Horizon 2020 Program (2014-2020)

Big data PPP

Research addressing main technology challenges of the data economy

Industrial-Driven Big Data as a Self-Service Solution

D5.5: Federated Resource Management for Data Analytics v3[†]

Abstract: The content of this report is mainly focused on the setup of the infrastructure layer, which includes the selected underlying storage and processing infrastructure of the I-BiDaaS solution. The document also describes the preliminary work carried out on the distributed large-scale layer, which is responsible for the orchestration and management of the underlying physical computational and storage infrastructure.

Contractual Date of Delivery	31/12/2020
Actual Date of Delivery	30/12/2020
Deliverable Security Class	Public
Editor	*Enric Pages (ATOS)*
Contributors	ATOS, BSC, UNSPMF, ITML, FORTH
Quality Assurance	*Giorgos Vasiliadis (FORTH),* *George Bravos (ITML)*

[†] *The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780787.*

The *I-BiDaaS* Consortium

Foundation for Research and Technology – Hellas (FORTH)	Coordinator	Greece
Barcelona Supercomputing Center (BSC)	Principal Contractor	Spain
IBM Israel – Science and Technology LTD (IBM)	Principal Contractor	Israel
Centro Ricerche FIAT (FCA/CRF)	Principal Contractor	Italy
Software AG (SAG)	Principal Contractor	Germany
Caixabank S.A. (CAIXA)	Principal Contractor	Spain
University of Manchester (UNIMAN)	Principal Contractor	United Kingdom
Ecole Nationale des Ponts et Chaussees (ENPC)	Principal Contractor	France
ATOS Spain S.A. (ATOS)	Principal Contractor	Spain
Aegis IT Research LTD (AEGIS)	Principal Contractor	United Kingdom
Information Technology for Market Leadership (ITML)	Principal Contractor	Greece
University of Novi Sad Faculty of Sciences (UNSPMF)	Principal Contractor	Serbia
Telefonica Investigation y Desarrollo S.A. (TID)	Principal Contractor	Spain

Document Revisions & Quality Assurance

Internal Reviewers

1. *Giorgos Vasiliadis (FORTH)*
2. *George Bravos (ITML)*

Revisions

Version	Date	By	Overview
0.1	10/11/2020	ATOS	Initial ToC
0.2	15/11/2020	ALL	ToC refinement
0.4	22/11/2020	ITML	Integration process inputs
0.5	28/11/2020	UNSPMF	Runtime environment inputs
0.6	28/11/2020	BSC	Runtime environment inputs
0.7	10/12/2019	ATOS	RMO contributions
0.8	15/12/2019	ATOS	Contributions merge
0.9	19/12/2019	Internal Reviewers	Document for review
1.0	28/12/2019	ATOS	Final document

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API: Application Programming Interface

CI: Continuous Integration

CPU: Central Processing Unit

CSP: Cloud Service Provider

GPGPUs: General Purpose Graphics Processing Unit

GPU: Graphics Processing Unit

HPC: High Performance Computing

IT: Information Technologies

ITER: Instituto Tecnológico de Energías Renovables

KVM: Kernel-based Virtual Machine

MVP: Minimum Viable Product

OASIS: Organization for the Advancement of Structured Information Standards

PoC: Proof of Concept

RMO: Resource Management and Orchestration

SQL: Structured Query Language

TOSCA: Topology and Orchestration Specification for Cloud Applications

VM: Virtual Machine

WP: Work Package

Executive Summary

The goal of I-BiDaaS project is to create new opportunities for self-service analytics by offering a unified Big Data as-a-service solution that will empower non-expert users to easily take advantage of the Big-Data technologies while at the same time increase the speed of data analytics. One of the key aspects to achieve this vision is the capability of the platform to handle large data volumes.

This report summarizes the work carried out in the scope of “Task5.1 Provisioning and configuration of infrastructure resources” and “Task5.2 Resource Management and optimized automatic usage of computational and storage resources”. The document presents the third version of the Federated Resource Management for Data Analytics Report which covers the period from M25 to M36. The work described covers the re-configuration and maintenance processes followed to setup the infrastructures required to support the Final version of I-BiDaaS. In addition, the document updates the design and specification of the Resource Management and Orchestration (RMO) software modules presented across D5.1 and D5.3, describing the environment set up for orchestrating resources on top of the I-BiDaaS infrastructure layer.

Finally, the document conclusions for the realisation of the final I-BiDaaS solution delivered at M36.

1 Introduction

1.1 Overview and Objectives

One of the project's main goals is to create new opportunities for self-service analytics towards a complete paradigm tailored for big data analytics. One of the key aspects to achieve this vision is the capability of the platform to handle large data volumes. As such, WP5 within I-BiDaaS aims to provide the distributed large-scale framework that permits powerful and scalable data processing on top of heterogenous and federated infrastructures.

This report has been focussed on the following WP objectives documented in the I-BiDaaS DoA:

- Management of diverse infrastructure resources
- Orchestration of computational resources across diverse resource providers
- Integration and exploitation of infrastructure elasticity capabilities.

To this end, I-BiDaaS offers a unified Big Data as-a-service solution that empowers non-expert Big-Data users to easily take advantage of the Big-Data technologies while at the same time, increase the speed of data analytics. This report updates iteratively the content described as part of "D5.1. Federated Resource Management for Data Analytics v1" [1] and "D5.3. Federated Resource Management for Data Analytics v2" [1].

1.2 Relation to other Tasks and Work Packages

The content of this report updates the work carried out in the scope of Tasks 5.1. and 5.2. The systems and software modules presented in this document have a direct relationship with the ones previously documented in earlier WP5 reports.

Due to the provisioning and because management of the computational resources plays an important role within the project solution, this report has links with other technical reports across WP2, WP3, and WP4 as well as with the experiments executed in the scope of WP6.

In order to get a better understanding of the content of this report, the suggested reading path is the following:

- "D.1.3 Positioning of I-BiDaaS" → which provides an overview of the industrial challenges of the data economy as well as sets the scene for the realisation of the I-BiDaaS platform.
- "D1.2 Architecture definition" → which describes the specification of the I-BiDaaS architecture and its software modules.
- "D5.1 Federated Resource Management for Data Analytics v1" → where the operational infrastructure environment supporting the realisation of the I-BiDaaS MVP prototype has been defined.
- "D5.2 Big-Data-as-a-Self-Service Test and Integration Report" → which describes the testing and integration work carried out towards the MVP prototype (M12).
- "D5.3 Federated Resource Management for Data Analytics v2" → where the operational infrastructure environment supporting the realisation of the 1st version of the I-BiDaaS prototype has been defined.
- "D5.4 Big-Data-as-a-Self-Service Test and Integration Report v2" → which describes the testing and integration work carried out towards the 1st integrated prototype (M24).

1.3 Target Audiences

The primary target of the document is internal I-BiDaaS technicians (e.g., from WP2 and WP5) involved in the prototyping and implementation of the platform. Additionally, this document can be also valuable to external technical personnel that are willing to adopt the I-BiDaaS solution and/or Cloud Service Providers (CSPs) willing to be incorporated as infrastructure and data providers within our solution.

1.4 Structure of the Document

The outline of this document is as follows: The first section introduces the document and its objectives. The second section describes the re-configuration and maintenance processes followed within the project to setup the required infrastructure resources. The third section presents the work carried out in the scope of Task 5.2, where the Resource Management and Orchestration software module was implemented. The runtime environment providing the execution environment for data analytics is described in section 4. Finally, the last section contains the conclusions (Section 5).

