

D4.1 COLA development testbed infrastructure



Cloud Orchestration at the Level of Application

Project Acronym: **COLA**

Project Number: **731574**

Programme: **Information and Communication Technologies
Advanced Computing and Cloud Computing**

Topic: **ICT-06-2016 Cloud Computing**

Call Identifier: **H2020-ICT-2016-1**
Funding Scheme: **Innovation Action**

Start date of project: 01/01/2017

Duration: 30 months

Deliverable:

D4.1 COLA development testbed infrastructure

Due date of deliverable: 31/12/2017

Actual submission date: 20/12/2017

WPL: Peter Gray

Dissemination Level: PU

Version: 1.0

1 Table of Contents

1	Table of Contents.....	2
2	List of Figures and Tables.....	3
3	Status and Change History	4
4	Glossary	5
5	Introduction.....	6
6	Testbed characteristics and capacity	7
6.1	CloudBroker Platform.....	8
6.1.1	Changes to the CloudBroker platform in the context of COLA.....	9
6.1.2	Integration of Cloud Infrastructure.....	12
6.2	Cloud deployments	14
6.2.1	UoW development testbed infrastructure	14
6.2.2	SZTAKI development testbed infrastructure.....	15
6.2.3	SICS development testbed infrastructure	18
6.2.4	CloudSigma development testbed infrastructure.....	19
6.2.5	Supplementary commercial cloud resources (Amazon Cloud)	24
7	Conclusion and next steps	24
8	Annex	26
8.1	Cloud characteristics and capacity summaries.....	26

2 List of Figures and Tables

Figures

Figure 1: CloudBroker Platform involvement in the COLA project	7
Figure 2: Generic architecture of the COLA testbed infrastructure	8
Figure 3: General overview of the CloudBroker Platform architecture	9
Figure 4: COLA prototype platform functionality	13
Figure 5: COLA prototype platform showing resources connected	13
Figure 6: Architecture of the UoW cloud.....	14
Figure 7: SICS Security Lab Cloud hardware view	19
Figure 8: CloudSigma WebApp.....	20
Figure 9: CloudSigma power tweaking tools	21
Figure 10: Screenshot showing the custom server configuration screen	23

Tables

Table 1: Status Change History	4
Table 2: Deliverable Change History	4
Table 3: Glossary.....	5
Table 4: Instance types	17

3 Status and Change History

Status:	Name:	Date:	Signature:
Draft:	Peter Gray, Bogdan Despotov	24/10/17	Electronic
Reviewed:	Nicola Fantini	18/12/17	Electronic
Approved:	Tamas Kiss	20/12/17	Electronic

Table 1: Status Change History

Version	Date	Author	Modification
v0.1	24/10	Peter Gray	Initial content, introduction, CloudSigma testbed configuration.
v0.2	26/10	Bogdan Despotov	CloudSigma testbed configuration, Preliminary testing of additional MICADO components.
v0.3	20/11	Anna Shevchenko, Andrey Sereda	CloudBroker platform and integration of cloud infrastructure. Amazon cloud configuration and setup.
V0.4	24/11	Gregoire Gesmier	UoW cloud testbed described.
V0.5	27/11	Jozsef Kovacs	SZTAKI cloud testbed described.
v0.6	06/12	Nicolae Paladi	SICS cloud testbed described.
v0.7	07/12	Peter Gray	Additional section provided on resource allocation and account creation.
V0.8	11/12	Peter Gray	Introduction and conclusions written with some minor restructuring.
V1.0	20/12	Tamas Kiss	Final approval

Table 2: Deliverable Change History

4 Glossary

ACL	Access Control List
API	Application Programming Interface
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
GUI / UI	Graphical User Interface / User Interface
IaaS	Infrastructure-as-a-Service
IP	Internet Protocol
NIC	Network Interface Controller
NUMA	Non-uniform Memory Access
SLA	Service Level Agreement
SSH	Secure Shell
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
ZH4 / ZH5	Equinix Zurich data centre locations

Table 3: Glossary

5 Introduction

This deliverable presents the initial deployment, operation and support of the COLA development testbed infrastructure as outlined in the DoW. The main work package objectives relating to this deliverable are the following:

- To collect and refine functional and non-functional cloud infrastructure and access layer level requirements of typical MiCADO microservices.
- To create a production and a testbed cloud infrastructure consisting of commercial and academic cloud resources based on the requirements of the microservices.
- To enhance the cloud access layer based on the requirements of the COLA use-cases as well as from the upper and lower layers.

The objectives not mentioned here will be acted upon in the coming months and addressed in future WP4 deliverables.

The progress being reported in this deliverable is predominantly from Task 4.1 Collecting and refining requirements of MiCADO microservices, Task 4.2 Deployment, operation and support of deployment testbed infrastructure, and Task 4.5 Implementation of enhancements of the cloud access layer. The work relating to Task 4.3 and Task 4.4 will appear in upcoming deliverables.

In Section 6 of this deliverable, we describe the general testbed characteristics and capacity of the combined testbed infrastructure consisting of the CloudBroker platform, three academic clouds (UoW, SZTAKI, and SICS) and CloudSigma's commercial cloud. Amazon AWS cloud is also included here as a supplementary commercial cloud provider external to project partners. Information regarding the general characteristics, resource allocation, account creation and access criteria is provided for each cloud operator respectively. We also describe the work required to get to this point, as well as the ongoing support and maintenance required to ensure continuous operation. Finally, we offer a conclusion and describe the next steps.

6 Testbed characteristics and capacity

The testbed infrastructure is comprised of one commercial cloud operated by CloudSigma, and three academic clouds provided by SZTAKI, UoW and SICS respectively. Each partner provides a certain amount of resources dedicated for use during the project and makes them available per use-case on demand. Additional cloud resources may be purchased on demand from Amazon EC2 as required. This will ensure the platform works without issue with a second, commonly used commercial cloud provider.

In order to support multiple heterogeneous cloud infrastructures, the COLA project utilizes the CloudBroker Platform at the cloud access layer. The advantage of using such generic access layer is that the results of the project can be prototyped and validated on multiple heterogeneous clouds. The COLA testbed includes both commercial cloud resources (CloudSigma and Amazon) and also private academic clouds based on widely used open-source cloud middleware (OpenStack and Open Nebula). The CloudBroker Platform involvement in the COLA project within the MiCADO architecture is reflected in Figure 1, while the generic structure of the COLA testbed is shown in Figure 2.

