Laniakea: an open solution to provide Galaxy “on-demand” instances over heterogeneous cloud infrastructures.

Marco Antonio Tangaro[1], Giacinto Donvito[2], Marica Antonacci[2], Matteo Chiara[3], Pietro Mandreoli[3], Graziano Pesole[1,4], Federico Zambelli[1,3]

1. Istituto di Biomembrane, Bioenergetica e Biotecnologie Molecolari (IBIOM), Consiglio Nazionale delle Ricerche (CNR), Via Amendola 165/A, 70126 Bari, Italy
2. Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Bari, Via Orabona 4, 70126 Bari, Italy
3. Dipartimento di Bioscienze, Università di Milano, via Celoria 26, 20133 Milano, Italy
4. Dipartimento di Bioscienze, Biotecnologie e Biofarmaceutica, Università di Bari, Via Orabona 4, 70126 Bari, Italy

Abstract

Background

Galaxy is rapidly becoming a de facto standard among workflow managers for bioinformatics thanks to its rich feature set, overall flexibility, and a thriving community. One of the main advantages of Galaxy consists in making complex analyses, e.g. involving numerous and large data sets, accessible even to users lacking computer proficiency, while at the same time improving results reproducibility and easing teamwork and data sharing among researchers. Currently, many Galaxy public services are available but there still exist situations in which a private Galaxy instance constitutes a preferable alternative, for example, scenarios involving heavy workloads, data privacy concerns or particular instance customization needs. In those cases, a virtual Galaxy instance can represent a viable solution that avoids the typical burdens of managing the local hardware and software infrastructure needed to run and maintain a production-grade Galaxy service.

Results

We present a robust and feature-rich software suite called Laniakea, ready to be deployed on any scientific or commercial Cloud infrastructure in order to provide a “Galaxy on demand” Platform as a Service (PaaS). Laniakea lays its foundations on the INDIGO-DataCloud middleware that has been developed targeting a large number of scientific communities and is therefore deployable on multiple hardware and provisioned over hybrid (private or public) e-infrastructures. The end user interacts with Laniakea through a front-end that allows a general setup of the Galaxy instance, then Laniakea takes charge of the deployment both of the virtual hardware and of all the software components, finally providing a production-grade, but still fully customizable, Galaxy virtual instance. Laniakea’s many features include support to the deployment of plain or cluster backed Galaxy instances, shared reference data volumes, encrypted user data volumes and rapid development of novel Galaxy flavours, that is Galaxy configurations tailored for specific tasks, through Ansible recipes. As a proof of concept, we provide a demo Laniakea instance hosted at an ELIXIR-IT Cloud facility.

Conclusions

The migration of scientific computational services towards virtualization and e-infrastructures is one of the most visible trends of our times. Laniakea provides Cloud administrators with a ready-to-use software suite that enables them to offer Galaxy, a popular workflow manager for bioinformatics, as an on-demand PaaS. We think that Laniakea can concur in making the many advantages of using Galaxy more accessible to a wider user base by removing most of the burdens involved in running a private instance. Furthermore, Laniakea's design has been imprinted to generality and modularity and could, therefore, be easily adapted to support different services and platforms beyond Galaxy.

Background

The recent improvements in our capacity to gather vast amounts of complex, multi-layered and interconnected biomolecular data demand a parallel development and enhancement of the computational tools that we employ to analyse and handle this wealth of information. On the other hand, the rapid proliferation of those tools can make the execution of complex bioinformatics workflows cumbersome due, among other things, to the existence of many different and incompatible data formats, long and convoluted command lines and the need of correctly handling input, output, and intermediate files. In turn, that not only makes harnessing the information contained in the biomolecular data unnecessarily onerous even for expert bioinformaticians but represents also a significant obstacle to reproducibility [1] as well as an intimidating barrier for biologists aspiring to explore their own data in autonomy [2], students [3], and health-care providers adopting *clinical bioinformatics* approaches within their medical protocols [4]. In the past few years, several workflow manager platforms for bioinformatics addressing those and other issues have been proposed (see [5] for a review). They usually provide integrated interfaces that deliver not only a more user-friendly work environment but improve results reproducibility, allow easier data sharing and enable collaborative data processing.