2 Provision and Maintenance of Infrastructures Resources

2.1 Infrastructure Testbed

Atos' Infrastructure Services offered in I-BiDaaS are delivered on top of the hardware resources from *Instituto Tecnológico de Energías Renovables* (ITER) Data Centre located in the Canary Islands. The Teide-HPC supercomputer is composed of 1100 Fujitsu computer servers, the core of the compute nodes is Fujitsu PRIMERGY CX250 S1 servers housed in a PRIMERGY CX400 chassis grouped in 4 nodes per chassis and featured with the latest Intel Sandy Bridge processors.

FORTH's commodity cluster contains several modern off-the-shelf commodity GPGPUs, such as NVIDIA GeForce GTX 1080 Ti and NVIDIA TITAN Xp. Such GPGPUs offer extremely high processing throughput for parallelizable workloads with a low power cost.

Additionally, within the last phase of the project, proof of concept deployments using docker containers have been tested on top of edge devices (NVIDIA Jetson Nano, Bull Sequana Edge).

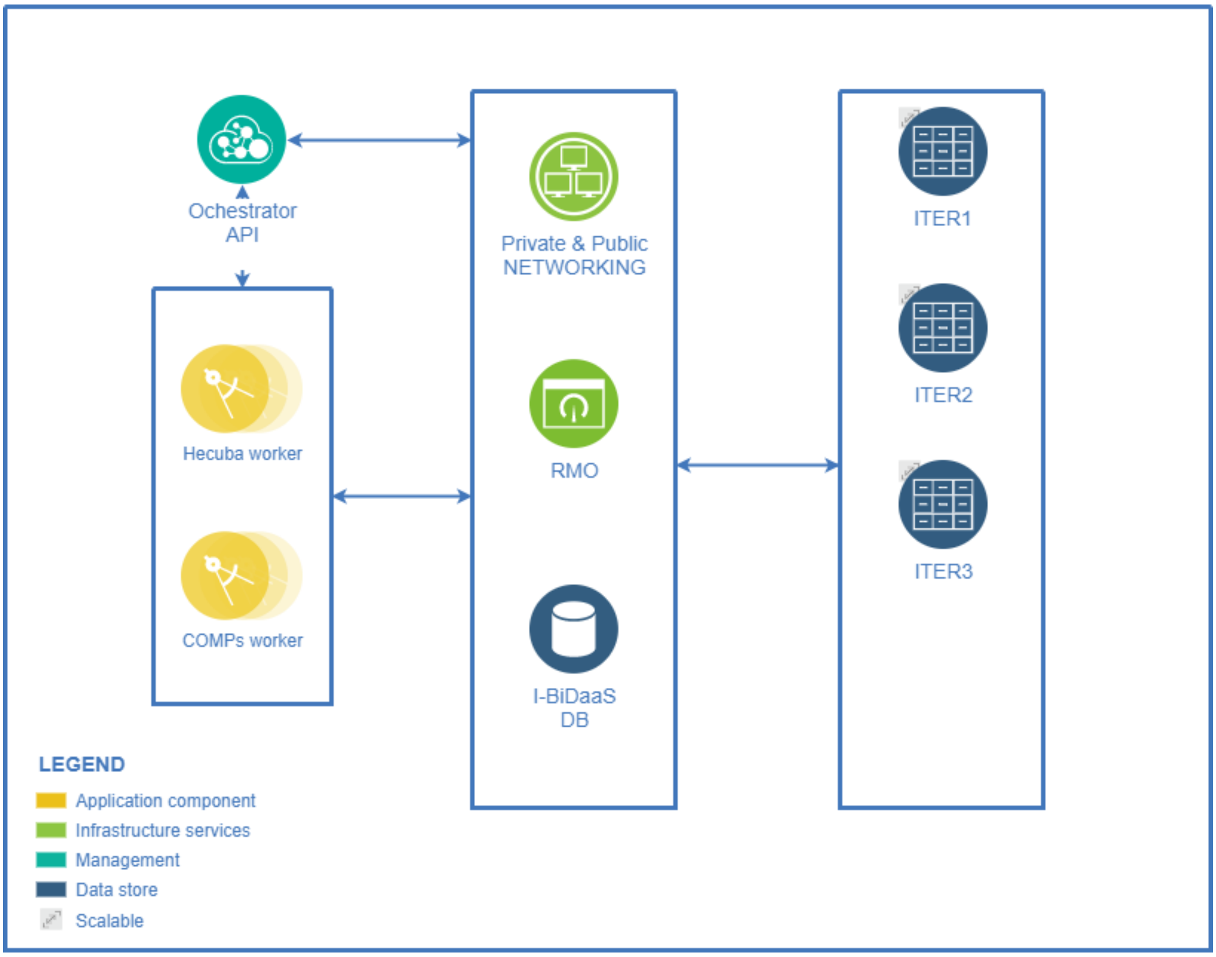


Figure 1. I-BiDaaS Cloud Infrastructure

2.2 Cloud Testbed Maintenance

Atos manages the Cloud computing environment, which was used during the development and validation phases of the project, in accordance to the following ISO standards:

- ISO 9001 (quality); specifies requirements for a quality management system when an organization:
  a) needs to demonstrate its ability to consistently provide products and services that meet customer and applicable statutory and regulatory requirements, and
  b) aims to enhance customer satisfaction through the effective application of the system, including processes for improving the system and the assurance of conformity to customer and applicable statutory and regulatory requirements.
- ISO14001 (environmental); which specifies the requirements for an environmental management system that an organization can use to enhance its environmental performance. ISO 14001:2015 is intended for use by an organization seeking to manage its environmental responsibilities in a systematic manner that contributes to the environmental pillar of sustainability.
- ISO 27001 (security) standards; is widely known for providing requirements for an information security management system (ISMS), though there are more than a dozen standards in the ISO/IEC 27000 family. Using them enables organizations of any kind to manage the security of assets such as financial information, intellectual property, employee details or information entrusted by third parties.

Atos' Infrastructure Services offered in I-BiDaaS are delivered on top of the hardware resources from Instituto Tecnológico de Energías Renovables (ITER) Data Centre located in the Canary Islands. The scope of this service comprises:

- Provision of bare-metal resources.
- Provision of server capacity on-demand.
- Provision of the storage capacity on-demand.
- Provision of network connection.
- Operation of the infrastructure.
- Online portal and/or API for service requests.

Other maintenance activities related to Task 5.1 are listed below:

- User access management.
- Resizing computational capacity of virtual resources.
- Resizing storage capacity of virtual resources. (with and without LVM support)
- Re-installation of a malfunctioning physical node.
- Re-configure the previous private Cloud Service Provider environment from all-in-one mode to a multi-node setup.
- Extend the previous computational capacity configuring one extra Nova compute node for our private cloud provider.
- Migrate from DevStack distribution to OpenStack Kolla based in containers.
- Fulfil the process requested by ITER security audit.

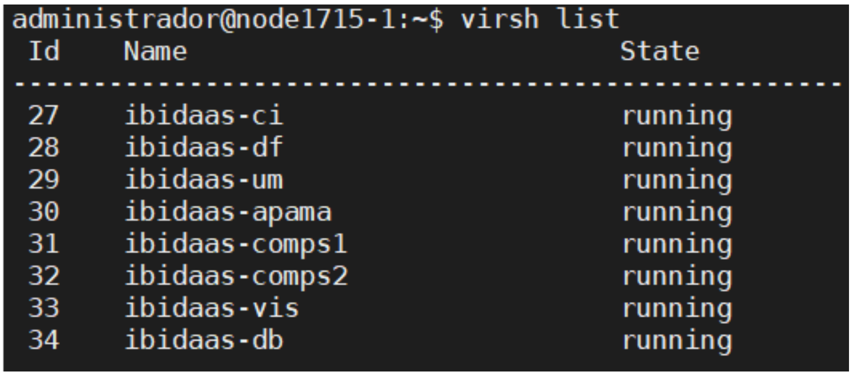
- Recover networking after the data center changes the telco operator during the project course.

