An innovative PaaS solution to support Big Data Analytics and Workflow management via Galaxy

G. Donvito, M. Antonacci, V. Spinoso, S. Nicotri,
F. Zambelli, M.A. Tangaro

Conferenza Annuale di Lifewatch Italia
Roma 25-27 Giugno 2018

Outline

Motivation

Service architecture

Galaxy production environment

Reference data availability

Storage encryption

Automatic elasticity

Conclusions and outlook

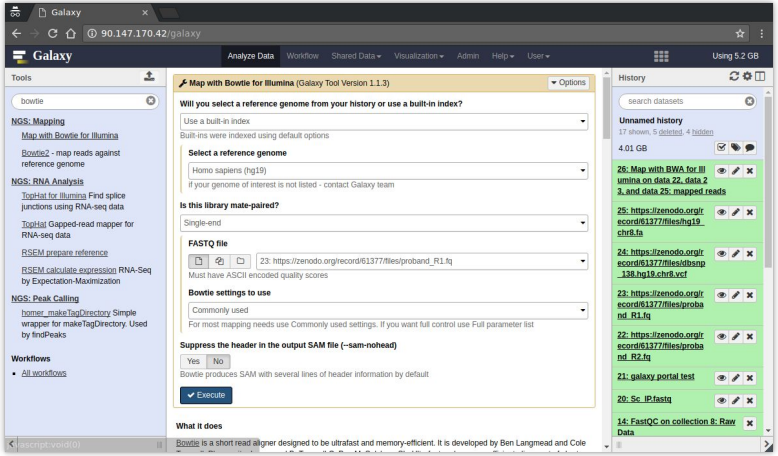
Motivation

Galaxy is a workflow manager adopted in many life science research environments in order to facilitate the interaction with bioinformatics tools and the handling of large quantities of biological data.

Through a coherent work environment and an user-friendly web interface it organizes data, tools and workflows providing reproducibility, transparency and data sharing functionalities to users.

Galaxy instances can be deployed in three ways, each one with pros and cons:

- public servers;
- local servers;
- commercial cloud solutions.



Motivation

	Ready to use	Quota	Galaxy Custom.	Maintenance	Costs	Data Privacy
Public Servers	✓	✗ Strongly Limited	✗	✓ Up to service provider	✓ No costs (usually)	✗
Local Install	✗	✓	✓	✗ Required	✗ Costly	✓
Cloud* (e.g. Amazon)	✓	✓ Costs Dependent	✓	✓ Only Galaxy Maintenance	✗ Costly	✗ ✓

(*) **Over 2400 Galaxy cloud servers launched in 2015** (Nucleic Acids Research (2016) doi: 10.1093/nar/gkw343)

Motivation

ELIXIR-Italy in the framework of the INDIGO-DataCloud project has developed a cloud Galaxy instance provider, allowing to fully customize each virtual instance through a user-friendly web interface, overcoming the limitations of others galaxy deployment solutions.

In particular, our goal was to develop a PaaS architecture to automate the creation of Galaxy-based virtualized environments exploiting the software catalogue provided by the INDIGO-DataCloud community.

The **INDIGO-DataCloud** project (H2020-EINFRA-2014-2) aimed to to develop an open source computing and data platform, targeted at multi-disciplinary scientific communities, provisioned over public and private e-infrastructures.

https://www.indigo-datacloud.eu/

www.indigo-datacloud.eu/service-component

INDIGO-DataCloud started in April 2015 and ended in September 2017.

Service architecture

Integrating different INDIGO-DataCloud technologies to automatically deploy a ready-to-use Galaxy production environment.

All Galaxy required components automatically deployed (Orchestrator and IM).

User friendly access, allowing to easily configure and launch a Galaxy instance (INDIGO FutureGateway portal).