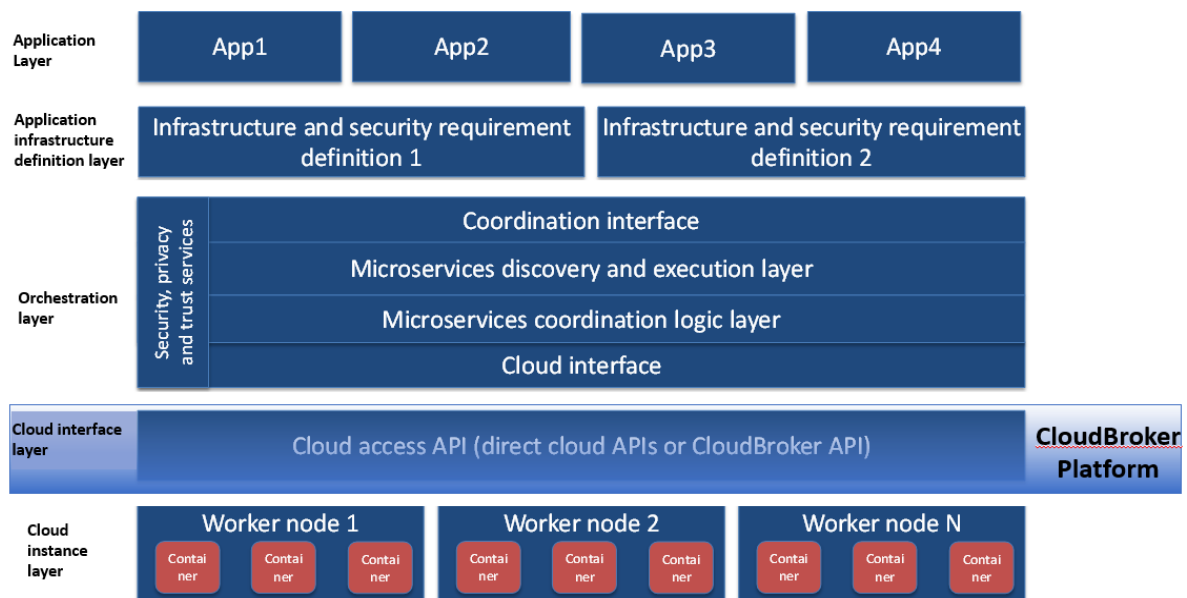


Figure 1: CloudBroker Platform involvement in the COLA project

D4.1 COLA development testbed infrastructure

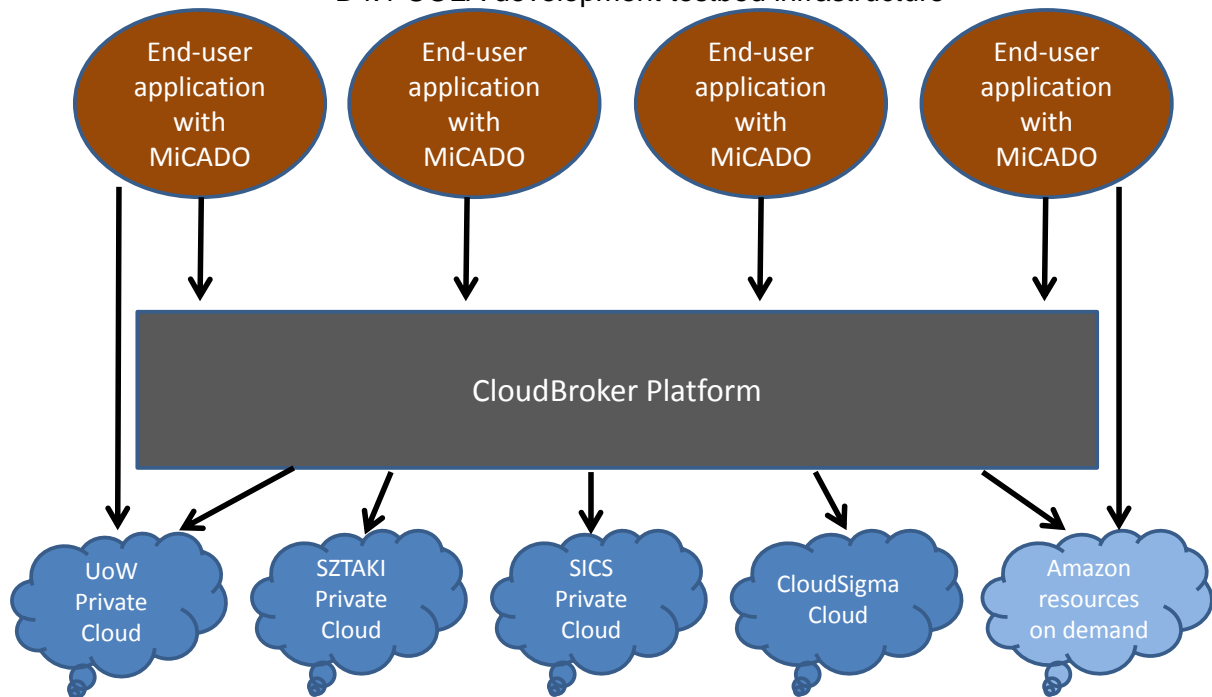


Figure 2: Generic architecture of the COLA testbed infrastructure

In order to begin planning for the deployment of MiCADO services on the testbed infrastructures provided by the four project partners, we first began by updating the existing testbed descriptions with the current characteristics and capacity available. Infrastructure providers are expected to provide timely information regarding the characteristics and capacity available via a spreadsheet. Google Sheets has been used to keep information up-to-date for the duration of the project. A time-stamped copy is periodically updated in the project document repository. More details about the functionality and available capacity of each testbed (incl. CPU cores, RAM and storage) is provided in Annex 1. Sheet 2 of the spreadsheet provides a progress log for cloud integration into CloudBroker. At the time of writing, all clouds have now been integrated. There was an issue with SICS cloud supporting OpenStack v3, while CloudBroker supports v2. However, OpenStack v3 support for the CloudBroker Platform has now been implemented and the issue has been marked as resolved.

In parallel to the testbed characteristics and capacity spreadsheet, a questionnaire created by Brunel University for Task 8.1 - Business and Technical Requirements of COLA Use-Cases - was distributed to all use-case partners to complete within the first weeks of the project. The information captured with this questionnaire, although not directly related to specific testbeds, is helpful for understanding the high-level requirements of each use-case and for matching those requirements with the appropriate testbed or a combination of testbeds.

6.1 CloudBroker Platform

The CloudBroker Platform is a web-based application store for the deployment and execution of compute-intensive scientific and technical software in the cloud. It provides a Work Package WP4

D4.1 COLA development testbed infrastructure

cloud service that allows for on-demand, scalable and pay-per-use access via the internet. The platform can be run at different physical places and under different legislation, inside and outside of the EU, if desired.

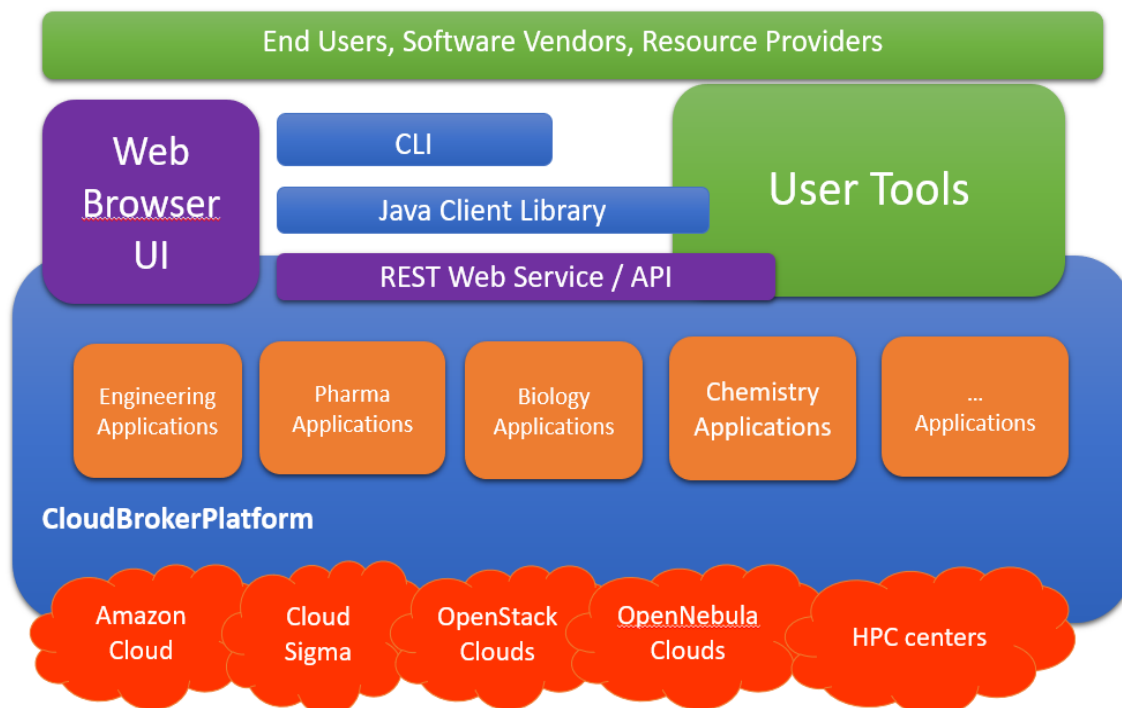


Figure 3: General overview of the CloudBroker Platform architecture

6.1.1 Changes to the CloudBroker platform in the context of COLA

While the CloudBroker Platform itself already exists for several years, as part of COLA project various CloudBroker Platform features and improvements were implemented. A more detailed description of these features and improvements is given in the chapter below.

6.1.1.1 Modifications of the CloudBroker Platform within M1-12

During the M1-12 of the project a number of upgrades to the CloudBroker Platform have been implemented. The related list and the details are provided in this chapter.

1. Setup of the dedicated CBP server for the COLA project. Shortly after the project started CloudBroker setup a specific CBP sever for the project needs. The server is used by the project partners in order to run tests and experiments such as testing of the newly integrated cloud infrastructure, MiCADO-related developments, specific use case experiments. The server is available via the link <https://cola-prototype.cloudbroker.com>.