For the realisation of the I-BiDaaS platform development and validation activities the following resource capacity has been provided during the project lifetime:

Table 1. Resource Capacity

	Total	**Per Node**
Computer Processing Capacity	48 cores (96 threads)	16 cores (32 threads)
Memory Capacity	192 GB	64 GB
Storage Capacity	36 TB	12 TB
Network Capacity	(2 x 1GbE) x 3	2 x 1GbE

The core modules of the platform are deployed across VMs deployed in one single node.



```
administrador@node1715-1:~$ virsh list
 Id    Name                           State
----------------------------------------------------
 27    ibidaas-ci                     running
 28    ibidaas-df                     running
 29    ibidaas-um                     running
 30    ibidaas-apama                  running
 31    ibidaas-comps1                 running
 32    ibidaas-comps2                 running
 33    ibidaas-vis                    running
 34    ibidaas-db                     running
```

Figure 2. I-BiDaaS Cloud Infrastructure on ITER1

2.3 Confidential Computing - Privacy Enhanced Execution

Applying privacy to cloud-environments is crucial and essential. I-BiDaaS provided some research results on establishing a secure environment for its users, using technologies such as the Intel SGX (Software Guard eXtensions). This technology manages to protect selected code and data from disclosure or modification and is available with any 7th Generation Intel Core processor platform and any Intel Xeon processor E3 v5 for data center servers. Intel SGX is an ideal solution for untrusted environments. Cloud environments can be considered untrusted, since the user has minor control over aspects like the storage of the data. SGX can guarantee privacy even with an untrusted service provider. Regarding data transfers from data sources, as long as the result reporting, privacy is also reassured through the SGX's remote attestation end-to-end technique.

Today, organizations and enterprises tend to outsource data processing workloads to cloud providers. Providing applications as-a-service has become a very convenient trend due to lower cost and maintenance complexity. Still, these workloads contain important information about the user or the organization. Processing sensitive information (such as user files, e-mails, logs, network traffic, etc.) needs to be taken seriously by complying to security and privacy preserving standards in order to guarantee confidentiality. Yet, it is quite challenging to provide strong guarantees regarding the safety of users' data, especially when handled by parties other than themselves. For instance, a public cloud environment is typically considered untrusted, since there is no control over the operating system, the hypervisor, the drivers, the management stack, the system's memory, I/O devices, etc. Furthermore, even in a fully clean environment, there is always the possibility of an honest-but-curious cloud provider, willing to learn or extract information regarding the users and their data.

Intel SGX is a hardware assisted mechanism in the form of an ISA extension to the Intel architecture. It is designed to allow secure attestation and sealing to application software executing in a secure environment that is known as enclave. The main purpose of these extensions is the protection of selected code parts and data from disclosure or modification in untrusted environments. The enclaves are protected by the CPU that is in charge of any access to the enclave memory or other protected areas of execution. Any instruction that reads or writes to the enclave and is not part of it, fails. Assuming an untrusted or even a malicious operating system, hypervisor or firmware, SGX protects the confidentiality of the enclave pages. An Intel SGX application consists of (i) the trusted code and (ii) a trusted enclave that it securely calls into.

We have already started experimenting with Intel SGX and how it can provide protection in cloud processing environments, such as I-BiDaaS. Our preliminary work has been published recently and can be found online [4] [5]. I-BiDaaS could utilize this approach to safeguard both end-users and the server from third-party cloud providers. In principle, we define three different entities: (i) the client, which transmits the necessary data to the remote server for processing, (ii) the server, which is responsible for performing the analysis in a privacy-preserving way, and (iii) the public cloud provider. The entire processing is performed in the cloud-based server, encapsulated inside Intel SGX enclaves, which communicates with the clients through a network connection. This encapsulation enables the protection of the data processing algorithms, and most importantly the privacy of the user's data. We assume that a client is installed and initially executed when the device is in a clean state, so no malicious executable has taken the control of the client or the device.

3 Resource Management and Optimized Usage of Resources

3.1 Introduction

I-BiDaaS solution is based on three main layers: the infrastructure layer, the distributed large-scale layer, and the application layer. The software module presented in this section belongs to the distributed large-scale layer. The Resource Management and Orchestration software module (RMO) is responsible for interfacing with the infrastructure layer allowing to programmatically interact with the set of virtual and physical resources that the application requires. The component aims to support multiple providers and various computational resource types.

After evaluating various technologies through the realization of Proof of Concepts, this report presents the technology enablers selected for the realization of the I-BiDaaS solution at large-scale.

The following subsections provide an update of the design and specification of the Resource Management and Orchestration (RMO) software modules presented in D5.1 [1] and D5.3 [1]. In addition, they describe the environment setup for orchestrating the virtual/containerized resources on top of the infrastructure layer. This software system is the glue between two layers, allowing COMPSs and Hecuba workers to deploy the resources planned to accommodate the needs of the application layer.

The work described in this section, which is part of Task 5.2, aims to achieve a fully operational deployment across diverse Cloud Service Providers.

3.2 Resource Management and Orchestrator System Capabilities

3.2.1 Technology enablers and Tool Chain Selected

The following table summarizes the technologies selected for our RMO system:

Table 2: Summary of technologies selected for I-BiDaaS RMO system

Cloudify
Cloudify is a Cloud Orchestrator used to manage the interconnection and interaction among cloud-based entities. The orchestration refers to the automation of processes and workflows required to meet the application's performance goals, minimizing the associated deployment and operation costs while maximizing the application performance. This technology offers us a reliable way to abstract various Cloud Service Providers.
TOSCA
TOSCA is an OSAIS standard specification used to describe cloud web services and their relationships. The language includes specifications to create or modify the web-services associated to a cloud-based topology. The usage of a well-known standard aims to ensure interoperability and sustainability in the future.
AWS Cloud Services
A set of Amazon Web Services (AWS) have been used to perform proof of concept deployments; such as Amazon EC2, Amazon S3 for compute and storage in the initial period of the project and in the last period Amazon ECS. Amazon Elastic Container Service (Amazon ECS) is an Amazon Web Service to run Docker applications on a scalable cluster.
Private Cloud Provider

OpenStack has been chosen to build our Private Cloud Provider. OpenStack is a free open standard cloud computing platform, mostly deployed as infrastructure-as-a-service (IaaS) to make available virtual servers. The list of services that compose the cloud stack are listed in Section 3.2.1.1.
Docker & Docker Swarms
Docker provides software components and tools for ship and run applications. The private cloud environment has been deployed on docker containers. Additional efforts have been made during the second half of the project to containerize as much software modules of the platform as possible.
Configuration Management System
The Adaptation Engine sub-module acts as a configuration management system capable to prepare and pre-package the resources that are going to be deployed across different Cloud Service Provider and Edge devices. It includes configuration recipes and a catalogue of VM and container templates.

3.2.1.1 Private Cloud Environment

OpenStack multi-node setup provides the necessary compute, network and data storage services for building a cloud-based Big Data cluster to meet the needs of the project resource deployments. The Big Data architectures benefit from high-performance storage backends.

The selected OpenStack services allow us to create storage pools to offer block storage services to the virtual instances. Moreover, storage services are offered for storing images and performing volume back-ups.

During the project course, different OpenStack distributions have been used, which in turn required reconfiguring our hosts to accommodate them.