Galaxy

Among those, the Galaxy platform is one of the most successful examples, having gathered a vast and thriving community and providing a consistent, user-friendly, flexible, effective and customizable gateway to a vast array of bioinformatics software and analysis workflows [6]. The software consists of an open source server-side application that interacts with the remote user through a simple interface giving access to a wealth of tools for datasets handling and analysis, workflow design, results visualization and sharing with collaborators or publicly. While its use is mostly considered as genuinely convenient for the user, the deployment of a production-grade Galaxy instance requires the setup and maintenance of an elaborate collection of helper components (e.g. database management system, web server, load balancer, etc...) and of the full set of bioinformatic tools and reference data going to be supported by the instance itself. This, together with the necessity of an adequate IT infrastructure in order to properly support the Galaxy service for any non-trivial amounts of users or workloads, has usually restricted the role of Galaxy service providers to institutions or groups with suitable IT facilities and the appropriate technical know-how. At the time of writing, about ninety Public Galaxy instances do exist (https://galaxyproject.org/public-galaxy-servers/), serving a vast community of users [6]. While useful and popular, the public nature of those services implies some hardly addressable shortcomings like limited quotas for computing and storage resources, lacking customization options for the end-user and potential concerns for data security and privacy, a worry particularly noticeable when processing human sensitive data. In turn, this can limit or outright interdict the choice of using Galaxy public instances for some applications or users' categories, e.g. analyses requiring big or huge computing workloads or precision medicine researchers and operators.

Cloud solutions to Galaxy provision

The cloud computing model [7] is rapidly gaining popularity within the life sciences [8–11] and the biomedical [4,12,13] communities. Among other advantages, it offers a set of solutions and features that can eliminate or mitigate the drawbacks of Galaxy public instances described above. Actually, many efforts have already been put forward in this regard: Globus Genomics [14] provides a Galaxy-based bioinformatics workflow platform, built on Amazon cloud services, for large-scale next-generation sequencing analyses, CloudMan [15,16] allows individual researchers to deploy Galaxy instances relying on arbitrarily sized compute cluster on the Amazon cloud infrastructure (it can also support OpenStack and OpenNebula through custom deployments), the Genomic Virtual Laboratory (GVL) [17] offers Galaxy through a middleware available on Nectar, the Australian cloud infrastructure for research, and again on the Amazon cloud, while Krieger and colleagues describe a possible configuration stack to deploy Galaxy on an OpenStack based IaaS (Infrastructure as a Service) [18]. The appeal of Galaxy cloud solutions is also made evident by the 2016 Galaxy update [19] reporting that over 2,400 Galaxy servers were launched on the Amazon cloud in 2015 alone, pointing to a strong demand for ready-to-use but private virtual Galaxy instances. The Amazon Galaxy service [20] is, however, still a commercial solution that can prevent many researchers and research or healthcare facilities from employing it due to funding or budget issues, ethical concerns or legal requirements (e.g. EU's General Data Protection Regulation). At the same time, other solutions like access to GVL through the Nectar cloud is not usually within the reach of European researchers nor it provides an ecosystem easily integrable with other resources available to them through European e-infrastructures. We hereby describe a software framework for the provision of *on-demand* Galaxy instances, designed to work on existing scientific e-infrastructures by leveraging the open and modular middleware architecture developed within the INDIGO-DataCloud H2020 project [21,22] and built to be simply deployable and maintainable by infrastructure managers and convenient for end-users.

Methods

INDIGO-DataCloud middleware components

The heterogeneity of the currently available scientific cloud infrastructures often entails portability issues between the different technologies employed that can result in the lack of one or more key features, e.g. automatic elasticity or multi-clouds deployments. In turn, these resources are usually less efficient than commercial counterparts relying on more homogeneous environments. The INDIGO-DataCloud H2020 project, officially concluded in September 2017 and passing on its legacy to the EOSC-Hub, (https://www.eosc-hub.eu/), DEEP-HybridDataCloud (https://deep-hybrid-datacloud.eu/) and XDC (http://www.extreme-datacloud.eu/) projects, aimed at improving this scenario easing cloud e-infrastructures exploitation by scientific communities. INDIGO's software catalogue [21] tackles e-infrastructures heterogeneity through an inclusive support to open standards and solutions, e.g. adopting both OpenStack (https://www.openstack.org) and OpenNebula (https://opennebula.org) as Cloud Management Platforms (CMPs), and allowing to transparently deploy applications using either virtual machines (VMs) or Docker containers. This has been achieved pooling the common needs of a wide range of use cases and then taking advantage of the available open-source cloud components, adapting and/or enhancing them when needed to obtain the desired functionalities and embarking in the development of new software only when absolutely necessary. A complete overview of the INDIGO PaaS software components and the installation sequence needed to correctly deploy them are provided in Fig. 1. Table S1 (in Supplementary2.docx) collects the URLs of all the INDIGO components required by Laniakea.

