

DEEP PaaS Orchestrator Test Report

Issue: 2.0
Date: 8 November 2019

Table of Contents

TABLE OF CONTENTS	1
INTRODUCTION.....	2
SUMMARY	2
<i>PaaS Orchestrator Test Report Summary</i>	2
<i>Functional/Integration Tests:</i>	3
VERIFICATION SHEETS	4
<i>QCG-TP-01</i>	4
<i>SEC-TP-01</i>	10
<i>HTTPS-TP-01</i>	15
<i>CONF-TP-01</i>	20
<i>GPU-TP-03</i>	21
ANNEX 1: TESTBED DESCRIPTION	24
ANNEX 2: TOSCA TEMPLATES	27
<i>TEMPL01.yml</i>	27
<i>TEMPL02.yml</i>	28
<i>TEMPL03.yml</i>	32
<i>TEMPL04.yml</i>	37
ANNEX 3: SHELL SCRIPT	39
<i>test-script.sh</i>	39

Introduction

This document contains the results of the tests carried out on the DEEP-2 release of the Paas Orchestrator (version **2.2.0-FINAL**). The tests are aimed at verifying the functionalities and fixing foreseen for this release of the software, in particular they refer to the test cases described in the Test Plan document.

The described tests have been conducted on the development testbed described in

For completeness sake, the versions of the services used in the testbed are reported in the Annex 1.

Annex 2 contains the TOSCA templates used in the tests while Annex3 contains the shell script used in some tests.

Note that, in order to interact with the Orchestrator REST APIs, the orchent CLI version has been used. The CLI has been configured with oidc-agent version 3.2.6.

Summary

PaaS Orchestrator Test Report Summary

- Product: PaaS Orchestrator
- Product Card:
- VCS Tag: v.2.2.0-FINAL
- DEEP Major Release: DEEP-2 (Rosetta)
- Author: Marica Antonacci, Michele Perniola (INFN)
- Date: 6-7 November 2019
- Test Report Template: v. 2.0

Functional/Integration Tests:

Test Procedure ID	Test Case ID	Status (PASSED/FAILED)
QCG-TP-01	QCG-TC-01: Deployment of a QCG job	PASSED
SEC-TP-01	SEC-TC-01: Deployment of a Marathon job with SECRETS	PASSED
HTTPS-TP-01	HTTPS-TC-01: Enable HTTPS in Marathon applications	PASSED
CONF-TP-01	CONF-TC-01: Query Orchestrator configuration endpoint	PASSED
GPU-TP-03	GPU-TC-03: Schedule VM deployment requesting GPU	PASSED

Verification sheets

QCG-TP-01

Component: PaaS Orchestrator		Release:	Procedure ID: GPU-TP-01
Step #	Description	Status	Notes (including JIRA Tickets)
1.	Take note of the date at the bottom of this form	Done	
2.	<p>Launch the deployment:</p> <pre>#orchent depcreate TEMPL01.yml '{ "output_url": "http://cloud.recas.ba.infn.it:8080/v1/AUTH_f41187320a50 4846b132582e172fa268/testcontainer/deep- qcg.out?temp_url_sig=5c163f3c9a639f4ae819db42ca139417a21 dcae8%26temp_url_expires=1569669298"}'</pre> <p>Verify that the deployment request is accepted and a deployment UUID is returned.</p> <p>Take note of the deployment UUID.</p>	Passed	<p>See output #1 below the sheet.</p> <p>Deployment ID: 11e9e437-8af1-b041-b6c2-fa163e7a2099</p>
3.	<p>Monitor the status of the deployment with the command:</p> <pre>#watch -n 10 orchent depshow <DEP_UUID></pre>	Done	See output #2 below the sheet