2. Technology update. Just before the new server was setup, the technology used for CBP has been updated to the newest version.

3. Clouds connected to the new CBP server. As requested by the COLA project partners, several clouds have been connected to the COLA Prototype Platform to be used for the test and experiments. The list of the clouds connected is as follows:

D4.1 COLA development testbed infrastructure

- a. **SICS cloud** (OpenStack)
- b. **SZTAKE cloud** (OpenNebula)
- c. **UoW cloud** (OpenStack)
- d. **CloudSigma cloud**
- e. **Amazon cloud**

4. Adapter updates. Several updates to the cloud adapters were introduced, namely:

- a. OpenStack Nova adapter update.
 - i. Support for v3 API added.
 - ii. Image generation mechanism updated – now the mechanism is more stable.
- b. OpenNebula adapter update.
 - i. Support for v4 API/authorization added.
 - ii. New tool is used for better communication.
- c. S3 compatible storage update.
 - i. Updated version of the tool is used for communication.
 - ii. Added support for the latest authorisation mechanism.
- d. CloudSigma adapter update.
 - i. Usage of Key Pair mechanism has been introduced.
 - ii. Handling of the security groups has been upgraded.
 - iii. Control over instance stopping has been enhanced.
- e. Amazon adapter update.
 - i. Mechanism of failure handling has been updated.

5. Shifting focus from jobs to instances. The CBP was initially designed to be job-oriented, namely minimal access to instance information was provided to user, there was no way to connect to an instance, there was no way to start an instance independently from jobs. During the COLA project it is expected that the focus of the Platform will be shifted a bit to become more instance-oriented. The shift has already started shortly before the project. Also, major steps were done during the project already. Now, the following new features are introduced:

- a. There is functionality to control instances independently from jobs, including API and UI.
- b. There exist different ways to connect to an instance, e.g. port opening, key pairs usage.
- c. CloudInit support has been introduced.

During the project it is expected to move further and make the Platform even more instance-oriented and, thus, more efficient and convenient for specific use cases that require more instance-oriented approach.

D4.1 COLA development testbed infrastructure

6. Instance control features upgrade. As already mentioned, a number of enhancements related to the instance control have been introduced.

- a. **Port opening** functionality has been extended – now ports can be opened by default by any user who has enough rights for it (without applying additional configurations as it was before).
- b. **Ranges of ports** are supported: if a user would like to open several ports or a range of ports – it is now possible to do so.
- c. **SSH key pairs are no longer restricted by IP.** Previously it was possible to connect to an instance in a cloud via SSH using the key pairs from specific IP only. Now it is possible to connect via SSH to an instance without being restricted by specific IP.
- d. **Several SSH key pairs** can be used to connect to an instance. When launching an instance from CBP, it was possible to attach only one key pair to the instance and then use it to connect to the instance. Now it is possible to attach several key pairs to an instance and use them for connection.
- e. **CloudInit support introduced.** Now it is possible to configure an instance using a CloudInit script provided on instance launch from CBP.
- f. **Instance password autogeneration** on launch was added. When instance is launched, each time a new password for instance access is generated making the instance more secure.

7. User documentation update. User API manual has been updated so that it is more convenient for the usage. Now partners can find the API information, description and examples are available online via <http://api-docs.cloudbroker.com/>.

6.1.1.2 Planned modifications of the CloudBroker Platform

During the next month it is planned to further work on the new CBP features depending on the requirements collected by WP8 from the use cases.

However, in addition to the use case requirements, there are some features that will be obviously beneficial for the project. The list of such features is provided below. It is worth mentioning, that the plans as for the features mentioned below are not final and CloudBroker will further work on the planning of the modification during the later stage of the project.

1. Moving focus further to instances. As it was mentioned in the previous chapter, CloudBroker Platform tends to become more instance-oriented. It was already mentioned that some significant steps have already been taken for this, however, there are still expected activities such as:

- a. It is planned to have less limitations on instance control: e.g. there should be no need for a user to have a cloud account in order to launch and manage instances. This will facilitate instance control and management by users.
- b. More instance tuning functionality is planned to be added: e.g. advanced firewall rules, etc..
- c. It is planned to introduce functionality to forbid Platform to connect to a cloud instance.

D4.1 COLA development testbed infrastructure

2. Enhanced Service Catalogue. It is planned to introduce extensions for the existing CBP cloud catalogue. The following features should be added:

- a. Keeping track of Clouds available:
 - i. It will be possible for users to access different clouds via the Platform. The whole list of the clouds will be provided via UI and API.
 - ii. If a cloud can be accessed instantly – a user will be able to use it.
 - iii. If a cloud is available upon a certain procedure only, the procedure will be described to a user.
 - iv. Additional resources will be added depending on the partners' requirements.
- b. SLA catalogue of cloud service providers:
 - i. Mechanism to check and monitor cloud SLAs will be developed.

3. Automatic Cloud Selection. Smart 'CloudAdvisor' system will be introduced:

- a. A querying system will be developed to provide a user with recommendations of a cloud to be used for his or her specific purposes.
- b. For this, customer profiling system will be developed, which will analyze specific needs of each user.

4. Enhanced Docker Support. Currently the very basic Docker support is available from the Platform. However, it would be helpful for the users to add enhanced Docker support so that users can freely use Docker via the Platform.

6.1.2 Integration of Cloud Infrastructure

Currently the following cloud adapters are supported on the CBP:

- CloudSigma adapter;
- Amazon EC2 adapter;
- OpenStack Nova;
- OpenStack EC2;
- OpenNebula.

Also, the following clusters are supported:

- HLRS NEC Cluster;
- CINECA Galileo Cluster;
- ROMEO Cluster;
- ETH Euler Cluster;
- IBNBADIS Cluster.

The figure below illustrating the functionality allowing to use any of the listed adapters during resource registration from CBP is shown below.

D4.1 COLA development testbed infrastructure

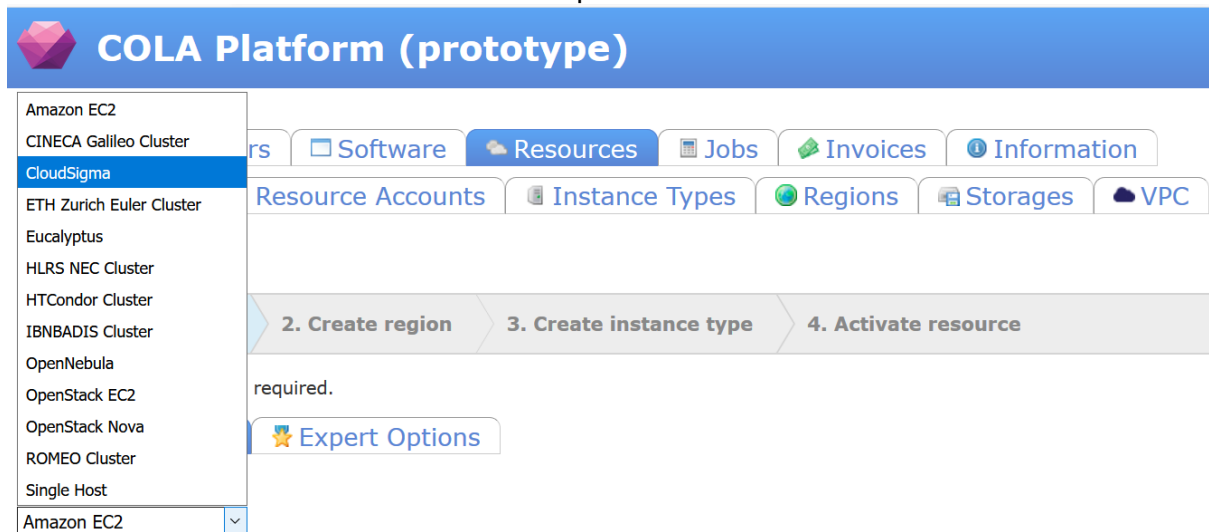


Figure 4: COLA prototype platform functionality

Thus, any cloud using the abovementioned technology or any of the listed clusters can be easily connected to the COLA Platform Prototype.