- DevStack has been used in single node during the first phase of the project (M18).
- During the second phase additional efforts have been made for the whole platform to offer the software modules as docker containers, for this purpose, OpenStack Kolla distribution based on containers has been chosen.
- In the last phase of the project, proof of concept deployments have been archived on top of MicroStack an OpenStack distribution specially conceived for Edge/IoT environments. The Clustering mode allows to install this distribution on several nodes and to create a cloud combining the nodes together.

Requirements for OpenStack Kolla Distribution

The Cloud environment resources have to fulfil minimum requirements:

- 2 network interfaces
- 8GB main memory
- 40GB disk space

Dependencies

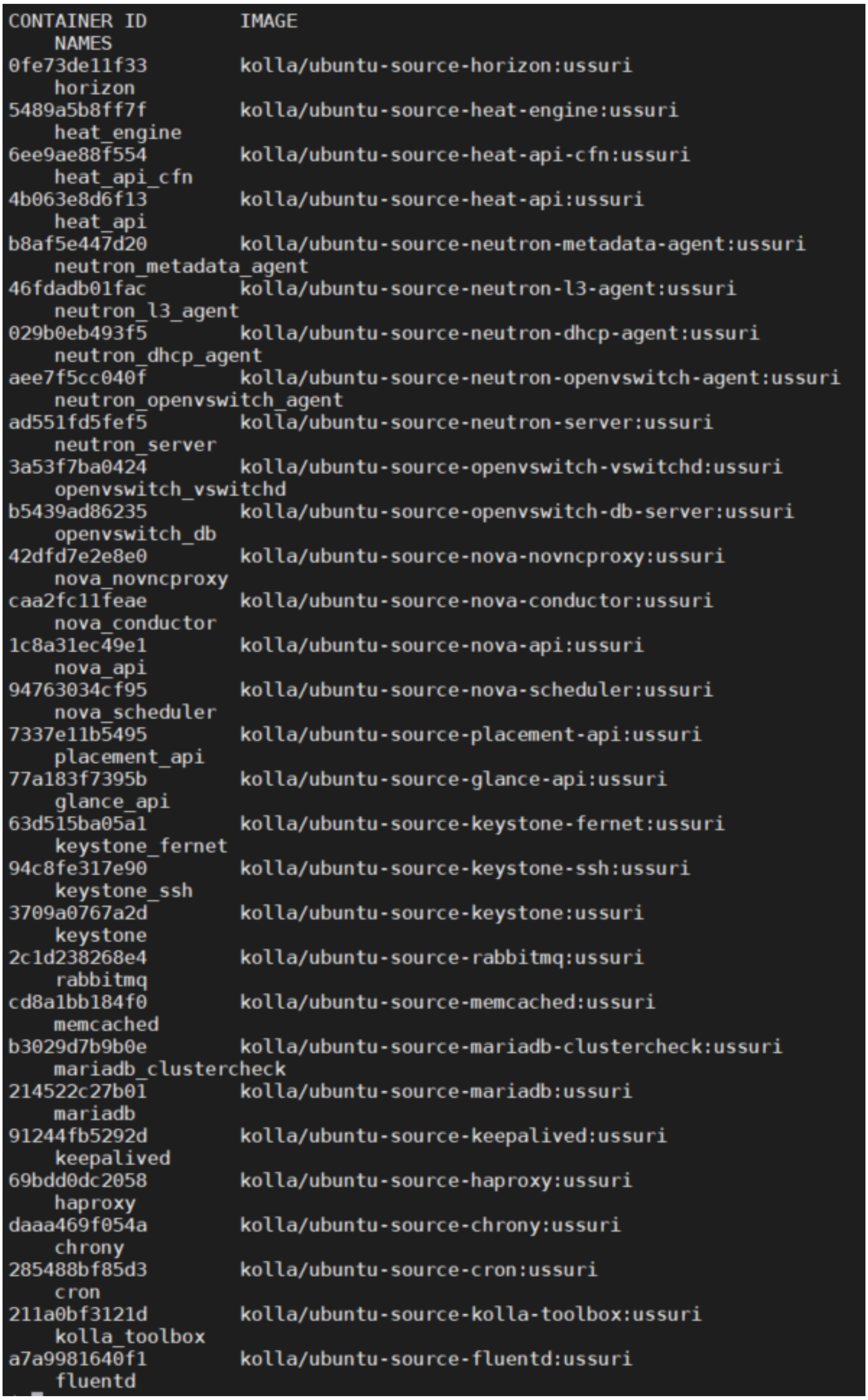
```
Docker // python3-dev // libffi-dev // gcc // libssl-dev
```

Setup

The configuration management system can be configured under **/etc/ansible/ansible.cfg**.

The first step is to configure the following configuration files: *globals.yml, passwords.yml and finally configure the* multimode *inventory*, choosing which nodes act as controller, monitoring and storage nodes and which ones are assigned as compute nodes.

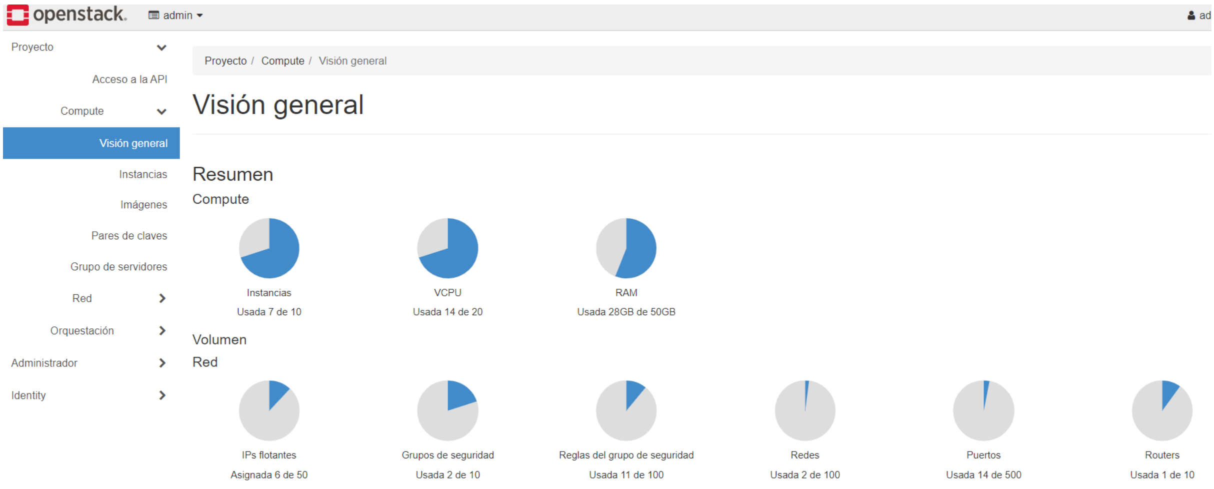
Figure 3. Openstack Kolla Controller on ITER2 shows the containers deployed on ITER2, which acts as our stack controller:



```
CONTAINER ID        IMAGE
    NAMES
0fe73de11f33        kolla/ubuntu-source-horizon:ussuri
    horizon
5489a5b8ff7f        kolla/ubuntu-source-heat-engine:ussuri
    heat_engine
6ee9ae88f554        kolla/ubuntu-source-heat-api-cfn:ussuri
    heat_api_cfn
4b063e8d6f13        kolla/ubuntu-source-heat-api:ussuri
    heat_api
b8af5e447d20        kolla/ubuntu-source-neutron-metadata-agent:ussuri
    neutron_metadata_agent
46fdadb01fac        kolla/ubuntu-source-neutron-l3-agent:ussuri
    neutron_l3_agent
029b0eb493f5        kolla/ubuntu-source-neutron-dhcp-agent:ussuri
    neutron_dhcp_agent
aee7f5cc040f        kolla/ubuntu-source-neutron-openvswitch-agent:ussuri
    neutron_openvswitch_agent
ad551fd5fef5        kolla/ubuntu-source-neutron-server:ussuri
    neutron_server
3a53f7ba0424        kolla/ubuntu-source-openvswitch-vswitchd:ussuri
    openvswitch_vswitchd
b5439ad86235        kolla/ubuntu-source-openvswitch-db-server:ussuri
    openvswitch_db
42dfd7e2e8e0        kolla/ubuntu-source-nova-novncproxy:ussuri
    nova_novncproxy
caa2fc11feae        kolla/ubuntu-source-nova-conductor:ussuri
    nova_conductor
1c8a31ec49e1        kolla/ubuntu-source-nova-api:ussuri
    nova_api
94763034cf95        kolla/ubuntu-source-nova-scheduler:ussuri
    nova_scheduler
7337e11b5495        kolla/ubuntu-source-placement-api:ussuri
    placement_api
77a183f7395b        kolla/ubuntu-source-glance-api:ussuri
    glance_api
63d515ba05a1        kolla/ubuntu-source-keystone-fernet:ussuri
    keystone_fernet
94c8fe317e90        kolla/ubuntu-source-keystone-ssh:ussuri
    keystone_ssh
3709a0767a2d        kolla/ubuntu-source-keystone:ussuri
    keystone
2c1d238268e4        kolla/ubuntu-source-rabbitmq:ussuri
    rabbitmq
cd8a1bb184f0        kolla/ubuntu-source-memcached:ussuri
    memcached
b3029d7b9b0e        kolla/ubuntu-source-mariadb-clustercheck:ussuri
    mariadb_clustercheck
214522c27b01        kolla/ubuntu-source-mariadb:ussuri
    mariadb
91244fb5292d        kolla/ubuntu-source-keepalived:ussuri
    keepalived
69bdd0dc2058        kolla/ubuntu-source-haproxy:ussuri
    haproxy
daaa469f054a        kolla/ubuntu-source-chrony:ussuri
    chrony
285488bf85d3        kolla/ubuntu-source-cron:ussuri
    cron
211a0bf3121d        kolla/ubuntu-source-kolla-toolbox:ussuri
    kolla_toolbox
a7a9981640f1        kolla/ubuntu-source-fluentd:ussuri
    fluentd
```