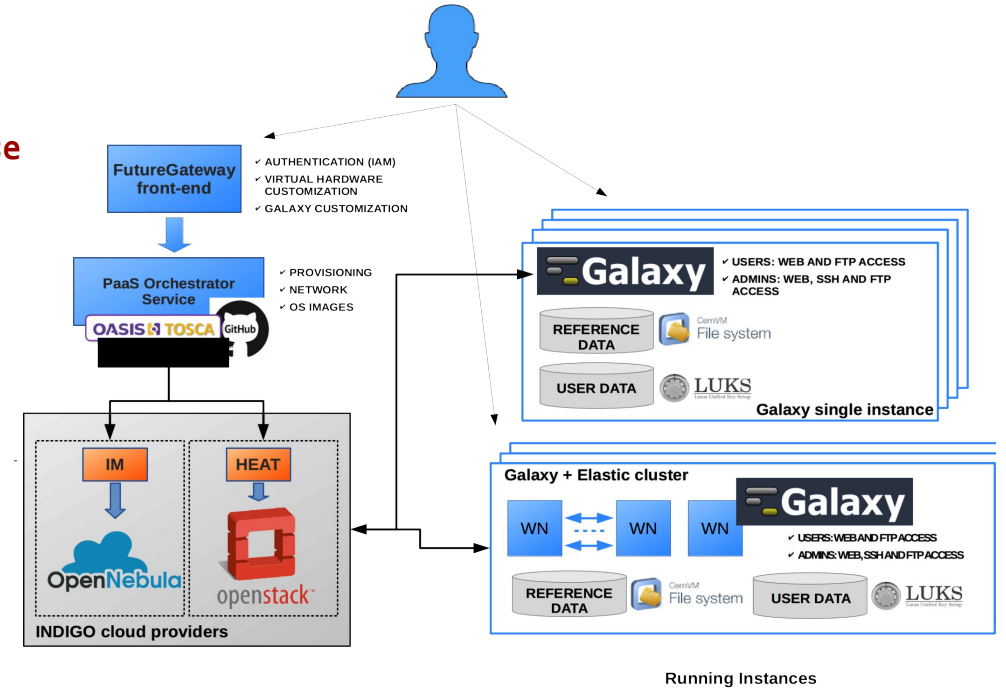
Authentication (IAM and FutureGateway).

Tools (Orchestrator and IM).

Reference data availability (CernVM File System).

Persistent storage, data security and privacy (IaaS block storage with/without filesystem encryption).

Cluster support with automatic elasticity (INDIGO CLUES).



Galaxy production environment

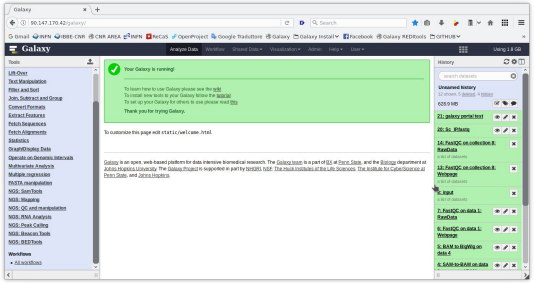
Galaxy is deployed for a multi-user production environment, i.e. there are some additional auxiliary application needed for the best performance (the basic Galaxy installation is suitable for development by a single user):

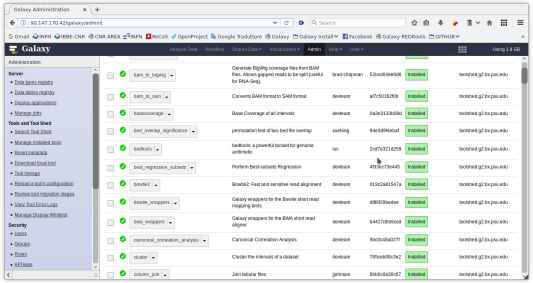
- PostgreSQL as database
- NGINX as web server (+ upload module)
- uWSGI link between the service and the web server
- Proftpd as FTP server

Each Galaxy instance is customizable, through the web front-end, with different sets of pre installed tools (e.g. SAMtools, BamTools, Bowtie, RSEM, etc...), exploiting CONDA as default dependency resolver and YAML recipe.

Current available tools presets:

- galaxy-no-tools
- galaxy-rna-workbench
- galaxy-epigen
- galaxy-testing (for test purpose)





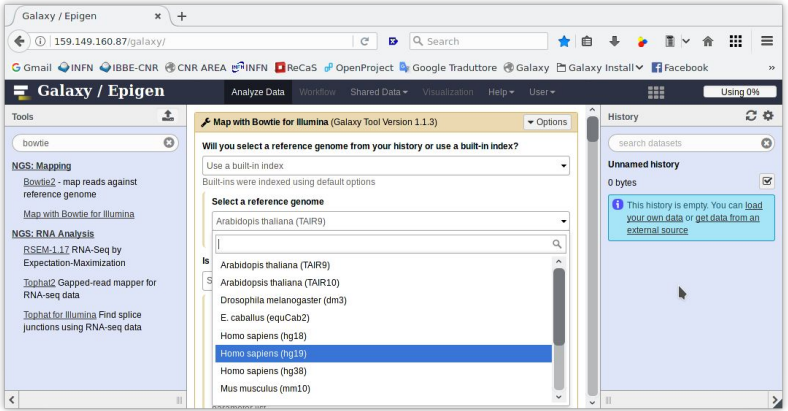
Reference data availability

Many Galaxy platform tools rely on the presence of reference data, such as alignment indexes or reference genome sequences, to efficiently work.