Furthermore, the academic and commercial clouds available for the COLA project has already been connected to the COLA prototype platform. The clouds connected are the following:

- SICS cloud (OpenStack)
- SZTAKE cloud (OpenNebula)
- UoW cloud (OpenStack)
- CloudSigma cloud

Additionally, Amazon cloud is also connected to the Platform for additional tests. In the figure below the abovementioned resources connected to the COLA Prototype Platform are shown.

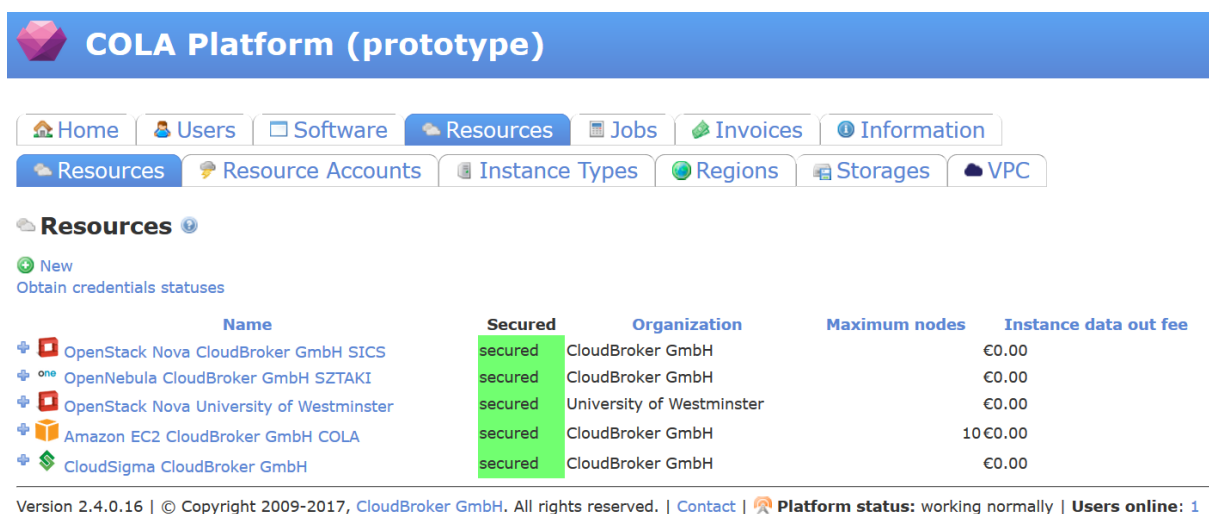


Figure 5: COLA prototype platform showing resources connected

D4.1 COLA development testbed infrastructure

In the future it is expected that more adapters will be added to the Platform and more clouds/HPCs will be connected depending on the use case requirements.

6.2 Cloud deployments

The following subsections describe in more detail the four available testbeds provided by the project partners, UoW, SZTAKI, SICS and CloudSigma.

6.2.1 UoW development testbed infrastructure

General characteristics: The University of Westminster operates an IaaS (Infrastructure-as-a-Service) cloud computing cluster intended for research use and teaching services provision. The cluster is an OpenStack Juno based infrastructure. The underlying software is based on LibVirt as virtualising API and KVM as hypervisor technology. This technology has been proven to be stable for several years and has become one of the virtualisation standards in the industry and the academy sectors.

Managing is done using the concept of *tenants* or *projects* defined by the cloud administrator, each *tenant* can use a defined number of resources set by the administrator. These resources include number of CPUs, GB of RAM memory, disk space and Public IPs.

Users manage the cloud resources either via EC2 and S3 APIs (Amazon compatible), NOVA API or Horizon web interface. Security is based on security groups that isolate the cloud instances (virtual machines running) from each other based on the users and/or project they belong to. Inside each project, users can also define their own firewall rules and as many security rules as the administrator has defined for them.

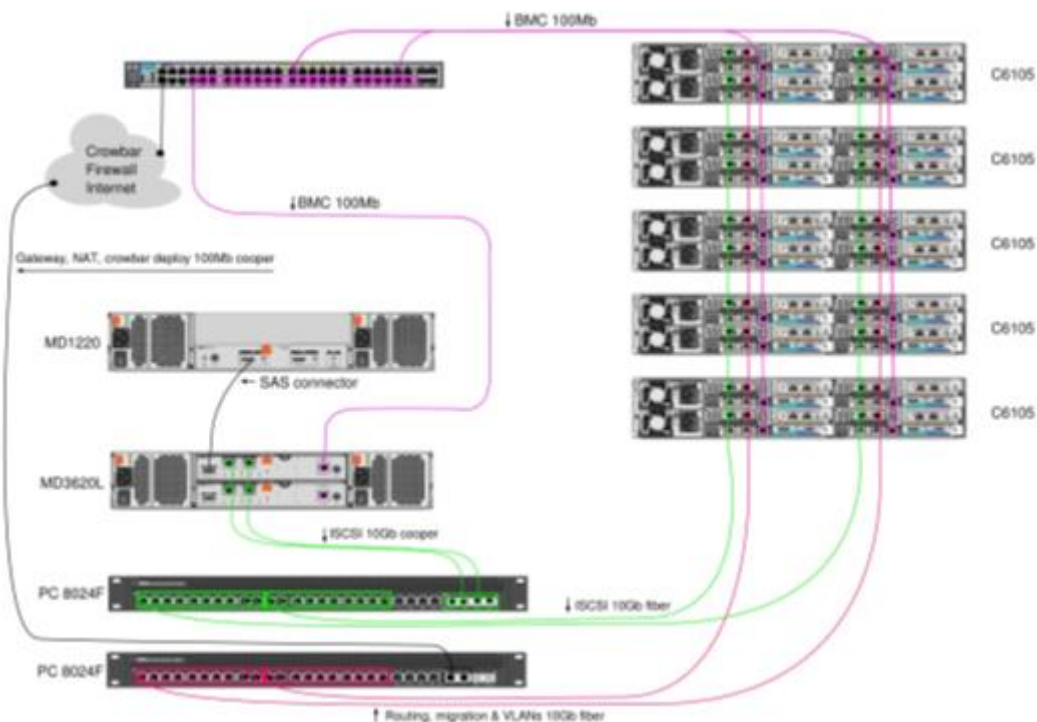


Figure 6: Architecture of the UoW cloud

D4.1 COLA development testbed infrastructure

Computer force is based on 28 Dell C6105 doing a total of:

- 248 CPUs (AMD Opteron 4122 Processor (2.2GHz, 4C, 4x512K L2/6M L3 Cache, 75W ACP), DDR3- 1333MHz).
- 2376GB RAM memory (Dual Rank LV RDIMMs 1333MHz)

Storage force is based on:

- 5 TB RAID1 based, local storage, 40 x (SATA 7.2k 2.5" HD Hot Plug).
- 12 TB RAID1 based, (PowerVault MD3620i External 10Gb iSCSI + PowerVault MD1220 Base extension).
- Huawei OceanStor 5500 (18 x 550GB SSD + 24 x 3 TB NL-SAS disks), replicated to Huawei OceanStor 5300 (24 x 3 TB NL-SAS disks)

Networking is based on:

- PowerConnect 8024F 10GbE optical fibre switches for VLAN and storage connectivity.
- Standard 100M ethernet switches for administration and live migration purposes.

Accessing the UoW cloud: To use the UoW OpenStack, the user needs to have an account on the Westminster organisation on the CloudBroker platform: (<https://cola-prototype.cloudbroker.com>). To do so she/he would need to request an account to admin of the Westminster organisation. External users can only access the UoW cloud to deploy processing application through the CloudBroker platform, as currently there's no support for user direct access. Once the account was created on the platform, the user will be allowed to use the UoW OpenStack. Detailed user guide on how to create a new Software which will be deployed on the UoW cloud through the platform is available via the following link:

<https://drive.google.com/open?id=0B6CGH9Z-GeVKNzdnWmJzYIJXVWs>

6.2.2 SZTAKI development testbed infrastructure

SZTAKI operates an IaaS cloud used both for research and hosting services. The infrastructure is based on OpenNebula 5.2. The underlying virtualisation technique is KVM (hardware virtualisation), and storage is provided both by a Ceph distributed storage cluster and a QCOW2 store.

Users can manage the cloud resources via browser based graphical interface (Sunstone) or via an EC2 interface. Additionally, an S3 based storage is part of the infrastructure (via RadosGW of Ceph).