4.	Verify that the final status of the deployment is CREATE_COMPLETE	Passed	
5.	Get the output from the job: #curl "http://cloud.recas.ba.infn.it:8080/v1/AUTH_f41187320a504846b132582e172fa268/testcontainer/deep-qcg.out?temp_url_sig=080692f2611280c8a264f37a0469554a7e2cdb04&temp_url_expires=1570194908" > deep-qcg.out	Done	See output #3 below the sheet
6.	Verify that the output file is accessible and that it is not empty	Passed	See output #4 below the sheet
7.	Verify the presence of the passed environment variable into the output file: #grep SOME_VAR deep-qcg.out	Passed	See output #5 below the sheet
8.	Remove test deployment: #orchent depdel <DEP_UUID>	Done	See output #6 below the sheet
Date:	11/07/2019	Test Conductor:	Michele Perniola Marica Antonacci

Output #1:

```
michele@mp-recas:~$ orchent decreate TEMPL01.yml '{ "output_url":  
"http://cloud.recas.ba.infn.it:8080/v1/AUTH_f41187320a504846b132582e172fa268/testcontainer/deep-  
qcg.out?temp_url_sig=5c163f3c9a639f4ae819db42ca139417a21dcae8%26temp_url_expires=1569669298"}'  
Deployment [11e9e44c-3ff3-7da1-9485-9625b6e1f581]:  
  status: CREATE_IN_PROGRESS  
  creation time: 2019-11-07T13:06+0000  
  update time: 2019-11-07T13:06+0000  
  callback:  
  status reason:  
  task: NONE  
  CloudProviderName:  
  outputs:  
  {}  
  links:  
    self [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11e9e44c-3ff3-7da1-9485-  
9625b6e1f581]  
    resources [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11e9e44c-3ff3-7da1-9485-  
9625b6e1f581/resources]  
    template [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11e9e44c-3ff3-7da1-9485-  
9625b6e1f581/template]
```

Output #2:

```
Every 10.0s: michele@mp-recas:~$ orchent depshow 11e9e44c-3ff3-7da1-9485-9625b6e1f581
Deployment [11e9e44c-3ff3-7da1-9485-9625b6e1f581]:
  status: CREATE_COMPLETE
  creation time: 2019-11-07T13:06+0000
  update time: 2019-11-07T13:06+0000
  callback:
  status reason:
  task: NONE
  CloudProviderName: PSNC
  outputs:
  {}
  links:
    self [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11e9e44c-3ff3-7da1-9485-9625b6e1f581]
    resources [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11e9e44c-3ff3-7da1-9485-9625b6e1f581/resources]
    template [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11e9e44c-3ff3-7da1-9485-9625b6e1f581/template]
```

Output #3:

```
curl "http://cloud.recas.ba.infn.it:8080/v1/AUTH_f41187320a504846b132582e172fa268/testcontainer/deep-qcg.out?temp_url_sig=080692f2611280c8a264f37a0469554a7e2cdb04&temp_url_expires=1570194908" > deep-qcg.out
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current				
			Dload Upload	Total	Spent	Left	Speed				
100	1476	100	1476	0	0	2007	0	--:--:--	--:--:--	--:--:--	2005

Output #4:

```
michele@mp-recas:~$ ls -l -s deep-qcg.out  
4 -rw-rw-r-- 1 michele michele 1476 nov  7 15:19 deep-qcg.out
```

Output #5:

```
michele@mp-recas:~$ grep SOME_VAR deep-qcg.out  
SOME_VAR=some_value
```

Output #6:

```
michele@mp-recas:~/$ orchent depdel 11e9e44c-3ff3-7da1-9485-9625b6e1f581  
deletion of deployment 11e9e44c-3ff3-7da1-9485-9625b6e1f581 successfully triggered
```

SEC-TP-01

Component: PaaS Orchestrator		Release:	Procedure ID: SEC-TP-01
Step #	Description	Status	Notes (including JIRA Tickets)
1.	Take note of the date at the bottom of this form.	Done	
2.	<p>Launch the deployment:</p> <pre>#orchent depcreate TEMPL02.yml '{ "service_password": "MyPassword"}'</pre> <p>Verify that the deployment request is accepted and a deployment UUID is returned.</p> <p>Take note of the deployment UUID.</p>	Passed	<p>See output #1 below the sheet.</p> <p>Deployment ID: 11e9e361-6c84-dbd0-b6c2-fa163e7a2099</p>
3.	<p>Monitor the status of the deployment with the command:</p> <pre>#watch -n 10 orchent depshow <DEP_UUID></pre>	Done	
4.	Verify that the final status of the deployment is CREATE_COMPLETE.	Passed	See output #2 below the sheet
5.	<p>Grab the endpoint of the service from retrieved status info:</p> <pre>--- outputs: { "endpoint": "xxx.xxx.xxx.xxx:nnnnn"</pre>	Done	endpoint: 90.147.170.127:10000

	<pre> } ---</pre>		
6.	<p>Connect to MySQL server using a client:</p> <pre>#mysql -hxxx.xxx.xxx.xxx -Pnnnnn -uroot -pMyPassword</pre>	Done	See output #3 below the sheet
7.	<p>Execute MySQL command:</p> <pre>mysql>show databases;</pre>	Done	
8.	<p>Verify output.</p>	Passed	See output #4 below the sheet
9.	<p>Quit MySQL client.</p> <pre>mysql>quit</pre>	Done	
10.	<p>Remove test deployment:</p> <pre>#orchent depdel <DEP_UUID></pre>	Done	See output #5 below the sheet
Date:	11/07/2019	Test Conductor:	Michele Perniola, Marica Antonacci

Output #1:

```
michele@mp-recas:~$ orchent decreate TEMPL02.yml '{ "service_password": "MyPassword"}'  
Deployment [11e9e361-6c84-dbd0-b6c2-fa163e7a2099]:  
  status: CREATE_IN_PROGRESS  
  creation time: 2019-11-07T09:05+0000  
  update time: 2019-11-07T09:05+0000  
  callback:  
  status reason:  
  task: NONE  
  CloudProviderName:  
  outputs:  
  {}  
  links:  
    self [https://deep-paas.cloud.ba.infn.it/orchestrator-03/deployments/11e9e361-6c84-dbd0-b6c2-fa163e7a2099]  
    resources [https://deep-paas.cloud.ba.infn.it/orchestrator-03/deployments/11e9e361-6c84-dbd0-b6c2-fa163e7a2099/resources]  
    template [https://deep-paas.cloud.ba.infn.it/orchestrator-03/deployments/11e9e361-6c84-dbd0-b6c2-fa163e7a2099/template]
```

Output #2:

```
Every 10.0s: michele@mp-recas:~$orchent depshow 11e9e361-6c84-dbd0-b6c2-fa163e7a2099
```

```
Deployment [11e9e361-6c84-dbd0-b6c2-fa163e7a2099]:
```

```
  status: CREATE_COMPLETE
```

```
  creation time: 2019-11-07T09:05+0000
```

```
  update time: 2019-1-07T09:05+0000
```

```
  callback:
```

```
  status reason:
```

```
  task: NONE
```

```
  CloudProviderName: RECAS-BARI
```

```
  outputs:
```

```
  {  
    "endpoint": "90.147.170.127:10000"  
  }
```

```
  links:
```

```
    self [https://deep-paas.cloud.ba.infn.it/orchestrator-03/deployments/11e9e361-6c84-dbd0-b6c2-fa163e7a2099]
```

```
    resources [https://deep-paas.cloud.ba.infn.it/orchestrator-03/deployments/11e9e361-6c84-dbd0-b6c2-fa163e7a2099/resources]
```

```
    template [https://deep-paas.cloud.ba.infn.it/orchestrator-03/deployments/11e9e361-6c84-dbd0-b6c2-fa163e7a2099/template]
```

Output #3:

```
michele@mp-recas:~$mysql -h 90.147.170.127 -P 10000 -u root -p MyPassword
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.27 MySQL Community Server (GPL)
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