Orchestrator and Infrastructure Manager

The INDIGO PaaS Orchestrator and the Infrastructure Manager (IM) components (https://github.com/indigo-dc/orchestrator) set-up the virtual infrastructure environment and deploy the software framework. They are based on the OASIS Topology and Specification for Cloud Applications (TOSCA) [23,24] open standard language. YAML syntax [25] is used to describe the cloud applications and services, together with their requirements and dependencies, regardless of the underlying platform. This enables the deployment of complex applications from small reusable building blocks, the "node types", and the topology of their relationships. The INDIGO PaaS Orchestrator coordinates the application deployment according to the specifications described in the TOSCA template, selecting the most suitable cloud infrastructure site among those available and delegating to the Infrastructure Manager (IM) (https://github.com/indigo-dc/im) the deployment and configuration of the virtual infrastructure on the target site. IM supports deployment on OpenStack, OpenNebula and commercial cloud providers alike, and serves as an abstraction layer for the definition and provision of the needed resources. Ansible roles (http://docs.ansible.com) are finally used for the software layout of the infrastructure's nodes, they instruct the automation engine on how to install and configure the end-user applications or services, like Galaxy, on bare OS images. All in all, this architecture allows for the deployment of a wide range of cloud agnostic applications and services

Authorization and Authentication Infrastructure

The INDIGO Identity and Access Management (IAM) is an Authentication and Authorisation Infrastructure (AAI) service that manages users' identities, attributes (e.g. affiliation and groups membership) and authorization policies for access to INDIGO based resources. IAM supports many token-based protocols, i.e. X.509 [26], OpenID Connect [27] and SAML [28], thus allowing different infrastructures to federate and the user to connect to federated PaaS and manage heterogeneous and distributed resources through just one AAI service. Furthermore, IAM allows researchers to use their account to access all INDIGO services at the same time, easing the overhead experienced by end-users in case of multiple INDIGO services running on the same infrastructure.

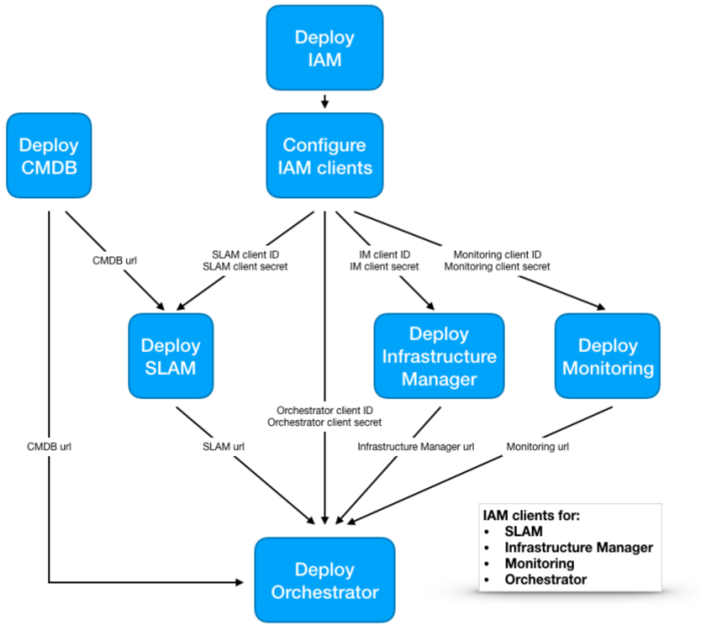


Fig. 1. INDIGO-DataCloud PaaS software components and deployment procedure. The ID and secret IAM clients are required during the configuration to allow the identification and authentication of the other connected services. After IAM configuration, the CMDB service is installed and configured, it will provide detailed information on the available cloud sites and the images and containers they support. The following steps are the deployment of SLAM, the Service Level Agreement Manager, and the IM and Monitoring stacks. Finally, the INDIGO PaaS-Orchestrator can be deployed.

Other INDIGO components

- The **CLUES** service is an elasticity manager for HPC clusters that enables dynamic cluster resources scaling, deploying and powering-on new working nodes depending on the workload of the cluster and powering-off and deleting them when no longer needed. Once new jobs are submitted to the cluster Resource Manager (RM) queue (e.g. SLURM [29] or Torque [30]), CLUES contacts the Orchestrator and starts the provisioning of available temporary resources, thus improving the overall efficiency of the infrastructure.

