TUTORIAL OF A HADOOP CLUSTER DEPLOYMENT

Hadoop cluster deployment tutorial for MTA Cloud users v1.0

MTA Cloud team
info@mta.cloud.hu

Table of contents

Overview

This tutorial sets up a complete Apache Hadoop infrastructure with the help of Occopus orchestration tool. It contains a Hadoop Master node and Hadoop Slave worker nodes, which can be scaled up or down. To register Hadoop Slave nodes Consul is used.

Features

- creating two types of nodes through contextualisation
- utilising health check against a predefined port
- using scaling parameters to limit the number of Hadoop Slave nodes
- manage cluster nodes with Consul

Prerequisites

- accessing a cloud through an Occopus-compatible interface (e.g EC2, Nova, OCCI, etc.)
    - 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 a base 14.04 Ubuntu OS image with cloud-init support
    - This image file can be found under Compute/Images/Public/Ubuntu 14.04 LTS image
- generated ssh key-pair (or for testing purposes one is attached)
    - We can create a new key pair by clicking on Compute/Access & Security/Key Pairs/Create Key Pair menu

Steps

Installation of Occopus

The deployment of the Apache Hadoop 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 Hadoop 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.hadoop-cluster. Install descriptors on a virtual machine which runs Occopus.

Note: In this tutorial, we will use nova cloud resources (based on our nova tutorials in the basic tutorial section). 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 same resource plugin presented in the tutorial
- 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.

Important: Do not modify the values of the contextualisation and the health_check section's attributes!

Important: Do not specify the server_name attribute for slaves so they are named automatically by Occopus to make sure node names are unique!

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 22`
- `TCP 8025`
- `TCP 8042`
- `TCP 8088`
- `TCP 8300-8600`
- `TCP 9000`
- `TCP 50000-51000`

3. Authentication

Make sure your authentication information is set correctly in your authentication file. Occopus requires the username / password pair used in MTA Cloud to authenticate itself and to be able to create virtual machines / infrastructures under a particular project. You must set your authentication data for the resource you would like to use. Setting authentication information is described here.

4. Set the size of the cluster

Update the number of Hadoop Slave worker nodes if necessary. For this, edit the infra-occopus-hadoop.yaml file and modifiy the min and max parameter under the scaling keyword. Scaling is the interval in which the number of nodes can change (min, max). Currently, the minimum is set to 2 (which will be the initial number at startup), and the maximum is set to 10. In the infrastructure descriptor (infra-hadoop-cluster.yaml) we can personalize the minimum and maximum number of Hadoop worker nodes in the cluster.

```
- &S
   name: hadoop-slave
   type: hadoop_slave_node
   scaling:
           min: 2
           max: 10
```

Important: Keep in mind that Occopus has to start at least one node from each node type to work properly and scaling can be applied only for Hadoop Slave nodes in this example!

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
```

Note: By editing the cloud init file in the nodes directory, advanced users can customize the Hadoop configuration files (cloud_init_hadoop_master.yaml, cloud_init_hadoop_slave.yaml).

Important: Occopus takes node definitions from its database when builds up the infrastructure, so importing is necessary whenever the node definition or any imported (e.g. contextualisation) file changes!

6. Deployment

Start deploying the infrastructure.

```
occopus-build infra-hadoop-cluster.yaml
```

7. The established infrastructure

After successful finish, the nodes 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:
hadoop-master:
    192.168.xxx.xxx (3116eaf5-89e7-405f-ab94-9550ba1d0a7c)
hadoop-slave:
    192.168.xxx.xxx (23f13bd1-25e7-30a1-c1b4-39c3da15a456)
    192.168.xxx.xxx (7b387348-b3a3-5556-83c3-26c43d498f39)

14032858-d628-40a2-b611-71381bd463fa
```

8. Check the cluster

You can check the health and statistics of the cluster through the following web pages:

```
Health of nodes: "http://<HadoopMasterIP>:50070"
Job statistics: "http://<HadoopMasterIP>:8088"
```

9. Running MapReduce applications

To launch a Hadoop MapReduce job copy your input and executable files to the Hadoop Master node, and perform the submission described here . To login to the Hadoop Master node use the private key attached to the tutorial package:

```
ssh -i builtin_hadoop_private_key hduser@[HadoopMaster ip]
```

10. Delete the infrastructure

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

```
occopus-destroy -i 14032858-d628-40a2-b611-71381bd463fa
```

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.