Output #4

```
+-----+
| Database          |
+-----+
| information_schema|
| mysql             |
| performance_schema|
| sys               |
+-----+
4 rows in set (0,10 sec)
```

Output #5

```
michele@mp-recas:~/$ orchent depdel 11e9e361-6c84-dbd0-b6c2-fa163e7a2099
deletion of deployment 11e9e361-6c84-dbd0-b6c2-fa163e7a2099 successfully triggered
```

HTTPS-TP-01

Component: PaaS Orchestrator		Release:	Procedure ID: HTTPS-TP-01
Step #	Description	Status	Notes (including JIRA Tickets)
1.	Take note of the date at the bottom of this form.	Done	
2.	<p>Launch the deployment:</p> <pre>#orchent deprecate TEMPL03.yml '{}'</pre> <p>Verify that the deployment request is accepted and a deployment UUID is returned.</p> <p>Take note of the deployment UUID.</p>	Passed	<p>See output #1 below the sheet.</p> <p>Deployment ID: 11ea008b-59cd-6b4e-b49f-7011248cddd5</p>
3.	<p>Monitor the status of the deployment with the command:</p> <pre>#watch -n 10 orchent depshow 11ea008b-59cd-6b4e-b49f-7011248cddd5</pre>	Done	
4.	Verify that the final status of the deployment is CREATE_COMPLETE.	Passed	See output #2 below the sheet
5.	<p>Grab the endpoint of the service from retrieved status info:</p> <pre>---</pre> <p>outputs:</p> <pre>{ "deepaas_endpoint": "mesos-lb.recas.ba.infn.it:10000",</pre>	Done	endpoint: mesos-lb.recas.ba.infn.it:10000

	<pre>"jupyter_endpoint": "mesos-lb.recas.ba.infn.it:10002", "monitor_endpoint": "mesos-lb.recas.ba.infn.it:10001" } ---</pre>		
6.	<p>Using a browser, connect to the newly created server using the HTTPS protocol:</p> <p><code>https://mesos-lb.recas.ba.infn.it:10000</code></p>	Done	
7.	Verify output.	Passed	See output #3 below the sheet. The endpoint is reachable, but it seems that the DEEPaaS endpoint is not working as expected. This has been notified to the DEEPaaS team.
8.	<p>Remove test deployment:</p> <pre>#orchent depdel 11ea008b-59cd-6b4e-b49f-7011248cddd5</pre>	Done	See output #4 below the sheet
Date:	11/06/2019	Test Conductor:	Michele Perniola, Marica Antonacci

Output #1:

```
michele@mp-recas:~$ orchent depcreate TEMPL03.yml '{}'  
Deployment [11ea008b-59cd-6b4e-b49f-7011248cddd5]:  
  status: CREATE_IN_PROGRESS  
  creation time: 2019-11-06T11:48+0000  
  update time: 2019-11-06T11:48+0000  
  callback:  
  status reason:  
  task: NONE  
  CloudProviderName:  
  outputs:  
  {}  
  links:  
    self [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea008b-59cd-6b4e-b49f-7011248cddd5]  
    resources [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea008b-59cd-6b4e-b49f-7011248cddd5/resources]  
    template [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea008b-59cd-6b4e-b49f-7011248cddd5/template]
```

Output #2:

```
Every 10.0s: michele@mp-recas:~$ orchent depshow 11ea008b-59cd-6b4e-b49f-7011248cddd5
Deployment [11ea008b-59cd-6b4e-b49f-7011248cddd5]:
  status: CREATE_COMPLETE
  creation time: 2019-11-06T11:48+0000
  update time: 2019-11-06T11:50+0000
  callback:
  status reason:
  task: NONE
  CloudProviderName: RECAS-BARI
  outputs:
  {
    "deepaas_endpoint": "mesos-lb.recas.ba.infn.it:10000",
    "jupyter_endpoint": "mesos-lb.recas.ba.infn.it:10002",
    "monitor_endpoint": "mesos-lb.recas.ba.infn.it:10001"
  }
  links:
  self [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea008b-59cd-6b4e-b49f-7011248cddd5]
  resources [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea008b-59cd-6b4e-b49f-7011248cddd5/resources]
  template [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea008b-59cd-6b4e-b49f-7011248cddd5/template]
```

Output #3

No API definition provided.