- **FutureGateway** is built on top of the Liferay open source portal framework (https://www.liferay.com) and provides GUI based portlets that interact with the PaaS layer, allowing the authentication through the integrated INDIGO IAM portlets and the customization of relevant parameters of the TOSCA templates by the user prior to dispatching them to the Orchestrator for deployment.

Encryption at block device level

The encryption layer is based on LUKS (Linux Unified Key Setup) [31] [https://gitlab.com/cryptsetup] that is the current standard for encryption on Linux platforms. It provides robustness against low-entropy passphrase attacks using salting and iterated PBKDF2 passphrase hashing. LUKS supports secure management for multiple user passwords, allowing to add, change and revoke passwords without having to re-encrypt the whole device. Supplementary1.docx contains a detailed description of the Laniakea encryption framework.

CernVM File System

CernVM File System (CVMFS) [32] is used to support common reference data repositories. This is a read-only POSIX file system originally developed to facilitate the distribution of High Energy Physics analysis software through HTTP protocol. Data files are hosted on any server and can be mounted concurrently on multiple compute nodes through a Linux filesystem module-based client (FUSE) that loads and caches only the small fraction of files needed on-demand. This solution is suitable for quasi-static files that have to be shared across several geographically distributed clusters, it also supports local caches to speed up read operations on the hosted data.

Results

The strategy underpinning Laniakea's architecture (Fig. 2) recognizes that a wide range of Galaxy users and instance types do exist. For example, users may stretch from plain account owners to instance administrators and from developers to training courses teachers. At the same time, Galaxy instances can have different footprints in terms of computational resources depending for example on typical usage or number of concurrent users; can be short-lived to satisfy temporary needs (e.g. a training course) or long-lived to provide a persistent work environment; public or private; needing advanced data security features in order to work on human sensitive data or not, etc… As a consequence, a Galaxy PaaS provider should be able to cover the requirements of the largest possible number of user/instance type combinations in order to maximize the convenience of its service.

Our development approach has therefore been aimed at providing the means to swiftly address these heterogeneous needs, empowering the end-user with a wide array of accessible customization options that can deliver from out-of-the-box, stable, production-grade Galaxy instances already configured with bioinformatics tools, reference data and cluster support, to blank-sheet Galaxy instances, ready to be tailored and personalized to meet the widest possible array of different needs.

Another key Laniakea's feature consists in the integration, for the first time at best of our knowledge on a Galaxy on-demand platform, of a built-in technology to encrypt storage volumes. This function can be used to provide strong data protection through state-of-the-art encryption protocols: the secure layer insulates stored sensitive data from unauthorized access both by malicious attackers or trusted users of the same cloud infrastructure, notably including the administrator(s) of the cloud and hardware layers themselves.

Finally, Laniakea supports reference data sharing via CVFMS and a high degree of instance scalability through an array of deployment configurations ranging from single node Galaxy instances to SLURM managed Galaxy clusters, including also a still preliminary support to cluster elasticity through CLUES. Table S2 (in Supplementary2.docx) collects the URLs of the components of Laniakea's software suite. In the following paragraphs, we describe in more detail Laniakea's components and features.

