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 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 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

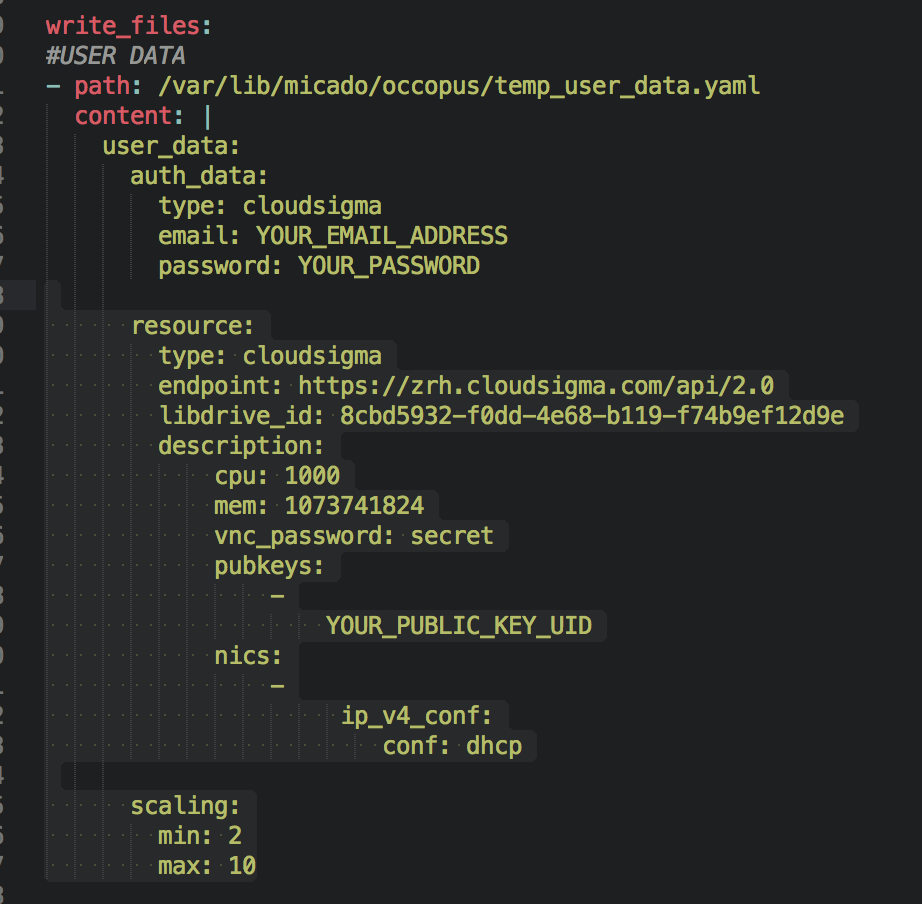
## 2.1 Download MICADO

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

<https://goo.gl/ZhgvUm>

## 2.2 Insert your user imputs

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 picture:

 This file specifies the user credentials for the tartget cloud, the resource IDs that will be used for the Virtual machines and a scaling section which specifies the scaling ranges. We provided you an already filled out configuration for the CloudSigma cloud. The only thing you have to change is your email and password credentials the UID 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 file and exit.

## 2.3 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 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.

## 3.1 Create Data Avenue

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

$ docker service create --publish 8080:8080 micado/dataavenue 172.31.0.6

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. Also on the “nodes” tab you should see at least 3 nodes (MICADO+min number of scaling ranges you specified).

## 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 the number of nodes after the scale up event, click on the “targets” page of Prometheus on the following link:

<http://ip_address_of_MICADO_VM:9090/targets>

If you have more than 2, then it means that MICADO successfully scaled up the application nodes from 2.

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. After a few minutes, the number of nodes in the cluster should be go back to its minimum (2 nodes).

## 4.4 Delete your infrastructure

When you are ready and wish to delete everything, you just need to select the VMs on the cloud and delete them.