Output #4

```
michele@mp-recas:~/$ orchent depdel 11ea008b-59cd-6b4e-b49f-7011248cddd5  
deletion of deployment 11ea008b-59cd-6b4e-b49f-7011248cddd5 successfully triggered
```

CONF-TP-01

Component: PaaS Orchestrator		Release:	Procedure ID: CONF-TP-01
Step #	Description	Status	Notes (including JIRA Tickets)
1.	Take note of the date at the bottom of this form.	Done	
2.	Query the orchestrator configuration endpoint: # curl -H "Authorization: Bearer \$IAM_TOKEN" http://<orchestrator_url>/configuration	Done	See output #1 below the sheet.
3.	Verify output.	Passed	See output #2 below the sheet
Date:	11/06/2019	Test Conductor:	Michele Perniola, Marica Antonacci

Output #1:

```
michele@mp-recas:~$ curl -H "Authorization: Bearer *****" https://deep-paas-dev.cloud.ba.infn.it/orchestrator/configuration | jq '
```

Output #2:

```
{  
  "cpr_url": "https://deep-paas-dev.cloud.ba.infn.it/cpr",  
  "slam_url": "https://deep-slam.cloud.ba.infn.it:8443/rest/slam",  
  "cmdb_url": "https://deep-paas-dev.cloud.ba.infn.it/cmdb",  
  "im_url": "https://deep-paas-dev.cloud.ba.infn.it/im",  
  "monitoring_url": "https://deep-paas-dev.cloud.ba.infn.it/monitoring-wrapper",  
  "vault_url": "https://vault.cloud.ba.infn.it:8200"  
}
```

GPU-TP-03

Component: PaaS Orchestrator		Release:	Procedure ID: GPU-TP-03
Step #	Description	Status	Notes (including JIRA Tickets)
1.	Take note of the date at the bottom of this form.	Done	
2.	Launch the deployment: #orchent depcreate TEMPL04.yml '{}'	Passed	See output #1 below the sheet. Deployment ID: 11ea20f7-14ce-cf1d-9f66-024250803cfb

	Verify that the deployment request is accepted and a deployment UUID is returned. Take note of the deployment UUID.		
3.	Monitor the status of the deployment with the command: <code>#watch -n 10 orchent depshow <DEP_UUID></code> Verify that the final status of the deployment is CREATE_COMPLETE	Passed	The Orchestrator has selected IFCA site since it's the only site providing GPUs in the development testbed.
4.	Access the deployed VM and verify the presence of the GPU(s): <code># lspci grep NVIDIA</code>	Passed	<code># lspci grep NVIDIA</code> 00:06.0 VGA compatible controller: NVIDIA Corporation Device 1b06 (rev a1)
5.	Remove test deployment: <code>#orchent depdel <DEP UUID></code>	Done	
Date:	11/06/2019	Test Conductor:	Michele Perniola, Marica Antonacci

```
# orchent deprecate TEMPL04.yaml '{}'
Deployment [11ea20f9-f8b6-b41e-9f66-024250803cfb]:
status: CREATE_IN_PROGRESS
creation time: 2019-11-06T18:20+0000
update time: 2019-11-06T18:21+0000
callback:
status reason:
task: NONE
CloudProviderName:
outputs:
{}
```

```
links:
  self [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea20f9-f8b6-b41e-9f66-024250803cfb]
  resources [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea20f9-f8b6-b41e-9f66-024250803cfb/resources]
  template [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea20f9-f8b6-b41e-9f66-024250803cfb/template]
```

```
# orchent depshow 11ea20f9-f8b6-b41e-9f66-024250803cfb
```

```
Deployment [11ea20f9-f8b6-b41e-9f66-024250803cfb]:
```

```
status: CREATE_COMPLETE
creation time: 2019-11-06T18:20+0000
update time: 2019-11-06T18:34+0000
callback:
status reason:
task: NONE
CloudProviderName: IFCA-LCG2
outputs:
{"node_creds": *****, "node_ip": 193.146.75.194}}
links:
  self [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea20f9-f8b6-b41e-9f66-024250803cfb]
  resources [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea20f9-f8b6-b41e-9f66-024250803cfb/resources]
  template [https://deep-paas-dev.cloud.ba.infn.it/orchestrator/deployments/11ea20f9-f8b6-b41e-9f66-024250803cfb/template]
```