Each instance comes with reference data (e.g. genomic sequences) already available for many species, shared among all the instances through the CERN-VM FileSystem (cernvm.cern.ch) technology, thus avoiding unnecessary and costly data duplication.

Galaxy automatically is configured to properly use them.

Recipes and documentation to automatically setup your own CVMFS server and ship your reference data.



Storage encryption

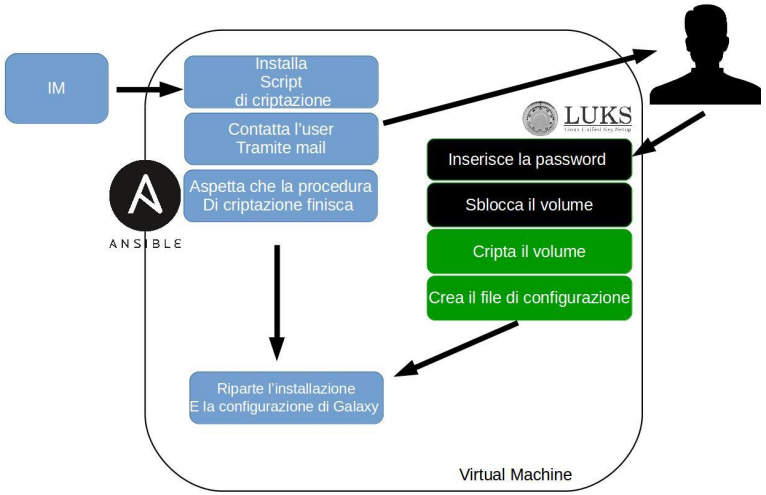
While the adoption of a distributed environment for data analysis makes data difficult to be tracked and identified by a malevolus attacker, full data anonymity and isolation is still not granted.

Data privacy is granted through **LUKS storage encryption** as a service: Users will be required to insert a password to encrypt/decrypt data directly on the virtual instance during its deployment, **avoiding any interaction with the cloud administrator(s).**

A notification mail is, sent to users describing how-to log into the VM and encrypt/decrypt the system.

User is only asked to insert their alphanumeric password 3 times:

1. Set password
2. Confirm password
3. Open LUKS volume.



Block Storage Encryption

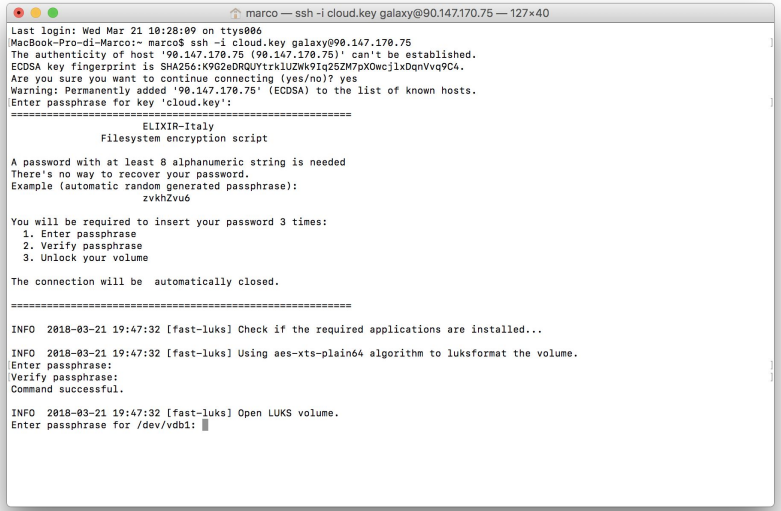
Automatic logout after password injection: the encryption procedure continues in background.

Default encryption algorithm:

- aes-xts-plain64 encryption
- 256 bit key
- sha256 as hash algorithm used for key derivation.

Script to easily manage the LUKS volume is added to each virtual instance:

- check if the volume is correctly mounted,
- Mount and open LUKS volumes.
- Close and umount LUKS volumes.



Test scenario (see backup slides):

- Volume not mounted. Impossible to access to its content.
- Volume opened and mounted, Galaxy running. Impossible to read data, even using cloud controller.

Automatic Elasticity

Virtual clusters through a dedicated section of the web front-end: allows to instantiate Galaxy with SLURM as Resource Manager and to customize the number of virtual nodes, nodes and master virtual hardware.