Figure 3. Openstack Kolla Controller on ITER2

Figure 4. Openstack Dashboard on ITER2 shows the central GUI to interact with the OpenStack services.



openstack. admin ▾ ad

Proyecto
Acceso a la API
Compute
Visión general
Instancias
Imágenes
Pares de claves
Grupo de servidores
Red
Orquestación
Administrador
Identity

Proyecto / Compute / Visión general

Visión general

Resumen

Compute

Instancias Usada 7 de 10

VCPU Usada 14 de 20

RAM Usada 28GB de 50GB

Volumen

Red

IPs flotantes Asignada 6 de 50

Grupos de seguridad Usada 2 de 10

Reglas del grupo de seguridad Usada 11 de 100

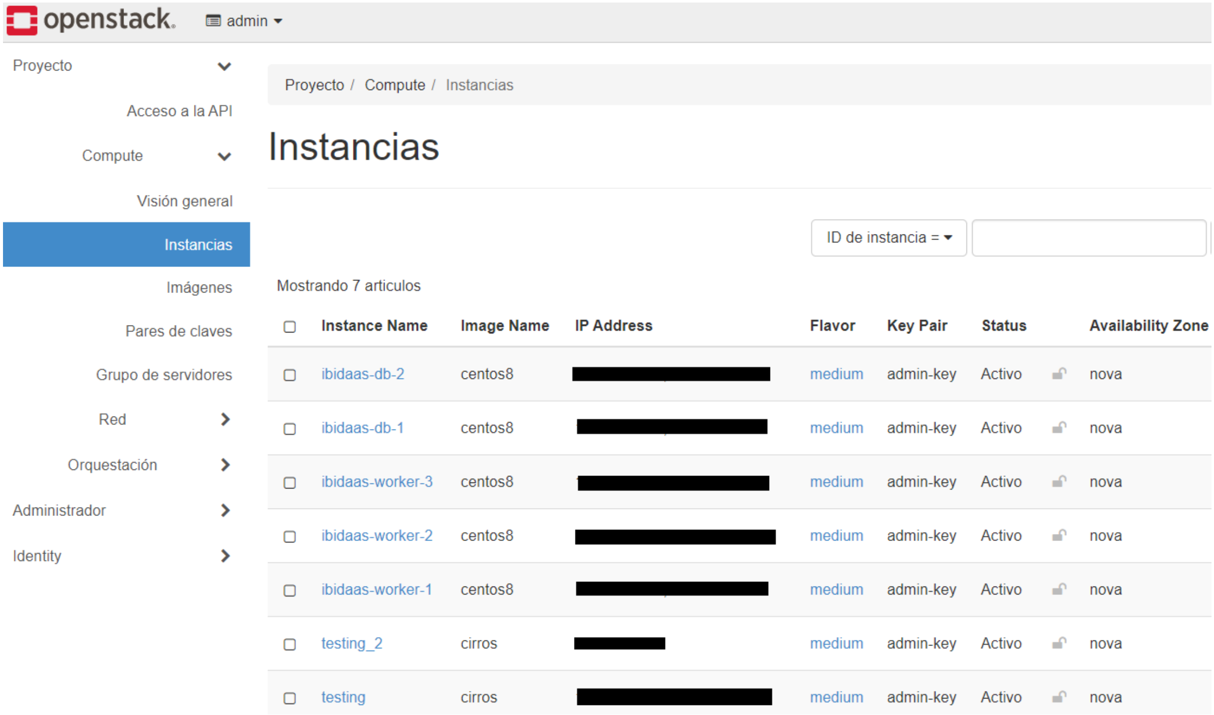
Redes Usada 2 de 100

Puertos Usada 14 de 500

Routers Usada 1 de 10

Figure 4. Openstack Dashboard on ITER2

In order to define the logical representation of a cloud-based application, it is necessary from the RMO module to specify various elements like the computing nodes to be used, how they relate to one another, or how they need to be deployed and maintained across the lifecycle of the application. The OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) specification offers a well-known standard approach for enhancing interoperability across cloud providers. The deployment descriptor blueprint defined is translated into COMPSs workers that are deployed across our configured Cloud Providers. Figure 5. Private Cloud running instances. shows the running instances on the private cloud.



openstack. admin ▾

Proyecto
Acceso a la API
Compute
Visión general
Instancias
Imágenes
Pares de claves
Grupo de servidores
Red
Orquestación
Administrador
Identity

Proyecto / Compute / Instancias

Instancias

ID de instancia = ▾

Mostrando 7 articulos

☐	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone
☐	ibidaas-db-2	centos8	[redacted]	medium	admin-key	Activo	nova
☐	ibidaas-db-1	centos8	[redacted]	medium	admin-key	Activo	nova
☐	ibidaas-worker-3	centos8	[redacted]	medium	admin-key	Activo	nova
☐	ibidaas-worker-2	centos8	[redacted]	medium	admin-key	Activo	nova
☐	ibidaas-worker-1	centos8	[redacted]	medium	admin-key	Activo	nova
☐	testing_2	cirros	[redacted]	medium	admin-key	Activo	nova
☐	testing	cirros	[redacted]	medium	admin-key	Activo	nova

Figure 5. Private Cloud running instances.