Annex 1: Testbed description

Type of resource	Partner	Endpoint of service/resource	Comments/status
PaaS/Orchestrator			tosca custom types v4.0.0

PaaS/CMDB	INFN-BA		Registered sites: RECAS-BARI, IFCA-LCG2
PaaS/SLAM			
PaaS/Monitoring		https://deep-paas-dev.cloud.ba.infn.it/monitoring-wrapper	<ul style="list-style-type: none"> ● zabbix/zabbix-web-apache-mysql:alpine-3.2-latest ● zabbix/zabbix-server-mysql:alpine-3.2-latest ● mysql:5.7 Zabbix agent docker images: <ul style="list-style-type: none"> ● zabbix/zabbix-agent:alpine-3.2-latest Zabbix Wrapper docker image: <ul style="list-style-type: none"> ● indigodatacloud/zabbix-wrapper:indigo_2
PaaS/CPR		https://deep-paas-dev.cloud.ba.infn.it/cpr	v0.7.0

Infrastructure Manager			
Mesos (CPU-only)	INFN-BA	(REST API) (REST API) (Web UI) (Web UI)	<i>Cluster LB: mesos-lb.cloud.ba.infn.it</i> <i>Mesos version: 1.9.0</i> <i>Marathon version: 1.5.6</i> <i>Chronos version: 3.0.2</i>
Mesos (with GPUs)	INFN-BA	(REST API) (REST API)	Cluster LB: mesos-lb.recas.ba.infn.it Mesos version: 1.9.0 Marathon version: 1.5.6 Chronos version: 3.0.2 (patched for GPU)
Onedata	INFN-BA		
Qcg	PSNC	http://qcg.lambda.ara.app.test.k8s.apps.psnc.pl/api	
Vault	INFN-BA	https://vault.cloud.ba.infn.it:8200	1.1.2

Annex 2: TOSCA Templates

TEMPL01.yml

```
tosca_definitions_version: tosca_simple_yaml_1_0
```

```
imports:
```

```
- indigo_custom_types: https://raw.githubusercontent.com/indigo-dc/tosca-types/v4.0.0/custom_types.yaml
```

```
description: TOSCA example for submitting a job to a QCG
```

```
topology_template:
```

```
  inputs:
```

```
    output_url:
```

```
      type: string
```

```
      required: yes
```

```
  node_templates:
```

```
    qcg_job:
```

```
      type: tosca.nodes.indigo.Qcg.Job
```

```
      properties:
```

```
        executable: "curl -s
```

```
https://gist.githubusercontent.com/mperniola/b06de5d7e7c6cfaab72b2257c14d2ab4/raw/eaee1991832abcb9286ad95cc34e8eb1e4cc38f3/test-script.sh | /bin/bash"
```

```
        environment:
```

```
          OUTPUT_URL: { get_input: output_url }
```

```
          SOME_VAR: some_value
```