Automatic elasticity, provided using the Infrastructure Manager and CLUES service components, enables **dynamic cluster resources scaling,** providing an efficient use of the resources, making them available only when really needed.

New working nodes are powered-on, depending on the cluster workload and powered-off when no longer needed:

- Each Galaxy tools preset installed with Galaxy have been tested to work with SLURM elastic Cluster.
- Each node is configured according to the Galaxy tools installed on the VM as selected by the user during the configuration phase.

Conclusion and outlook

Our service aims to provide the Galaxy workflow manager to end users ranging from small research groups to institutions or SMEs, on suitable computation resources, removing the need to maintain their own hardware and software infrastructure and using resources in a more efficient way, ensuring improved reliability, better performances and the capability to handle larger research activities exploiting the features of the INDIGO-Datacloud components, opening the route for the migration of public Galaxy instances to this service.

INDIGO ended in September 2017, but the development of the services exploited as backend for this service continues in H2020 project Deep-HybridDataCloud and eXtreme-DataCloud:

- Support the transparent access to specialized computing hardware (GPUs, Infiniband, etc.) and HPC resources.
- Improve the workflow for hybrid deployments

Future improvements:

- Deployment of dockerized Galaxy and jobs on Mesos clusters.

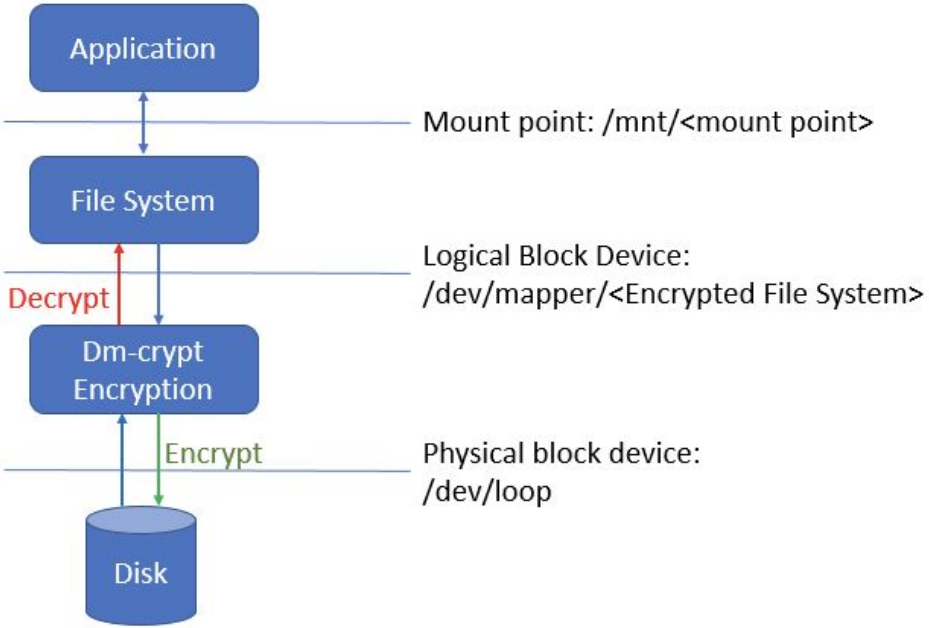
Thank you!

CONTACTS:

- **Marco Antonio Tangaro (CNR-IBIOM) ma.tangaro@ibiom.cnr.it**
- **Federico Zambelli (ELIXIR-ITA technical coordinator) f.zambelli@ibiom.cnr.it**
- Giacinto Donvito (INFN) giacinto.donvito@ba.infn.it
- Graziano Pesole (head of ELIXIR-ITALY Node) g.pesole@ibiom.cnr.it

Backup

Block Storage Encryption



Block Storage Encryption

Bash scripting + Ansible + INDIGO PaaS Orchestrator:

- Storage Encryption as a Service
- Dependency resolution
- Script instance lock, i.e. is not possible to run two instances of the encryption script.
- Configurable (encryption algorithm, key size, hash algorithm, mountpoint, filesystem).
- Automatic configuration file creation to open/close the volume with one command.