3.2.1.2 Resource Manager Orchestrator

The brain of the RMO modules is the resource orchestrator, which is based on Cloudify. This technology allows us to describe our deployments using a common specification across different types of providers. In this way, our solution aims to avoid vendor lock-in situation through the abstraction of different providers that could be integrated via plugins to our environment. The entire system can be managed through a central dashboard.

In order to define the logical representation of a cloud-based application, it is necessary to specify various elements like the computing nodes to be used, how they relate to one another, or how they need to be deployed and maintained across the lifecycle of the application. The OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) specification offers a well-known standard approach for enhancing interoperability across cloud providers.

3.3 RMO Architecture

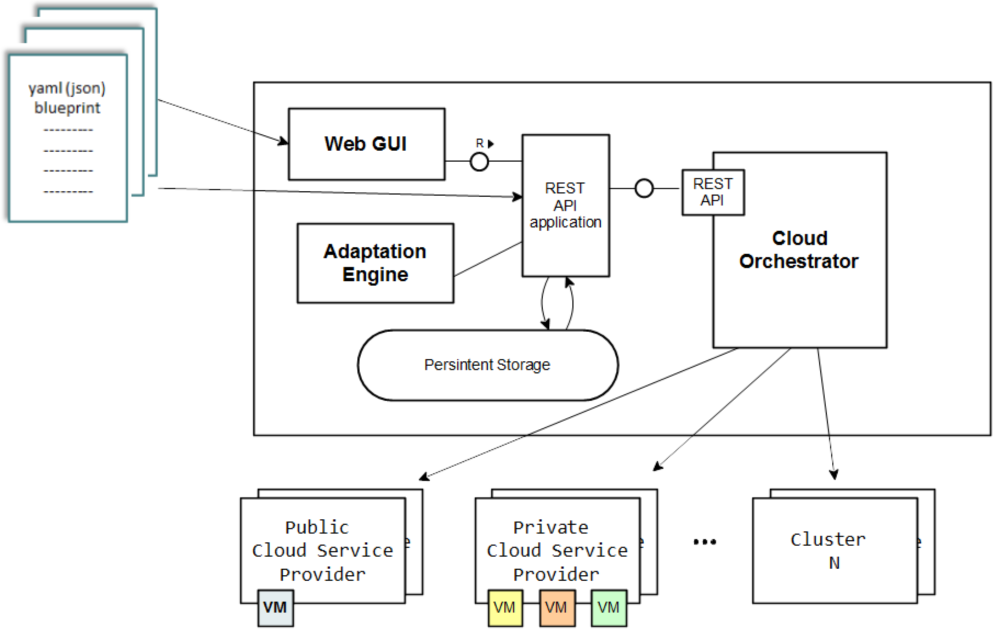


Figure 6. RMO sofware module architecture.

The architecture is depicted in Figure 6, and the software module is composed of the following components:

- A REST API that acts a northbound interface enacting the required calls through the Cloud Orchestrator module to perform the deployment of virtual resources. It consumes deployment templates describing the requirements of the service and the conditions to meet at runtime.
- The Adaptation Engine that is responsible to ensure that the service conditions are fulfilled; this is evaluated through the information exposed by the Cloud Orchestrator module, which manages the lifecycle of the service applying pre-defined elasticity rules at runtime.
- A SQL-like database for storing the application states.

The following figure (Figure 7), maps each of the targeted software modules with one of the technology enablers selected to build our environment.

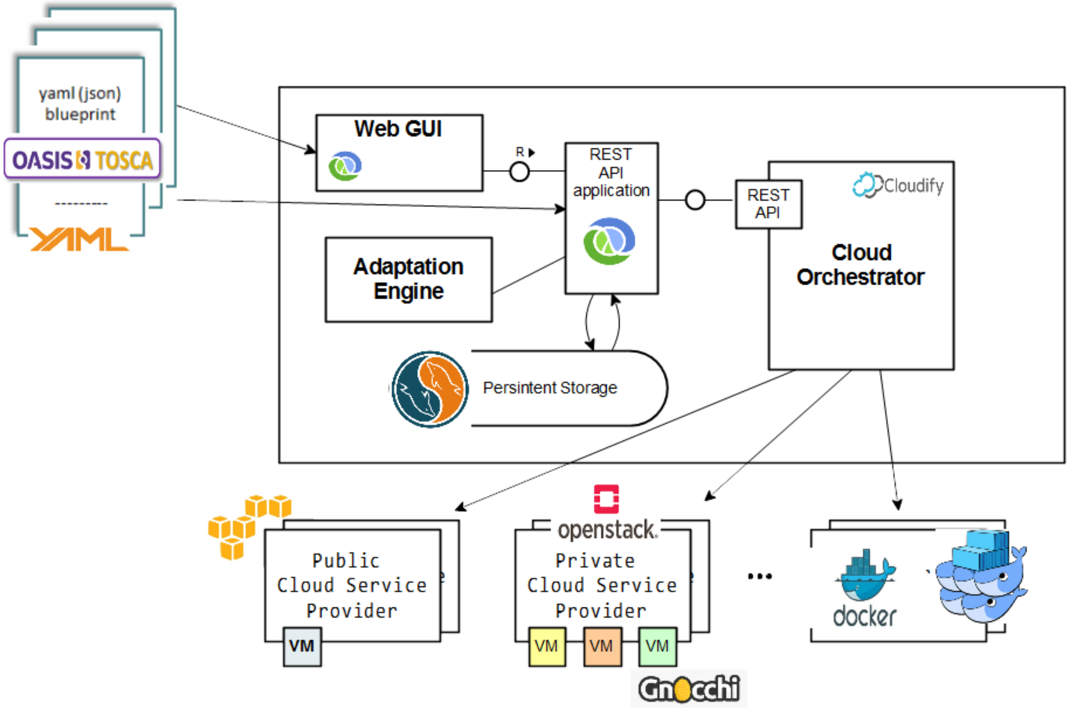
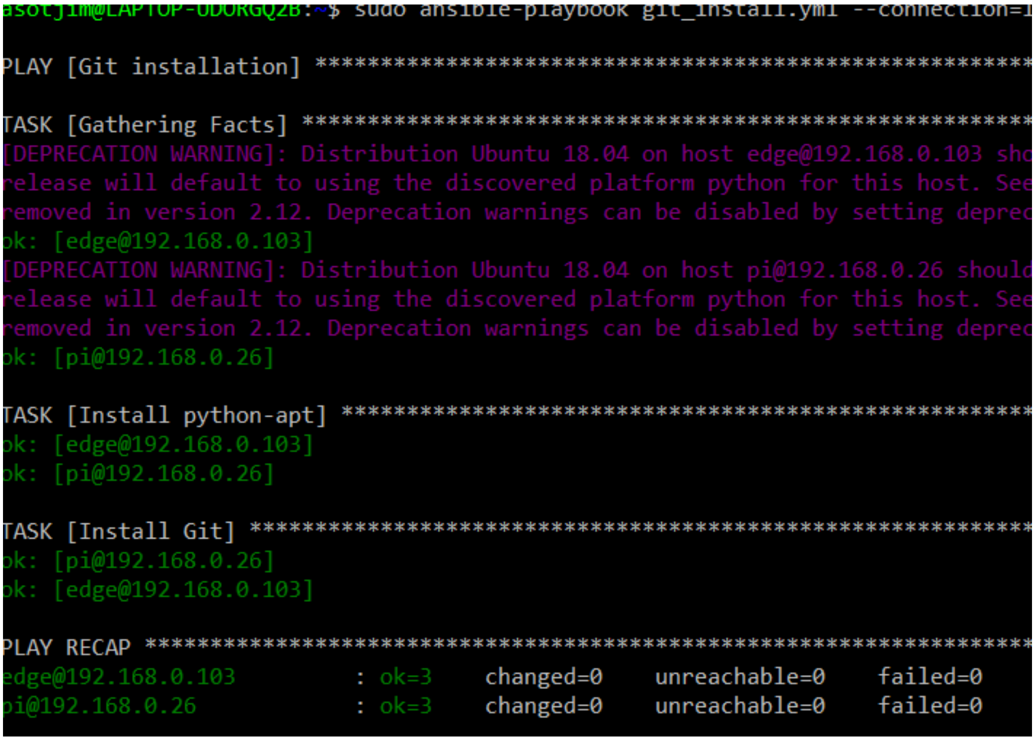


Figure 7. RMO sofware module technology map.