TEMPL02.yml

```
tosca_definitions_version: toska_simple_yaml_1_0

imports:
  - indigo_custom_types: https://raw.githubusercontent.com/indigo-dc/tosca-
types/v4.0.0/custom_types.yaml

description: >
  TOSCA template for specifying Marathon applications to enable the
  specification of long-running services in INDIGO using SECRETS.

topology_template:

  inputs:

    cpus:
      type: float
      description: Number of CPUs for this service
      required: yes
      default: 1.0

    mem:
      type: scalar-unit.size
      description: Amount of Memory for this service
      required: yes
      default: 1 GB

    docker_image:
      type: string
      description: Docker image to be used to run the container application
      required: yes
      default: "mysql:5.7"
```

```
port:
  type: integer
  description: service port (exposed by the docker container)
  required: yes
  default: 3306

service_password:
  type: string
  description: password for access
  required: yes

node_templates:

marathon:
  type: toska.nodes.indigo.Container.Application.Docker.Marathon
  properties:
    # environment_variables:
    uris: []
    labels:
      HAProxy_GROUP: external
    secrets:
      MYSQL_ROOT_PASSWORD: { get_input: service_password }
  artifacts:
    image:
      file: { get_input: docker_image }
      type: toska.artifacts.Deployment.Image.Container.Docker
  requirements:
    - host: docker_runtime

docker_runtime:
```

```
type: toscanodes.indigo.Container.Runtime.Docker
capabilities:
  host:
    properties:
      num_cpus: { get_input: cpus }
      mem_size: { get_input: mem }
      publish_ports:
        - protocol: tcp
          source: { get_input: port }

outputs:
  endpoint:
    value: { concat: [ { get_attribute : [ marathon, load_balancer_ips, 0 ] }, ':', { get_attribute :
[ docker_runtime, host, publish_ports, 0, target ] } ] }
```

TEMPL03.yml

```
tosca_definitions_version: "tosca_simple_yaml_1_0"
imports:
  - indigo_custom_types: "https://raw.githubusercontent.com/indigo-dc/tosca-types/v4.0.0/custom_types.yaml"

topology_template:
  inputs:

    docker_image:
      type: string
      description: 'deephdc/deep-oc-mods'
      required: yes
      default: "deephdc/deep-oc-mods"

    mem_size:
      type: string
      description: Amount of memory
      required: no
      default: "4096 MB"

    num_cpus:
      type: integer
      description: Number of required CPUs
      required: no
      default: 1

    num_gpus:
      type: integer
      description: Number of required GPUs
      required: no
      default: 0
```



```
flaat_disable:
  type: string
  description: disable flaat authentication
  required: no
  default: "no"

rclone_conf:
  type: string
  description: rclone.conf location
  required: no
  default: "/srv/.rclone.conf"

rclone_url:
  type: string
  description: remote storage link to access via webdav
  required: no
  default: "https://nc.deep-hybrid-datacloud.eu/remote.php/webdav/"

rclone_vendor:
  type: string
  description: rclone vendor
  required: no
  default: "nextcloud"

rclone_user:
  type: string
  description: rclone user to access remote storage
  required: no
  default: ""

rclone_password:
  type: string
  description: rclone user password
```

```
    required: no
    default: ""

jupyter_password:
  type: string
  description: jupyter password
  required: no
  default: "s3cret"

jupyter_config_url:
  type: string
  description: url to download some jupyter config
  required: no
  default: ""

run_command:
  type: string
  description: default command to run
  required: yes
  default: "deepaas-run --listen-ip=0.0.0.0"

node_templates:
  Docker:
    type: "tosca.nodes.indigo.Container.Runtime.Docker"
    capabilities:
      host:
        properties:
          publish_all: false
          publish_ports:
            - protocol: "tcp"
              source: 5000
            - protocol: "tcp"
              source: 6006
```

```

    - protocol: "tcp"
      source: 8888
      mem_size: { get_input: mem_size }
      num_cpus: { get_input: num_cpus }
      num_gpus: { get_input: num_gpus }
scalable:
  properties:
    min_instances: 1
    max_instances: 1
    default_instances: 1
marathon:
  type: "tosca.nodes.indigo.Container.Application.Docker.Marathon"
  properties:
    force_pull_image: true
    command: { get_input: run_command }
    enable_https: true
  environment_variables:
    RCLONE_CONFIG: { get_input: rclone_conf }
    RCLONE_CONFIG_DEEPNC_TYPE: webdav
    RCLONE_CONFIG_DEEPNC_URL: { get_input: rclone_url }
    RCLONE_CONFIG_DEEPNC_VENDOR: { get_input: rclone_vendor }
    RCLONE_CONFIG_DEEPNC_USER: { get_input: rclone_user }
    RCLONE_CONFIG_DEEPNC_PASS: { get_input: rclone_password }
    DISABLE_AUTHENTICATION_AND_ASSUME_AUTHENTICATED_USER: {get_input: flaat_disable}
    jupyterPASSWORD: {get_input: jupyter_password}
    jupyterCONFIG_URL: {get_input: jupyter_config_url}
  labels:
    dummy: "mylabel"
  privileged: false
  requirements:
    - host:
      node: "Docker"
      capability: "tosca.capabilities.indigo.Container.Docker"