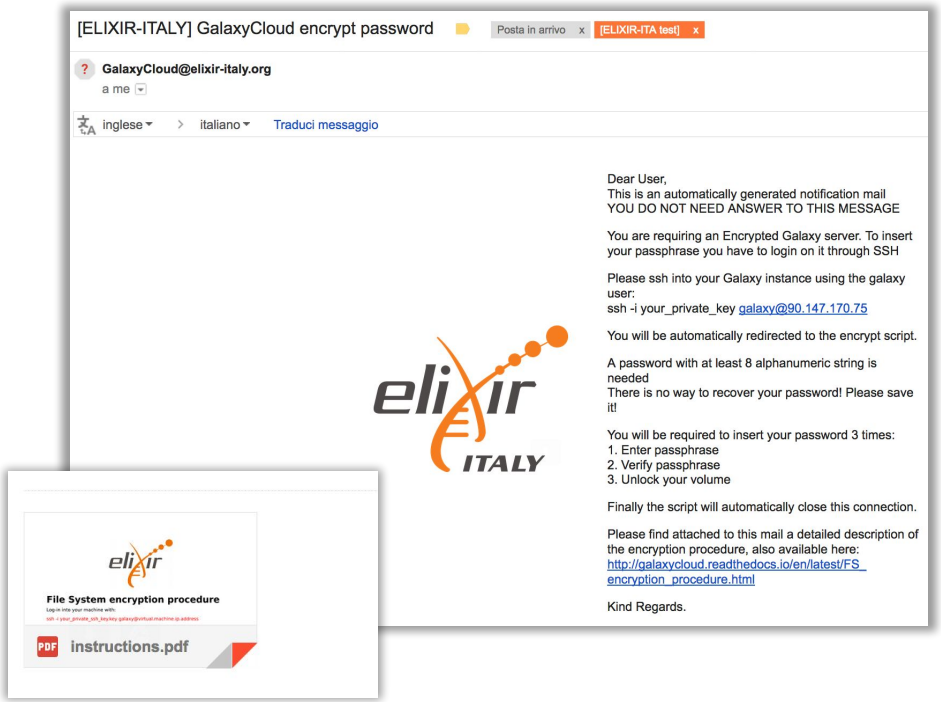
Block Storage Encryption

Ansible automates the encryption procedure, installing the scripts, informing, by mail, the user once the system is ready to accept the password.

The encryption procedure summary is reported by mail, while a detailed step-by-step how-to is sent attached.

Script to easily manage the LUKS volume is added to each virtual instance:

- check if the volume is correctly mounted,
- Mount and open LUKS volumes.
- Close and umount LUKS volumes.



Block Storage Encryption

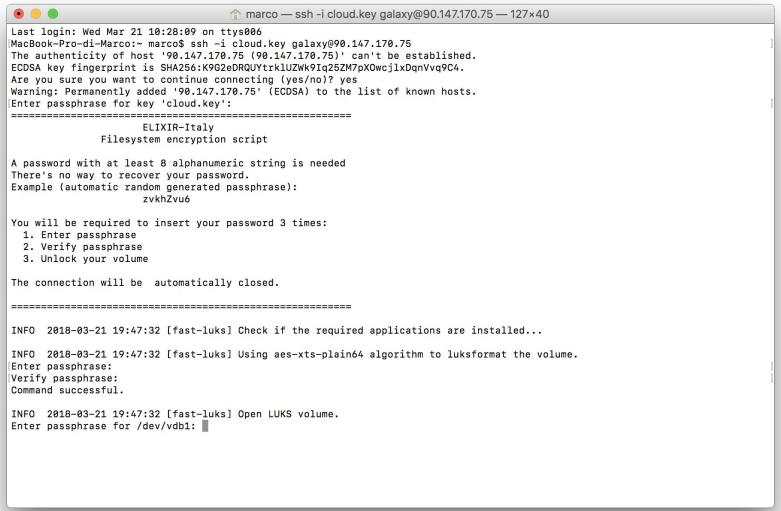
Automatic logout after password injection: the encryption procedure continues in background.

Default encryption algorithm:

- aes-xts-plain64 encryption
- 256 bit key
- sha256 as hash algorithm used for key derivation.

Script to easily manage the LUKS volume is added to each virtual instance:

- check if the volume is correctly mounted,
- Mount and open LUKS volumes.
- Close and umount LUKS volumes.



Block storage encryption

- Test on unmounted encrypted devices:
  - Create two volumes, one encrypted
  - Put inside the same file
  - Umount volumes
  - Create volume binary images and HexDump the binary image with xdd
  - Grep non-zero bytes and search for the file content

  It is possible to see the file content only on the un-encrypted volume.

- Try to open the volume when active (LUKS volume opened and mounted, Galaxy running) in the Virtual Machine.

  Test executed on the cloud controller as administrator.

  It is not possible to mount the volume without the user password.

Automatic elasticity

ELIXIR-IIB: Galaxy as a Cloud Service