Currently 4 VM hosts (HPE ProLiant DL385p Gen8 and Dell PowerEdge R815 models) are allocated for the cluster with combined 144 CPU cores and 512GB RAM:

- Host 1: 2x AMD Opteron(tm) Processor 6376 2.3GHz CPU (32 Cores total), 128GB RAM
- Host 2: 4x AMD Opteron(tm) Processor 6262 HE 1.6GHz CPU (64 Cores total), 256GB RAM
- Host 3: 4x AMD Opteron(tm) Processor 6164 HE 1.7GHz CPU (48 Cores total), 128GB RAM

D4.1 COLA development testbed infrastructure

Storage is available for the cluster as follows:

- 18.1TB QCOW2 storage
- 31.5TB Ceph based distributed storage.

Networking is provided by Cisco Nexus 3548 (10Gb), and HP 5920 (10Gb Ethernet) switches. Ceph storage nodes are connected via 2x 10Gb Ethernet links to the network. Additionally, 1Gb Ethernet switches are used for management purposes.

6.2.2.1 Accessing SZTAKI Cloud

Details of resource limits and accessing the infrastructure are available in the following document: <https://gitlab.com/lpds-public/documents/blob/master/COLA/sztaki-opennebula-access.md>

Next we include the text from the document above at the time of writing of this deliverable. Please always refer to the link above for the up to date documentation.

Accessing the SZTAKI cloud directly

In this access mode, you are authorised to launch virtual machines under your personal account created for you. You will be able to login to the user interface (sunstone) of the SZTAKI OpenNebula cloud and will have full control over your virtual machines and personal settings. You can also launch virtual machines via EC2 interface under your account.

Quota restrictions per user:

- 4 VMs (mapped on max 2 physical cores)
- 4 GB RAM
- No public IP address

Accessing the SZTAKI cloud through CloudBroker

In this access mode, you initiate the creation of virtual machines under your account on the COLA CloudBroker platform. The virtual machines will then be launched on the SZTAKI OpenNebula cloud under a predefined user account owned by CloudBroker.

Quota restrictions (combined for all VMs created via CloudBroker):

- 16 VMs
- 16 GB RAM
- No public IP address

How to login to the virtual machines

All virtual machines launched under the SZTAKI OpenNebula cloud for COLA users have only private ip addresses. To access your virtual machine, we provide OpenVPN access for the SZTAKI OpenNebula network to which the private ips belong to. For this purpose we provide a client OpenVPN configuration file. More details can be found below on this topic.

Access request

D4.1 COLA development testbed infrastructure

You can get access to the SZTAKI OpenNebula cloud by sending a request by email to devnull@lpds.sztaki.hu. The mail must contain the following information:

- surname
- family name
- company
- framework ("COLA project")
- access mode ("Direct" or "CloudBroker")

As an answer you will get the followings:

- For CloudBroker access mode:
 - OpenVPN configuration file (e.g., sztaki_opennebula.ovpn)
- For direct access mode:
 - Username/Password for SZTAKI OpenNebula
 - OpenVPN configuration file (e.g., sztaki_opennebula.ovpn)

Technical parameters

- SZTAKI OpenNebula (Sunstone GUI):
 - endpoint: <https://opennebula.lpds.sztaki.hu>
 - username/password: sent by email
- SZTAKI OpenNebula (EC2 endpoint):
 - endpoint: <https://opennebula.lpds.sztaki.hu:4567>
 - accesskey: equals to the username sent by email
 - secretkey: SHA1SUM of the password sent by email (see help below)
 - regionname: ""
 - image id: ami-00000243 (preferred Ubuntu 16.04)
 - instance types: e.g., t2.small (see below)
- CloudBroker details for VM creation:
 - endpoint: <https://cola-prototype.cloudbroker.com>
 - resource: "OpenNebula CloudBroker GmbH SZTAKI"
 - deployment: "Linux Ubuntu 14.04 Budapest x86_64 R1"
 - instance type: "OpenNebula CloudBroker GmbH SZTAKI m1.small"
 - region: "Budapest".

Type	VCPU	Mem (GiB)
t2.nano	1	0.5
t2.micro	1	1
t2.small	1	2
t2.medium	2	4
t2.large	2	8

Table 4: Instance types

Calculating the SHA1SUM of your password

- Linux
 - Type your password (sent by email) as input for the following command:
 - `read -s pwd; echo -n $pwd | sha1sum | cut -f1 -d' '; unset pwd`
- Windows
 - Create a txt file with your password (e.g. c:\mypwd.txt)
 - Open a command prompt (cmd.exe)
 - Run the following commands:
 - powershell
 - `Get-FileHash C:\mypwd.txt -Algorithm SHA1 | Format-List`
 - Delete c:\mypwd.txt

Using OpenVPN

- Linux
 - Install the openvpn package: e.g. `apt-get install openvpn`
 - Start openvpn using the received openvpn config file (e.g., `sztaki_opennebula.ovpn`): `openvpn ../sztaki_opennebula.ovpn`
- Windows
 - Download the OpenVPN client: <https://swupdate.openvpn.org/community/releases/openvpn-install-2.4.4-l601.exe>
 - Install the OpenVPN client.
 - Import the received openvpn config file (e.g., `sztaki_opennebula.ovpn`), and connect.

6.2.3 SICS development testbed infrastructure

General characteristics: The Security Lab at RISE SICS operates an Infrastructure-as-a-Service computing deployment dedicated exclusively for research purposes. The deployment includes 8 Dell R220/R230 rack servers based on Intel chipsets of two generations. The cluster operates on the OpenStack Otaca cloud computing platform using the *Libvirt* virtualization API and *KVM* hypervisors.

Tenants: The OpenStack cloud platform implements resource partitioning between *tenants* operating over a quota set by the administrator. To request a tenant account or changes in the quota definition, email the administrator. Nicolae Paladi is currently the administrator of the Security Lab research cloud; up to date contact details: <https://www.sics.se/people/nicolae-paladi>

Tenants can manage their allocated resources via the NOVA API or the Horizon web interface. OpenStack enforces resource isolation between the tenant quotas. In turn, tenants can create security groups within their quota and are exclusively responsible for managing them.

The aggregate set of resources in the cluster is as follows:

D4.1 COLA development testbed infrastructure

- 32 cores distributed as follows:
 - 4 x Intel Xeon E5-2430 2.20GHz, 15M Cache
 - 4 x Intel Xeon E3-1230v6 3.7GHz, 8M cache, 4C/8T
- 448 GB RAM (64GB RDIMM, 1333 MHz, Low Volt, Dual Rank)
- 14 TB RAID1 local storage, 14 x 3.5-in, 7.2K RPM Hard Drive (Hot-plug)
- 2 x Gigabit Ethernet ports on each host.

The figure below illustrates the hardware view of the SICS Security Lab cloud.

