUTORIAL OF A DOCKER- SWARM CLUSTER DEPLOYMENT

Docker-Swarm cluster deployment tutorial for MTA
Cloud users v1.0

Table of contents

Overview.....	2
Steps	2
Installation of Occopus.....	2
Download	2
1. Fill the resource section	3
2. Creating firewall rules	3
3. Authentication.....	3
4. Set the size of the cluster	3
5. Import.....	4
6. Deployment	4
7. The established infrastructure	4
8. Delete the infrastructure.....	4
Report bugs or any other project related claims	4

Overview

This tutorial sets up a complete Docker infrastructure with Swarm, Docker and Consul software components. It contains a master node and predefined number of worker nodes. The worker nodes receive the ip of the master node and attach to the master node to form a cluster. Finally, the docker cluster can be used with any standard tool talking the docker protocol (on port 2375).

Features

- creating two types of nodes through contextualisation
- passing ip address of a node to another node
- using the occi resource handler
- utilising health check against a predefined port
- using parameters to scale up worker nodes

Prerequisites

- accessing an Occopus compatible interface
 - Occopus tool can launch virtual machines on MTA Cloud, and we recommend the use of nova interface (the url of the nova interface can be found under Compute/ Access & Security / API Access / Identity menu)
 - **Important: the use of Occopus tool currently works only on the SZTAKI branch of the MTA Cloud**
- target cloud contains an Ubuntu 14.04 image with cloud-init support
 - This image file can be found under Compute/Images/Public/Ubuntu 14.04 LTS image

Steps

Installation of Occopus

The deployment of the Docker-Swarm cluster will be established with the help of Occopus tool, therefore we need to install the Occopus tool first. You can install the Occopus orchestration tool with just one command. For more information about Occopus itself, and how to install it, visit the [following link](#). We recommend launching an Ubuntu-based virtual machine in MTA Cloud to install the Occopus tool on it.

Download

Occopus works based on descriptors. We have prepared the descriptors for the installation of the Docker-Swarm cluster for the end-users. Based on these descriptors, Occopus will build the infrastructure in the target cloud. They can be downloaded from the following link: [tutorial.examples.docker-swarm](#). Install descriptors on a virtual machine which runs Occopus.

Note: In this tutorial it uses CloudSigma resources. However, feel free to use any Occopus-compatible cloud resource for the nodes, but we suggest to instantiate all nodes in the same cloud.

1. Fill the resource section

Open the file `nodes/node_definitions.yaml` and edit the resource section of the nodes labelled by `node_def:`.

- you must select an [Occopus compatible resource plugin](#)
 - this will be the nova resource plugin
- you can find and specify the relevant [list of attributes for the plugin](#)
- you may follow the help on [collecting the values of the attributes for the plugin](#)
- you may find a resource template for the plugin in the [resource plugin tutorials](#)

It is important that end-users should personalize the node definition file to the user before launching. In this file, we add the resource identifiers we will use, such as project ID, virtual machine size, and so on. We can not provide these identifiers for the user, but they can be easily collected from the MTA Cloud web interface. For detailed assistance, visit [this link](#) or the [documentation below](#). The downloadable package in this example contains the Nova plugin resource template (also used for MTA Cloud). The downloadable package for this example contains a resource template for the Nova plugin.

2. Creating firewall rules

Components in the infrastructure connect to each other, therefore several port ranges must be opened for the VMs executing the components. Log in to the MTA Cloud OpenStack interface. Under "Compute / Access & Security" you can create a new firewall rule by clicking the "Create Security Group" button. After creation, you can edit the firewall rule by clicking the "Manage Rules / Add Rule" button. Add the following ports to the security group:

- TCP 2375
- TCP 2377
- TCP 7946
- The ports which is needed to be opened for your Docker application

3. Authentication

Make sure your authentication information is set correctly in your authentication file. You must set your email and password in the authentication file. Setting authentication information is described [here](#).

Load the node definition for `dockerswarm_master_node` and `dockerswarm_worker_node` nodes into the database.

4. Set the size of the cluster

Update the number of worker nodes if necessary. For this, edit the `infra-docker-swarm.yaml` file and modify the `min` parameter under the `scaling` keyword. Currently, it is set to 2.

```
- &W
  name: worker
  type: dockerswarm_worker_node
  scaling:
    min: 2
```

5. Import

Load the node definitions into the database. Make sure the proper virtualenv is activated!

```
occopus-import nodes/node_definitions.yaml
```

Make sure that the proper virtualenv is activated! If you have not done this before, use the following command to activate the Occopus virtual environment:

```
source occopus/bin/activate
```

Important: Occopus takes node definitions from its database when builds up the infrastructure, so importing is necessary whenever the node definition (file) changes!

6. Deployment

Start deploying the infrastructure.

```
occopus-build infra-docker-swarm.yaml
```

Note: It may take a few minutes until the services on the master node come to live. Please, be patient!

7. The established infrastructure

After successful finish, the node with `ip address` and `node id` are listed at the end of the logging messages and the identifier of the newly built infrastructure is printed. You can store the identifier of the infrastructure to perform further operations on your infra or alternatively you can query the identifier using the **occopus-maintain** command.

```
List of nodes/ip addresses:
master:
  <ip-address> (dfa5f4f5-7d69-432e-87f9-a37cd6376f7a)
worker:
  <ip-address> (cae40ed8-c4f3-49cd-bc73-92a8c027ff2c)
  <ip-address> (8e255594-5d9a-4106-920c-62591aabd899)
77cb026b-2f81-46a5-87c5-2adf13e1b2d3
```

8. Check the cluster

Check the result by submitting docker commands to the docker master node!

9. Delete the infrastructure

Finally, you may destroy the infrastructure using the infrastructure id returned by `occopus-build`

```
occopus-destroy -i 77cb026b-2f81-46a5-87c5-2adf13e1b2d3
```

Report bugs or any other project related claims

Communication and support for MTA Cloud services are in the form of email. The common e-mail address is info@cloud.mta.hu. A notification form generated from this error will be generated by a designated member of the MTA Cloud team.