MICADO V3 basic

This tutorial will guide you through how to install MICADO and try it out with an example application. The tutorial builds a scalable architecture framework and performs the automatic scaling of the application based on Occopus, Docker Swarm and Prometheus.

**NOTE:** We advise you to use CloudSigma as a target cloud!

1. Prerequisites

Generally, MICADO requires the following requirements. Please make sure you provided these for the virtual machines where we will deploy MICADO.

* 1. Target cloud

You will need an account for a cloud which provides you an “Ubuntu 16.04” OS image with cloud-init support.

* accessing a cloud through an Occopus-compatible interface (e.g. EC2, OCCI, Nova, etc.)
* target cloud contains a base 16.04 LTS Ubuntu OS image with cloud-init support (image id, instance type)

1.2 Port ranges

While most of the clouds doesn’t require you to configure which ports you want to open (like Cloudsigma), is it still important to make sure that the following ports are open for MICADO:

* TCP:22,53,80,443,2375,2377,7946,8300,8301,8302,8400,8500,8600,9090,9093,9095,9100,9200
* UDP:7946,8301,8302,8600

1.3 internet access for the VMs

MICADO needs to pull some files from github and dockerhub. Make sure the virtual machines have internet access and also reach each other.

# 2. Deployment of MiCADO

## 2.1 Download MICADO

Please download the installation file of MICADO from the following link:

<https://goo.gl/ZhgvUm>

## 2.2 Insert your user inputs

Now you have to modify the file that you downloaded. In the beginning of the file you will see a section called “USER DATA” as shown in this code:

write\_files:

USER DATA - Cloudsigma

- path: /var/lib/micado/occopus/temp\_user\_data.yaml

content: |

user\_data:

auth\_data:

type: cloudsigma

email: YOUR\_EMAIL

password: YOUR\_PASSWORD

resource:

type: cloudsigma

endpoint: YOUR\_ENDPOINT

libdrive\_id: UBUNTU\_16.04\_IMAGE\_ID

description:

cpu: 1000

mem: 1073741824

vnc\_password: secret

pubkeys:

-

KEY\_UUID

nics:

-

ip\_v4\_conf:

conf: dhcp

conf: dhcp

scaling:

min: 1

max: 10

This file specifies the user credentials for the target cloud, the resource IDs that will be used for the Virtual machines and a scaling section which specifies the scaling ranges. We provided you multiple configuration for the CloudSigma, Openstack, Amazon AWS.

The user can choose from these different configurations depending on the target cloud. Please uncomment the one you will use and fill out the parameters!

When you are ready, save the file and exit.

## 2.3 Check the syntax

Before deploying MICADO we advise you to check the syntax of your file. Since it is a yaml formatted file you should make sure of the syntax. To do so just copy paste your MICADO file to an online yaml checker: <http://www.yamllint.com/>

## 2.4 Start MICADO

To start MICADO click on the “Wizzard” button on the compute tab if you use CloudSigma.

* Choose the favour type “small-2”
* An Ubuntu 16.04 LTS Image
* Attach your ssh key
* Paste the previously downloaded file to the cloud-init box and activate it.
* The click on “Create”

**NOTE:** If you wish to use another cloud, the steps should be almost the same.

3. Deployment of Application

This part will guide you have to start an example application. We will use a stress testing application. It will stress test the cluster and the application will be overloaded automatically. MICADO will automatically adjust the resources and scale up both the number of application services running as Docker services, and also the number virtual machines on the cloud.

## 3.1 Create the Application

SSH inside your MICADO virtual machine on the Cloud, and run the following command as root.

$ ssh cloudsigma@[ipaddress\_of\_micado]

$ docker service create progrium/stress --cpu 2 --io 1 --vm 2 --vm-bytes 128M

4. Testing

This section is for testing purposes. If you experience some problems you can find out how to solve it here. TODO link for advanced guide

## 4.1 Test if the system is operational

On your browser enter the following URL:

http://ip\_address\_of\_MICADO\_VM:8500

You should see the web page of Consul. If you see every running service with a green box, you are good to go but if some of them are red, there are some problem. Also on the “nodes” tab you should see at least one node (MICADO + minimum number of scaling ranges you specified).

## 4.2 Test if scaling working properly

The stress testing application should automatically overload and stress down the worker cluster in a few minutes.

If everything went well in a few minutes you could see VMs booting on your cloud.

To check out the number of nodes after the scale up event in MICADO, check Prometheus on the following link:

<http://ip_address_of_MICADO_VM:9090/targets>

If you have more than 2 rows under the cluster, then it means that MICADO successfully scaled up the application nodes from the starting position that was two.

Now we also would like to test if it scales down if there is no load on the cluster. To do so, stop delete the application and this way delete the load on the cluster After a few minutes, the number of nodes in the cluster should be go back to its minimum value (specified in the user\_data, scaling part).

## 4.4 Delete your infrastructure

## When you are ready and wish to delete everything with the following command:

$ curl -X DELETE <http://[micado_master_ip]:5000/infrastructures/micado_worker_infra>

This will delete all the worker nodes. To delete MICADO itself, you have to do it by hand, and delete it on your cloud.

If you would like to run your own application in MICADO or need some help and more details, please visit the advanced user guide: TODO