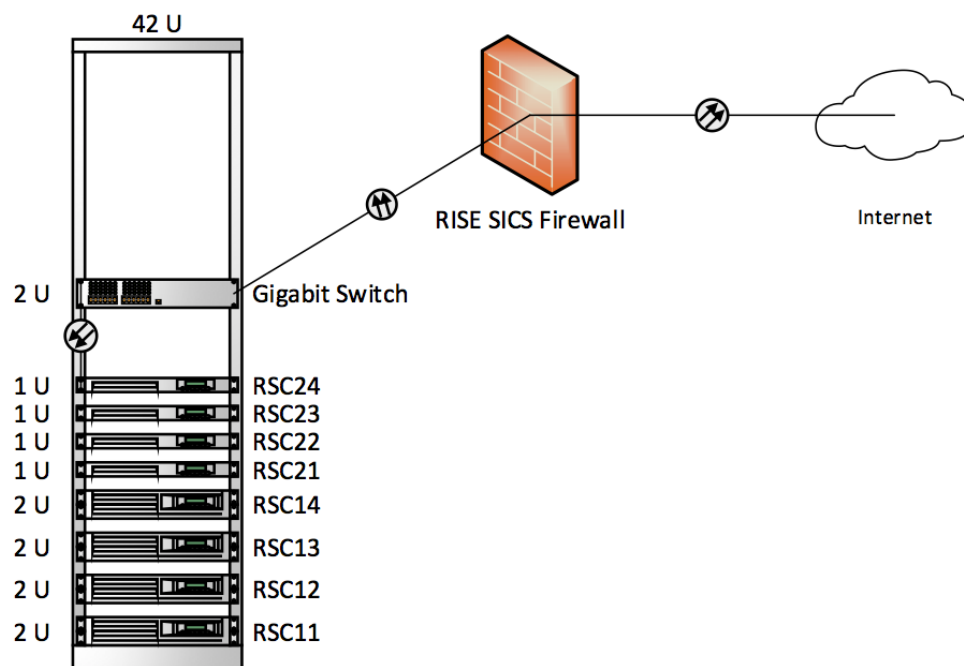


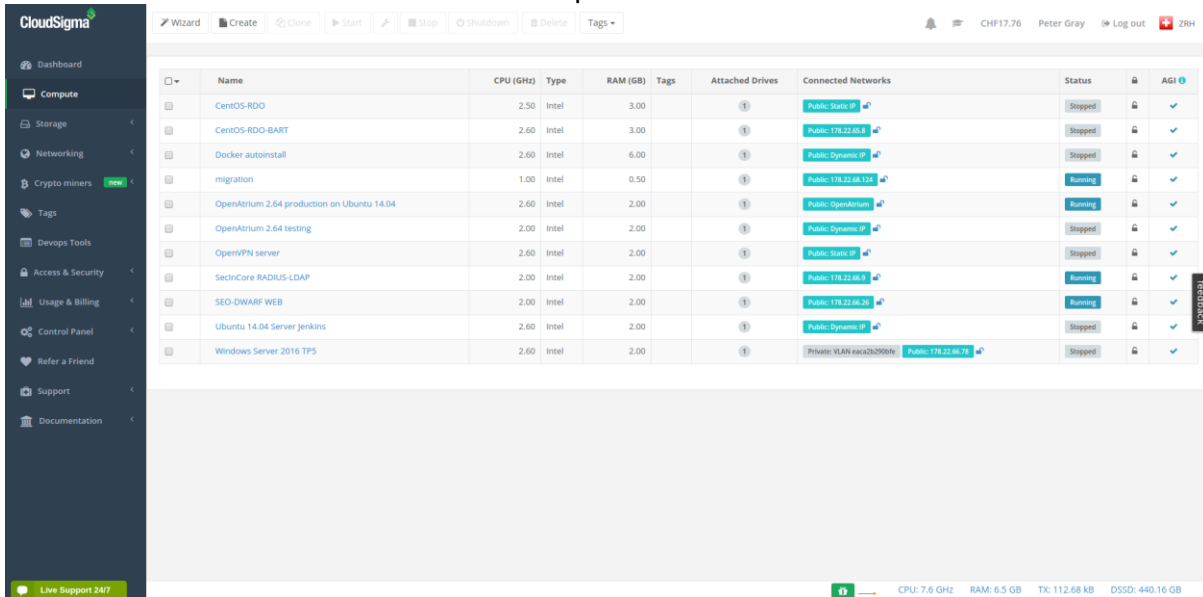
Figure 7: SICS Security Lab Cloud hardware view

Access and Usage: To access the Security Lab research cloud, the user must request an account with the administrator of the Security Lab research cloud. The request must contain a specification of the amount of resources requested for the respective tenant quota. The user may have to read and acknowledge (through a signature) a user agreement that regulates the use of the cloud resource.

6.2.4 CloudSigma development testbed infrastructure

CloudSigma is one of the most customizable cloud providers on the market with a focus on open design and flexibility with regards to computing deployments. The cloud platform is designed to provide an environment with the same degrees of freedom that private in-house environments are able to offer. All functionality is available via an API or the end-user WebApp. Below we describe in more detail how this is offered to the project.

D4.1 COLA development testbed infrastructure



Name	CPU (GHz)	Type	RAM (GB)	Tags	Attached Drives	Connected Networks	Status	AGI
CentOS-RDO	2.50	Intel	3.00		1	Public Static IP	Stopped	✓
CentOS-RDO-BART	2.50	Intel	3.00		1	Public 178.22.63.8	Stopped	✓
Docker autoinstall	2.50	Intel	6.00		1	Public Dynamic IP	Stopped	✓
migration	1.00	Intel	0.50		1	Public 178.22.68.124	Running	✓
OpenAtrium 2.64 production on Ubuntu 14.04	2.50	Intel	2.00		1	Public OpenAtrium	Running	✓
OpenAtrium 2.64 testing	2.00	Intel	2.00		1	Public Dynamic IP	Stopped	✓
OpenVPN server	2.50	Intel	2.00		1	Public Static IP	Stopped	✓
SecuCore RADIUS-LDAP	2.00	Intel	2.00		1	Public 178.22.66.1	Running	✓
SEO-DWARF-WEB	2.00	Intel	2.00		1	Public 178.22.66.25	Running	✓
Ubuntu 14.04 Server Jenkins	2.50	Intel	2.00		1	Public Dynamic IP	Stopped	✓
Windows Server 2016 TP5	2.50	Intel	2.00		1	Private VLAN vsw20160601 Public 178.22.66.79	Stopped	✓

Live Support 24/7

CPU: 7.6 GHz RAM: 6.5 GB TX: 112.68 KB DSSD: 440.16 GB

Figure 8: CloudSigma WebApp

CloudSigma provides a full-featured production cloud consisting of primary (Equinix ZH4) and secondary (Equinix ZH5) Swiss cloud locations. The total cloud resources dedicated to the COLA project as indicated in the Grant Agreement are: 100Ghz CPU, 100GB RAM, 1500GB SSD.

In a real-world scenario, customers are able to provision processing, storage, networks and other fundamental computing resources in an unbundled way meaning CPU, RAM, storage and bandwidth can be purchased and combined independently to allow the best combination of cloud resources without the limitation of fixed sizes. This means any combination of CPU and RAM can be achieved along with multiple drives mounted to a server and multiple networking interfaces. The same level of flexibility is offered to the COLA project, as long as the total resource limits are not exceeded. Initially, the resources were allocated to the CloudBroker account in CloudSigma, as it should be used as a gateway for all future deployments. However, in the beginning of the project, the complete functionality of the CloudSigma API was not exposed in CloudBroker, therefore some partners needed individual accounts for testing, which CloudSigma provided. Due to this, CloudSigma has created a custom script that is run periodically and checks the overall resource usage for all partner accounts. In a commercial situation, each resource is billed separately and transparently as either a subscription or as pay-as-you-go in 5-minute billing segments, enabling customers to track exactly how much their cloud servers are costing over time and how they are billed for it accurately. CloudSigma provides the resources listed above by way of a heavily discounted 3 year subscription to cover data centre costs, and overheads.

Essentially, the project can take advantage of the same cloud offering CloudSigma provides to paying customers, big or small. We describe some of the main advantages below.

Uniquely, any x86 based operating system and software can be used with complete administrator/root control including all variants of BSD, Linux and Windows. End-users can Work Package WP4

D4.1 COLA development testbed infrastructure

upload their own raw ISO image, attach CPU and RAM to it and boot it up. This allows full backwards compatibility from the platform.

All cloud servers and drives are persistent and modelled on the same methodology as physical dedicated server equivalents (i.e. drives, NICs etc.) VLANs and IP addresses are also controlled using standard behaviour and support all types of traffic including multicast and broadcast traffic, which is critical for high availability infrastructure in failover.

By creating a tighter link between the application and infrastructure layer through the virtual abstraction of the hypervisor, the IaaS cloud platform exposes a number of power tools allowing end-users to achieve greater performance levels from their cloud servers. For performance sensitive workloads this is especially important. These power tools include the ability to define the virtual core size, to expose NUMA topology and to tweak hypervisor timer settings for maximum performance. Additional power features include exposing the full CPU instruction set to the virtual machine, allowing much faster processing of certain calculations. This is ensured by enabling the 'host CPU' setting as can be observed below.

For example, an end-user may have a virtual machine of size 20GHz and 32GB of RAM. For an application that benefits from parallel processing, the number of virtual cores can be set to 20 thus giving twenty CPU threads of 1GHz each. On the other hand for an application that's bound by core speed, the virtual core number would be set to eight, giving eight threads of 2.5GHz each. Therefore, small changes can have a profound impact on performance.

Figure 8 includes a screenshot from CloudSigma's provisioning portal showing some of these power tweaking tools.