From M25 to M36 the Adaptation module has been developed and tested across different computing environments, the aim of the software module is to reconfigure the deployments' packages across the computing continuum (cloud-edge).

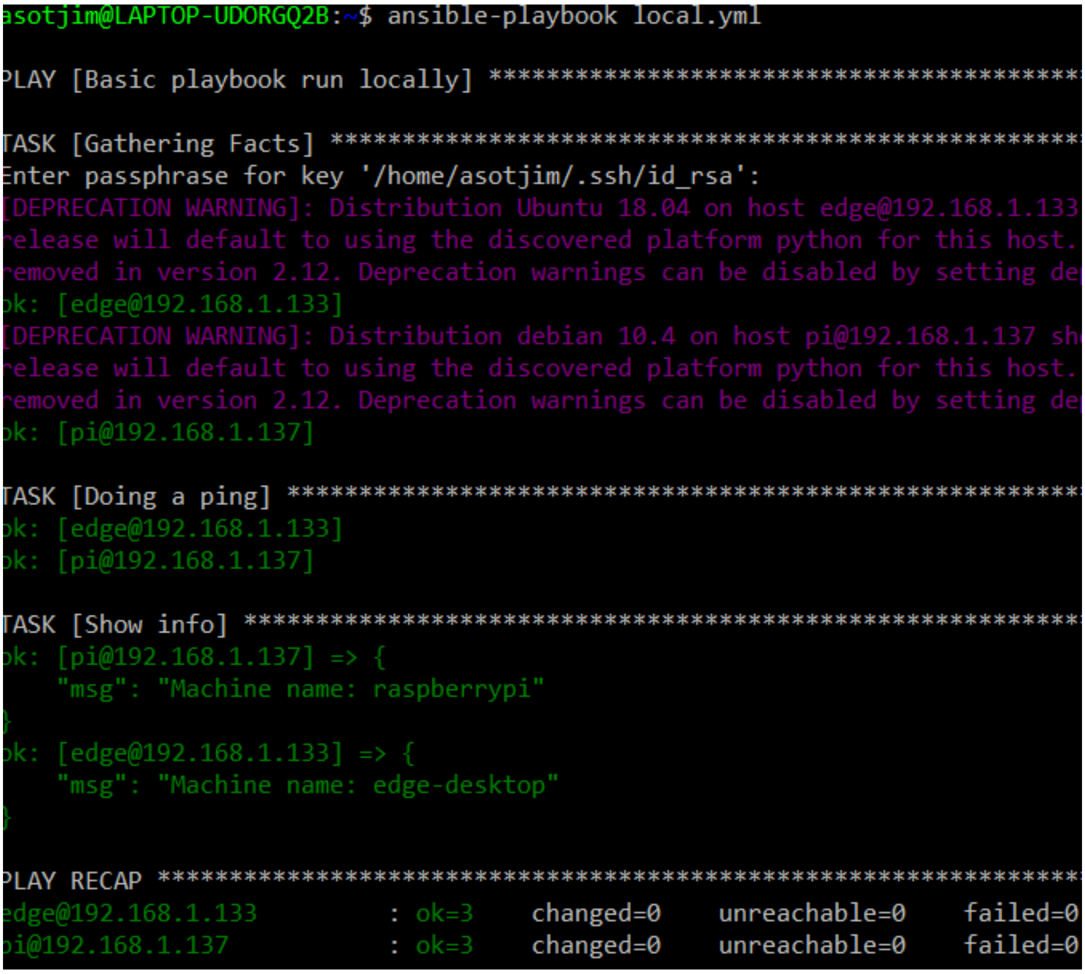
The Adaptation Engine provides a configuration management system capable to automate the lifecycle of the cloud-based deployments; it includes a catalogue of predefined automation recipes, configuration files and pre-packaged containers and VM templates. Within I-BiDaaS it has been deployed on D-Alix datacentre located in the grounds of ITER institute of Science and Technology.

Additional proof of concept deployments have been made outside of the scope of the I-BiDaaS platform to test different computing devices. Figure 8 and Figure 9 show sample deployments made on an NVIDIA Jetson Nano and Raspberry Pi 4.



```
asotjim@LAPTOP-UDORGQ2B:~$ sudo ansible-playbook git_install.yml --connection=l

PLAY [Git installation] *****************************************************

TASK [Gathering Facts] ******************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host edge@192.168.0.103 sho
release will default to using the discovered platform python for this host. See
removed in version 2.12. Deprecation warnings can be disabled by setting deprec
ok: [edge@192.168.0.103]
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host pi@192.168.0.26 should
release will default to using the discovered platform python for this host. See
removed in version 2.12. Deprecation warnings can be disabled by setting deprec
ok: [pi@192.168.0.26]

TASK [Install python-apt] ***************************************************
ok: [edge@192.168.0.103]
ok: [pi@192.168.0.26]

TASK [Install Git] **********************************************************
ok: [pi@192.168.0.26]
ok: [edge@192.168.0.103]

PLAY RECAP ******************************************************************
edge@192.168.0.103         : ok=3    changed=0    unreachable=0    failed=0
pi@192.168.0.26            : ok=3    changed=0    unreachable=0    failed=0
```

Figure 8. Setup environments for NVIDIA JetSon Nano and Raspberry Pi v.3 and v.4



```
asotjim@LAPTOP-UDORGQ2B:~$ ansible-playbook local.yml

PLAY [Basic playbook run locally] *******************************************

TASK [Gathering Facts] ******************************************************
Enter passphrase for key '/home/asotjim/.ssh/id_rsa':
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host edge@192.168.1.133
release will default to using the discovered platform python for this host.
removed in version 2.12. Deprecation warnings can be disabled by setting de
ok: [edge@192.168.1.133]
[DEPRECATION WARNING]: Distribution debian 10.4 on host pi@192.168.1.137 sh
release will default to using the discovered platform python for this host.
removed in version 2.12. Deprecation warnings can be disabled by setting de
ok: [pi@192.168.1.137]

TASK [Doing a ping] *********************************************************
ok: [edge@192.168.1.133]
ok: [pi@192.168.1.137]

TASK [Show info] ************************************************************
ok: [pi@192.168.1.137] => {
    "msg": "Machine name: raspberrypi"
}
ok: [edge@192.168.1.133] => {
    "msg": "Machine name: edge-desktop"
}

PLAY RECAP ******************************************************************
edge@192.168.1.133         : ok=3    changed=0    unreachable=0    failed=0
pi@192.168.1.137           : ok=3    changed=0    unreachable=0    failed=0
```

Figure 9. Running configuration recipes

4 Runtime Environment

4.1 Introduction

The runtime environment of the I-BiDaaS platform is responsible for providing both the execution environment for data processing applications and the interface for the data storage. The platform “should support diversified, analytic processing, machine learning and decision support techniques to support multiple stages of analysis (FR4)”, as it can be derived from the user requirements analysis. It is essential to provide the possibility to process large volumes of data. Hence, programming productivity also represents an important aspect. In order to enable distributed, parallel implementation and execution of various processing algorithms, a distributed processing capability is a necessity. However, this implies the emerging of some significant challenges in programming due to the nature of concurrent and distributed computing. These include among others working with threading, messaging, data partitioning and transfer. The complexity of the distributed computing infrastructure could also introduce some additional challenges.

