MICADO V3 basic discr.

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 with the help of Occopus and performs the automatic scaling of the application based on Occopus, Docker Swarm and Prometheus. 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 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, 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 Install Occopus

To install MICADO, we will use a cloud orchestrator tool called Occopus. First you have to install Occopus to a new virtual machine on your cloud. Here you can find a step by step tutorial how to install and configure it:

<http://www.lpds.sztaki.hu/occo/user/html/setup.html>

Make sure you provided your user credentials to Occopus as you saw it on the Authentication part! When you are ready type:

$ Source /occopus/bin/activate

## 2.2 Download the project files

You can now download and decompress the project files to the same VM where Occopus is running:

TODO

## 2.2 Modify the node\_definition.yaml file

Now you have to modify the /master/nodes/node\_definition.yaml file that you downloaded. This file specifies the endpoint and cloud details of your target cloud. We provided you an already filled out configuration for the CloudSigma cloud. The only thing you have to change is the ID of your public key in the “pubkeys” section. You can find or create your key on Cloudsigma by clicking on the left-hand side on “Access and Security” and then select “key management on the CloudSigma website. You will find the uuid of you key there. When you are ready save the node\_definition file and exit.

## 2.3 Start MICADO

- Import the project files:

$ Occopus-import (location of the node\_definition.yaml )

- Build MICADO:

$ Occopus-build (location of the infrastructure\_description.yaml )

3. Deployment of Application

This part will guide you have to start an example application called Data Avenue (DA) into your auto scalable MICADO infrastructure. The application does simple file transfers from one data store to another. You will be able to try out and send a predefined large 1GB file to an SQL database. As more file transfers are running parallel the application will be overloaded, 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.

If you want to use your own application click here: TODO

## 3.1 Create an SQL DB for DA

SSH inside the VM where MICADO runs, then run the following command (requires root privileges:

$ docker run -d --name MYSQL\_DATABASE -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=dataavenue -e MYSQL\_USER=da -e MYSQL\_PASSWORD=da -p 3306:3306 mysql/mysql-server:5.5

## 3.2 Create Data Avenue

$ docker service create --limit-cpu=0.8 --publish 8080:8080 micado/dataavenue "ip\_address\_of\_your\_head\_node)

4. Testing

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

## 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 webpage 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, and here you can find help to solve it.

## 4.2 Test if the application running properly

You can reach the webpage of the Data Avenue application if you type on your browser the following URL:

http://ip\_address\_of\_MICADO\_VM/blacktop3

## 4.3 Test if scaling working properly

To test the auto scaling feature of MICADO we have to overload the application. To do so we will start many file transfers to our SQL database parallel. Run the following command in many instances on your terminal ( at least 10 copy ). Don’t forget to change the IP address in the command!

curl -k -o /dev/null -H "X-Key: 1a7e159a-ffd8-49c8-8b40-549870c70e73" -H "X-URI:https://autoscale.s3.lpds.sztaki.hu/files\_for\_autoscale/1GB.dat" http://[ MICADO IP address]/blacktop3/rest/file

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

To check out of the CPU usage of your application cluster check out the Prometheus webpage:

http://ip\_address\_of\_MICADO\_VM:9090 TODO

Also, you can check the running alerts in Prometheus, if you click on the “Alerts” tab. Here you can see the current status of your auto scaling events. You should be able to see alerts with different colours. Red means it’s actually in progress, Yellow is in pending state and green showing if nothing is happening.

To check out the number of nodes after the scale up event, click on the “targets” page. If you have more than one, then it means that MICADO successfully scaled up the application nodes.

Now we also would like to test if it scales down if there is no load on the cluster. To do so just stop all your file transfers. Now if you check the alerts you should be able to see the other alerts firing and on the “target” page that some of the targets are not reachable any more, meaning that they are deleted from the cloud.

MICADO will always leave one instance running with your application to ensure that it is reachable from the end user.

## 4.4 Delete your infrastructure

When you are ready and wish to delete everything, you can either select the VMs on the cloud and delete by hand or do it with the help of Occopus. SSH inside the VM running your Occopus installation and run the following commands:

Occopus-maintain -I

With the infrastructure ID in the output, run the following command:

Occopus-destroy -I “id of the infrastructure”

Need to distinguish levels:  
- Basic level description  
  + how to make it alive and use it  
- Detailed description  
  + internal details  
  
TODOS generally:  
- create a webpage with explaining what is MiCADO for devops people  
- restructure MiCADO tutorials, should not detail previous versions  
  may be one page is enough where overview is at the beginning  
  and add section which version to choose (if more than one available)  
- fix service docker start parameters (conflict: -p --publish)  
- fine tuning DNS configuration in cloud-init  
- find out why prometheus did not start in Nick's test  
- explain the expected output of "docker node ls" and "docker service ls"  
- ...