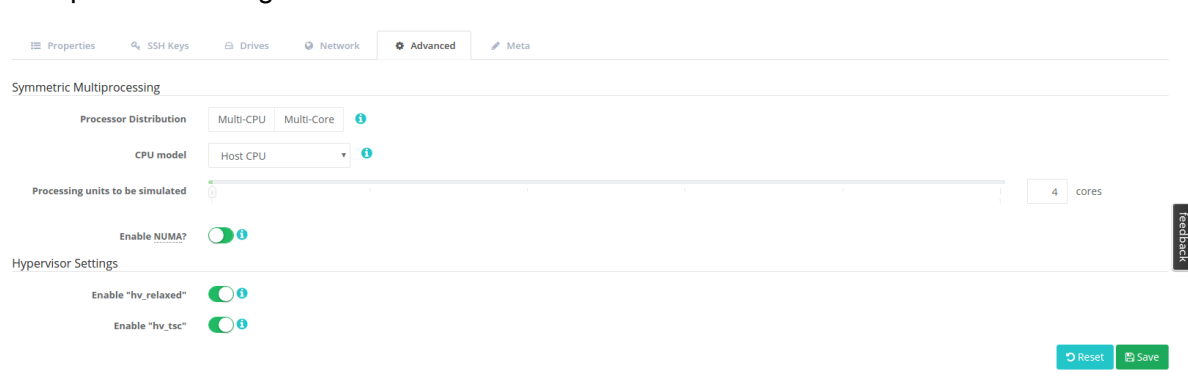


Figure 9: CloudSigma power tweaking tools

Specifically, the following can be configured under the Advanced tab:

Processor distribution - This determines whether the symmetric multiprocessing units set for the server are distributed and exposed as CPU cores or CPU sockets. It is recommended to set this to Multi-Core with Windows (due to licensing requirements for Multi-CPU) and to Multi-CPU for Linux distributions using NUMA (see below).

D4.1 COLA development testbed infrastructure

CPU model - The virtual CPU model (Host CPU or KVM64) can be selected for mitigating compatibility issues between the guest operating system and the underlying host's CPU. By default, all of the hypervisor's CPU capabilities are passed directly to the virtual machine.

Processing units to be simulated - The number for cores can be manually set.

NUMA - With Non-Uniform Memory Access, the memory access time depends on the memory location relative to a processor. NUMA can be enabled for servers with more than six processing units. If it is enabled for Linux distributions, it is recommended to use a Multi-CPU distribution.

hv_relaxed - This is a hypervisor setting, which enables relaxed timing for the CPU. hv_relaxed should be enabled for Windows, as it considerably increases performance. However, for Linux distributions, this should be disabled.

hv_tsc - This hypervisor setting enables the Time Stamp Counter to be passed through from the host to the server. hv_tsc" should be enabled for Windows, as it considerably increases performance. It should be disabled for Linux distributions.

At an account level, account administrators are able to assign specific access and control rights over certain account related operations using access control policies. This allows one account to grant certain rights to another account (or user). This is done through a 'labeling' feature. The account administrator may for example label a set of servers 'project x' and through an access control policy grant full access to 'project x' labelled infrastructure to one or more other users. On the other hand, the account administrator might wish to grant read-only access to billing information for the accounting department. The flexibility and powerful possibilities achieved by pairing labeling with ACLs is clearly evident.

Virtual machines (VMs) can be provisioned in a matter of seconds with a high degree of flexibility and control. The average provisioning time for a new cloud server/VM is 15 seconds. This is critical for running end-user facing web services that need to flexibly provision new resources in line with fluctuating end-user demand.

Users have the option of full API access with all account actions available, i.e. 100% API coverage, allowing complete automation and remote infrastructure monitoring, or the option of a feature-rich, yet intuitive web browser based GUI. The CloudSigma WebApp has been designed to allow easy resource management via any web browser. CPU and RAM can be specified to the nearest MHz and MB.

D4.1 COLA development testbed infrastructure



The screenshot shows a configuration interface for a 'Test Machine'. It includes tabs for Properties, SSH Keys, Drives, Network, Advanced, and Meta. The configuration details are as follows:

- Name:** Test Machine
- Description:** (Empty text area)
- Hypervisor:** KVM
- CPU Type:** Intel
- CPU:** A slider set to 3 GHz, with a scale from 0.25 to 100.
- RAM:** A slider set to 4 GB, with a scale from 0.25 to 128.
- Optimize for:** A dropdown menu set to 'Linux VM'.
- Cloudinit:** A checkbox that is checked.

Figure 10: Screenshot showing the custom server configuration screen

VM provisioning is achieved via a simplified wizard or custom server creation tool. The end-user can choose from a wide selection of ready system images including a number of BSD, Linux and Windows based operating systems as well as being able to quickly and easily upload their own ready ISO image. End-users can customise these marketplace images using cloud initialisation frameworks such as cloudinit (see <https://help.ubuntu.com/community/CloudInit>), allowing them to take a standard installation and contextualise/customise it to their specific requirements on-the-fly first boot-up.

6.2.4.1 Resource allocation and account creation

CloudSigma has allocated dedicated cloud resources for use by project partners for the duration of the project in accordance with the Grant Agreement (100Ghz CPU, 100GB RAM, 1500GB SSD). The agreed subscription (@ zrh.cloudsigma.com) was applied to the COLA associated CloudBroker account and the aforementioned resources applied. As already stated above, some partners required individual test accounts and subaccounts until complete functionality was made available via CloudBroker. Temporary accounts were setup for CloudSME, Saker Solutions, Inycom, UoW and Brunel University. Resources associated with these accounts (approx. 50Ghz CPU, 35GB RAM, 3000GB SSD at the time of writing) are provided in-kind by CloudSigma. An additional account was created for SZTAKI for testing if Occopus still runs optimally on CloudSigma. The following resources were provided: 2GHz CPU, 2GB RAM, 100GB SSD storage, all for the period of 6 months. Again, these resources were provided in-kind by CloudSigma, who also provided technical support throughout the process.

Due to the need for additional accounts outside of the CloudBroker Platform a script has been created to monitor resource usage across all COLA related accounts, including resources combined by CloudBroker. The script runs every three days and logs individual usage and calculates the aggregate. The script outputs to a spreadsheet and is time-stamped. If one or more resource type exceeds the total agreed in the Grant Agreement an

D4.1 COLA development testbed infrastructure

email alert is sent to CloudSigma staff (e.g. Resource SSD exceeds set threshold of 1500GB. Current usage 3010GB). However, it should be noted that CloudSigma will allow for some over-use as long as it is not excessive. Total usage will be assessed periodically and a strategy for dealing with excessive over-use devised on a case-by-case basis in cooperation with the other infrastructure providers within WP4.

6.2.4.2 Preliminary testing of additional MiCADO components

Occopus, a hybrid orchestration tool for configuring and orchestrating distributed applications on single or multi-cloud environments, has been tested on CloudSigma's cloud and it has been concluded that it is now running optimally. A test account was created for SZTAKI for the purpose of testing, consisting of 2Ghz CPU, 2GB RAM and 100GB of SSD storage for the period of 6 months. As Occopus will be a core component in the MiCADO framework, CloudSigma, along with other partners were asked to review the application and its integration within the testbed infrastructure. Occopus features tutorials specifically compiled for different cloud stacks. CloudSigma reviewed and tested each tutorial, checked for inconsistencies and sent feedback to SZTAKI. As part of the wider testing, CloudSigma also identified potential problems, debugged several issues that occurred during testing and offered some suggestions for improvement to the responsible developers at SZTAKI. Occopus will soon be included in the CloudSigma Partnership page of the provisioning portal / WebbApp and will be featured in a blog post on the CloudSigma marketing website.

6.2.5 Supplementary commercial cloud resources (Amazon Cloud)

General overview: The AWS Cloud provides a broad set of infrastructure services, such as computing power, storage options, networking and databases that are delivered as a utility: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 90 AWS services are available. New services can be provisioned quickly, without upfront capital expense. This allows enterprises, start-ups, small and medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements. (Sajee, 2017)

Due to a number of enhanced features such as high availability, secureness, flexible billing it was decided to use AWS Cloud as a supplementary commercial cloud resource for COLA tests and experiments.