```

```
    relationship: "tosca.relationships.HostedOn"
artifacts:
  image:
    file: { get_input: docker_image }
    type: "tosca.artifacts.Deployment.Image.Container.Docker"
outputs:
  deepaas_endpoint:
    value: { concat: [ { get_attribute : [ marathon, load_balancer_ips, 0 ] }, ':', { get_attribute : [
  Docker, host, publish_ports, 0, target ] } ] }
  monitor_endpoint:
    value: { concat: [ { get_attribute : [ marathon, load_balancer_ips, 0 ] }, ':', { get_attribute : [
  Docker, host, publish_ports, 1, target ] } ] }
  jupyter_endpoint:
    value: { concat: [ { get_attribute : [ marathon, load_balancer_ips, 0 ] }, ':', { get_attribute : [
  Docker, host, publish_ports, 2, target ] } ] }
```

TEMPL04.yml

```
tosca_definitions_version: tosca_simple_yaml_1_0

imports:
  - indigo_custom_types: https://raw.githubusercontent.com/indigo-dc/tosca-types/v4.0.0/custom\_types.yaml

description: >
  Launch a compute node getting the IP and SSH credentials to access via ssh

metadata:
  display_name: Start a virtual machine
  icon: https://indigo-paas.cloud.ba.infn.it/public/images/vm-icon.png

topology_template:

  inputs:
    num_cpus:
      type: integer
      description: Number of virtual cpus for the VM
      default: 1
      constraints:
        - valid_values: [ 1, 2, 4 ]
    num_gpus:
      type: integer
      description: Number of virtual gpus for the VM
      default: 1
      constraints:
        - valid_values: [ 1, 2 ]
    mem_size:
      type: scalar-unit.size
      description: Amount of memory for the VM
      default: 2 GB
      constraints:
```

```
- valid_values: [ 2 GB, 4 GB ]

node_templates:

  simple_node:
    type: toska.nodes.indigo.Compute
    capabilities:
      endpoint:
        properties:
          network_name: PUBLIC
    scalable:
      properties:
        count: 1
    host:
      properties:
        #instance_type: m1.small
        num_cpus: { get_input: num_cpus }
        num_gpus: { get_input: num_gpus }
        mem_size: { get_input: mem_size }
    os:
      properties:
        distribution: ubuntu
        type: linux

outputs:
  node_ip:
    value: { get_attribute: [ simple_node, public_address, 0 ] }
  node_creds:
    value: { get_attribute: [ simple_node, endpoint, credential, 0 ] }
```

ANNEX 3: Shell script

test-script.sh

```
#!/bin/bash

date >> output.txt
printenv >> output.txt
URL=$(echo $OUTPUT_URL | sed 's/\%26/\&/g')
curl -i -XPUT -T output.txt $URL
```