Therefore, the runtime environment should provide simple ways for developing parallel and distributed applications that are able to process large amounts of data. This can improve programming productivity without sacrificing performance. The runtime environment should also provide transparent interoperability with the resource management and orchestration components, and a set of tools and interfaces that provide an efficient and easy interaction with non-relational databases for the programmers. The reason behind this is that non-relational databases are the most common solution when dealing with large data volumes and massive query workloads.

4.2 Enhanced Capabilities

The advanced machine learning module includes a set of efficient, scalable machine learning algorithms. When dealing with problems of relatively large data volumes that cannot be solved by a sequential algorithm, an optimal number of workers can be determined for each algorithm. This means that increasing the number of workers leads to execution time decrease until the optimal number of workers is reached. The developed algorithms rely on the COMPSs framework. Within I-BiDaaS final version, COMPSs and Hecuba workers use docker containers and VM templates for execution.

4.3 COMPSs and Hecuba Runtime

As described by Lordan et al. [8], COMPSs applications are implemented in a sequential way, without using APIs that deal with the details of the distributed infrastructure or with duties of parallelization and distribution (synchronizations, data transfers, …). It is very important to offer a unique and simple programming interface to create applications. This, on the one hand, means that the application will not be based on a specific API to express the interaction with the infrastructure, thus avoiding vendor lock-in and lack of portability of applications. On the other hand, we are adopting sequential programming as the main programming paradigm, which is the easiest paradigm to offer to end users, therefore achieving an easy way for users to program applications. Users do not need to think of how precisely their program is going to be run in the distributed infrastructure because the COMPSs runtime will take care of the actual execution of the application on the resources available. Instead, users only need to focus on their specific domain of knowledge to create a new program that will be able to run on the cloud or any other distributed infrastructure. Another key aspect of providing a cloud-unaware programming model is that programs can be developed once and run in multiple clouds without changing the implementation. This is very important when portability between clouds must be achieved. In COMPSs, the programmer is freed from having to deal with the specific cloud details, because COMPSs runtime will oversee it. The runtime follows a plug-in approach to deal with several cloud frameworks, hiding this burden to the end user and enabling interoperability.

Regarding COMPS's capacity to work at large scale, as a mature programming model, with more than 10 years invested in its development and testing, many different testing results have been obtained during all its development years. The most remarkable are in Amelan et al. [9], where more than 1000 execution cores are used for testing quite complex workflow structures; and in the BioExcel Project [10], where an execution using 800 nodes of MareNostrum IV (using 38,400 cores) was performed for Molecular Dynamics simulations.

While COMPSs takes care of the distribution of the computation, Hecuba[1] is in charge of optimizing the data management. Hecuba uses the NoSQL open-source database Apache Cassandra to store the data, allowing to delegate on the database the management of the global view of the data, thus avoiding explicit synchronization points and maximizing the parallelism degree. Hecuba enables users to access the data as regular Python objects stored in memory. The only requirement is previously the class that supports each data structure, specifying the data types. Hecuba allows great code simplification and it guarantees performance improvement over alternative storage solutions, such as files over high-performance parallel file systems [11].

One of the functionalities of Hecuba that is relevant for the I-BiDaaS platform is the ability to manage arrays in the format defined by the *numpy* library[2], which is highly used by AI algorithms' programmers. Hecuba, transparently to the programmer, is able to partition and distribute a *numpy* array across all the nodes of the distributed database, to facilitate the parallelism in the access. The programmer can use these arrays as regular *numpy* arrays, using all the functions implemented in the *numpy* library, regardless of whether the array is already loaded on memory or stored in the database.

[1] https://github.com/bsc-dd/hecuba

[2] https://numpy.org/

5 Conclusions

According to the DoA, for WP5 the "*the primary objective is to provide the distributed large-scale framework that permits powerful and scalable Data processing on top of heterogeneous and federated infrastructures.*"

The list of actions performed to overcome the WP challenges are summarized in the table below:

Table 3. Overall challenges of WP5 and counter-actions.

Challenge	Summary of actions to M36
Management of diverse infrastructure resources for Data Analytics (including cloud and GPU resources)	• Maintenance and extension of the private cloud infrastructure. • Maintenance of GPU resource cluster. • Privacy enhanced execution with SGX. (see Section 2)
Orchestration of infrastructure resource management across diverse resource providers	• Implementation and testing of the Resource Management and orchestration software modules and their interactions. (see Section 3)
Seamless integration and exploitation of infrastructure elasticity capabilities by the runtime environments	• The integration processes and tool chain selected toward the final I-BiDaaS prototype have been described in D5.6. • Specification of the runtime environment which provides the execution environment for data processing applications as well as interface with the data storage pools. (see Section 4)

The work carried out within this period allow us to have an operational infrastructure environment supporting the realisation of the I-BiDaaS final prototype as well as the deployment of the software modules which are part of the I-BiDaaS solution.

6 References

[1] I-BiDaaS Consortium. D5.1: Federated Resource Management for Data Analytics v1, 2018, Project Report.

[2] I-BiDaaS Consortium. D5.2: Big-Data-as-a-Self-Service Test and Integration Report v1, 2018, Project Report.

[3] I-BiDaaS Consortium. D5.3: Federated Resource Management for Data Analytics v2, 2019, Project Report.

[4] Dimitris Deyannis, Eva Papadogiannaki, George Kalivianakis, Giorgos Vasiliadis, and Sotiris Ioannidis. TrustAV: Practical and Privacy Preserving Malware Analysis in the Cloud. In Proceedings of the 10th ACM Conference on Data Application Security and Privacy (CODASPY). March 2020, New Orleans, LA, USA.

[5] Nikolaos Chalkiadakis, Dimitris Deyannis, Dimitris Karnikis, Giorgos Vasiliadis, Sotiris Ioannidis. The Million Dollar Handshake: Secure and Attested Communications in the Cloud. In Proceedings of the 2020 IEEE CLOUD 2020. October 2020.

[6] I-BiDaaS Consortium. D5.4: Big-Data-as-a-Self-Service Test and Integration Report v2, 2019, Project Report.

[7] I-BiDaaS Consortium. D1.3: Positioning of I-BiDaaS, 2018, Project Report.

[8] Lordan, F., Tejedor, E., Ejarque, J., Rafanell, R., Álvarez, J., Marozzo, F., ... & Badia, R. M. (2014). Servicess: An interoperable programming framework for the cloud. *Journal of grid computing*, *12*(1), 67-91.

[9] Amela, R., Ramon-Cortes, C., Ejarque, J., Conejero, J., & Badia, R. M. (2018). Executing linear algebra kernels in heterogeneous distributed infrastructures with PyCOMPSs. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles*, *73*, 47.

[10] BioExcel Center of E (Santamaria, 2019) xcellence (Horizon 2020 Framework program) under contracts 823830, and 675728.

[11] Santamaria, P., Oden, L., Gil, E., Becerra, Y., Sirvent, R., Glock, P., & Torres, J. (2019, June). Evaluating the Benefits of Key-Value Databases for Scientific Applications. In International Conference on Computational Science (pp. 412-426). Springer, Cham.