Configuration and setup: To make the AWS Cloud available for the COLA partners, a specific AWS cloud account has been registered by CloudBroker GmbH. The account credentials then were provided to the COLA Prototype Platform to connect the Amazon resource to it. Thus, all the activities of the project onto AWS Cloud are done under this account; all the related costs are tracked there as well.

7 Conclusion and next steps

We have progressed as planned and can report the successful deployment of a development testbed infrastructure for the project. Some early issues were resolved quickly

Work Package WP4

D4.1 COLA development testbed infrastructure

and easily due to good communication between the partners participating in WP4. We will continue to assess total resource usage on CloudSigma's commercial cloud and resolve issues on an ad hoc basis. The script for monitoring and calculating total usage will be extended if necessary. In the coming weeks WP4 will concentrate on requirements gathering and performance benchmarking of microservices. A benchmarking methodology will be defined and distributed among project partners. We will then be able to make recommendations based on the requirements and the tuning and optimisations available to them. Results will be published in D4.2. Finally, support will be provided to services owners on an ongoing basis as we move from providing a development testbed to production infrastructure that replicates real world deployment.

8 Annex

8.1 Cloud characteristics and capacity summaries

In the table below we include the summarised cloud characteristics and capacity compiled for the spreadsheet. Cloud operators are responsible for keeping the spreadsheet up-to-date and accurate throughout the duration of the project.

CloudSigma Swiss cloud (zrh.cloudsigma.com)	
Testbed description	Commercial IaaS platform in Zurich (Zrh). The platform combines a proprietary stack with open-source technologies to provide a utility approach to IaaS provisioning. The platform offers high level of control and flexibility in the provision of computational power, RAM, storage (SSD as well as conventional magnetic), and networking.
General testbed configuration	
Hypervisor	KVM
IaaS stack / version	Proprietary CloudSigma stack
VM monitoring	Intra-VM testing tools, at the discretion of the VM owner, NewRelic third party integration
Access methods	API via https
Connectivity	Internet, VPN, Secure Remote User Access, Direct private patch to local switch
Cloud-Interface	
Provisioning	API, API middleware, Web console, Python library (Pycloudsigma). API documentation found here. https://cloudsigma-docs.readthedocs.io/en/latest/
Cloud integration / drivers	OpenStack HEAT, CloudInit, Apache Libcloud, JClouds, Fog, Ansible, Abiquo Hybrid Cloud, pycloudsigma Library
Networking	API, Web Console
CloudBroker integration	Integrated (up-to-date)
Compute capacity	
CPU (Ghz/core)	2.3Ghz
CPU (Total available to the project)	100Ghz

D4.1 COLA development testbed infrastructure

RAM (GB / VM)	128GB
RAM (Total available to the project)	100GB
Number of VMs	-
Provisioning speed for Windows VM (speed)	2 sec
Provisioning speed for Linux VM (speed)	2 sec
Storage	
Available storage interfaces	Volume storage
Image format	RAW
SSD capacity (GB)	8TB per image, subject to availability. More than 1 image can be mounted per VM
HDD capacity (GB)	-
Total storage available to the project	1500GB SSD
Networking	
Max internal network bandwidth per VM (Gb)	20
Max external network bandwidth per VM (Gb)	10
Max inter-VM latency (ms)	1
Total cloud external network bandwidth (Gb)	10+
Other	
Other relevant components	<p>CloudSigma offers full use of the underlying CPU instruction set, along with any/all optimisations available. NUMA, SIMD, SSE etc. technologies are made available to the VMs.</p> <p>Any x86 compatible OS can be deployed on our infrastructure and full root rights are afforded to users.</p>

D4.1 COLA development testbed infrastructure

	<p>Compute, RAM and Storage are offered independently and are not bundled in any way. Drive images can be attached to more than one VM or none at all. The images are persistent, even when not attached to a VM.</p> <p>CloudSigma does not place upper limits on internal and external data throughput.</p>
Issues	None

SICS	
Testbed description	Research IaaS platform in Kista, Sweden; OpenStack deployment
General testbed configuration	
Hypervisor	KVM
IaaS stack / version	OpenStack Ocata
VM monitoring	-
Access methods	OpenStack API v3
Connectivity	NOVA API or Horizon web interface
Cloud-Interface	
Provisioning	OpenStack API v3
Cloud integration / drivers	OpenStack
Networking	Web Console, API
CloudBroker integration	As of 02/03/2017:
Compute capacity	
CPU (Ghz/core)	3.7
CPU (Total available to the project)	32
RAM (GB / VM)	Depends on flavor
RAM (Total available to the project)	128

D4.1 COLA development testbed infrastructure

Number of VMs	50 project quota
Provisioning speed for Windows VM (speed)	-
Provisioning speed for Linux VM (speed)	-
Storage	
Available storage interfaces	swift (object storage), cinder (volume storage)
Image format	QCOW2, RAW
SSD capacity (GB)	-
HDD capacity (GB)	2000
Total storage available to the project	2000
Networking	
Max internal network bandwidth per VM (Gb)	20
Max external network bandwidth per VM (Gb)	20
Max inter-VM latency (ms)	1
Total cloud external network bandwidth (Gb)	20
Other	
Other relevant components	-
Issues	<p>Regarding CloudBroker integration requirements; Conclusion is that SICS cloud supports v3, while scaletools supports v2.</p> <p>Scaletools will investigate implementing support for v3.</p>

D4.1 COLA development testbed infrastructure

Testbed description	Research IaaS platform in London; OpenStack deployment
General testbed configuration	
Hypervisor	KVM
IaaS stack / version	OpenStack Juno
VM monitoring	-
Access methods	EC2 and S3 APIs (Amazon compatible), NOVA API or Horizon web interface
Connectivity	ssh / http / https - other methods and protocols available but require further configuration of firewall.
Cloud-Interface	
Provisioning	Web console, CloudBroker, API.
Cloud integration / drivers	CloudInit, OpenStack HEAT, EC2
Networking	Web Console, API
CloudBroker integration	Integrated (prototype platform)
Compute capacity	
CPU (Ghz/core)	160 CPUs
CPU (Total available to the project)	100 CPUs
RAM (GB / VM)	62.9 GB, 94.4 GB (node capacity thus less for a VM to run as systems need ram to run)
RAM (Total available to the project)	100GB
Number of VMs	50 project quota
Provisioning speed for Windows VM (speed)	-
Provisioning speed for Linux VM (speed)	-
Storage	

D4.1 COLA development testbed infrastructure

Available storage interfaces	swift (object storage), cinder (volume storage)
Image format	QCOW2, RAW
SSD capacity (GB)	-
HDD capacity (GB)	2000GB
Total storage available to the project	No quota defined
Networking	
Max internal network bandwidth per VM (Gb)	20
Max external network bandwidth per VM (Gb)	20
Max inter-VM latency (ms)	1
Total cloud external network bandwidth (Gb)	20
Other	
Other relevant components	<p>The Storage is not restricted for the project. UoW does not place upper limits on internal and external data throughput.</p> <p>Object storage system is Ceph through rados gateway which provides Swift and S3 compatible API interface.</p>
Issues	Storage problem if the output data is more than 50GB (with jobs run through CloudBroker platform)

SZTAKI

Testbed description	Research IaaS platform in Budapest; OpenNebula deployment.
General testbed configuration	
Hypervisor	KVM
IaaS stack / version	OpenNebula 5.2
VM monitoring	Zabbix

D4.1 COLA development testbed infrastructure

Access methods	EC2
Connectivity	-
Cloud-Interface	
Provisioning	EC2
Cloud integration / drivers	EC2, CloudInit
Networking	N/A
CloudBroker integration	Not yet available.
Compute capacity	
CPU (Ghz/core)	1.6 (AMD Opteron(™) Processor 6376)
CPU (Total available to the project)	-
RAM (GB / VM)	-
RAM (Total available to the project)	-
Number of VMs	-
Provisioning speed for Windows VM (speed)	-
Provisioning speed for Linux VM (speed)	-
Storage	
Available storage interfaces	EC2 -> volume storage
Image format	RAW
SSD capacity (GB)	-
HDD capacity (GB)	-
Total storage available to the project	-
Networking	
Max internal network	1

D4.1 COLA development testbed infrastructure

bandwidth per VM (Gb)	
Max external network bandwidth per VM (Gb)	1
Max inter-VM latency (ms)	1
Total cloud external network bandwidth (Gb)	10
Other	
Other relevant components	-
Issues	-