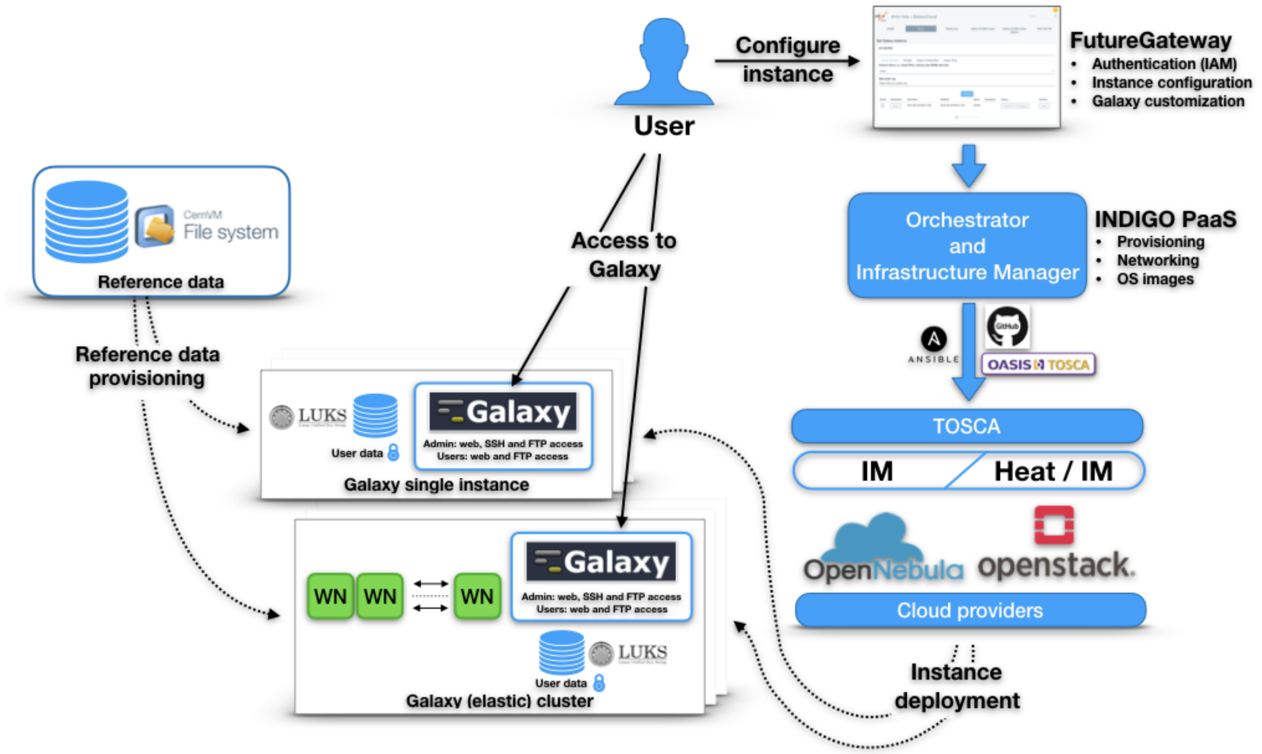
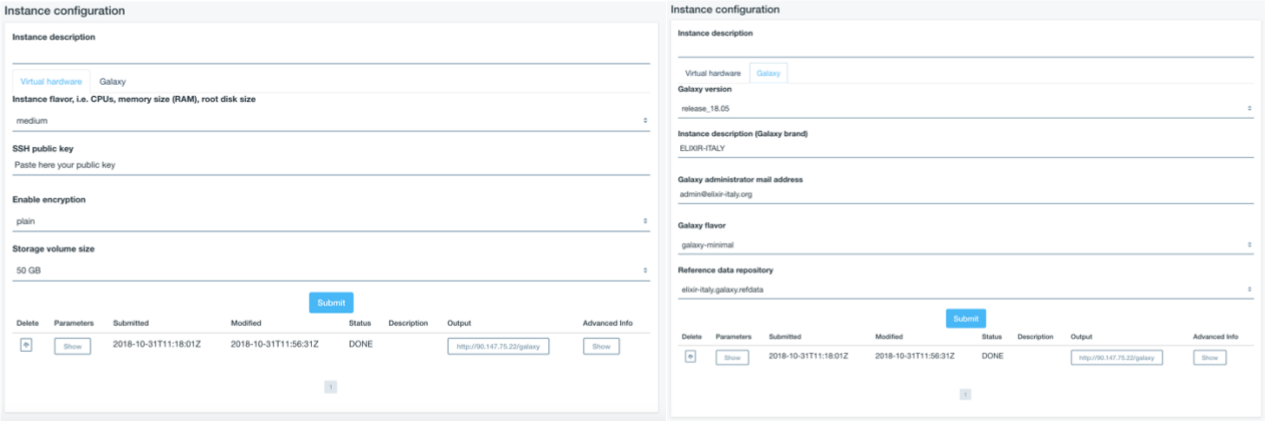


Fig. 2. Laniakea architecture and deployment flow. The FutureGateway portal provides users with the front-end to configure and manage Galaxy instances, the resulting TOSCA templates are sent to the PaaS layer that employs INDIGO services to deploy instances over the IaaS, that in turn provides virtual hardware, storage, and networking. Finally, the Galaxy instance (single or cluster backed) is configured with the requested flavour and attached to a plain or LUKS encrypted storage volume (depending on user's choice) and to the CernVM-FS shared volume with reference data. At the end of the process, the public IP address of the new Galaxy instance is provided to the user.

The web front-end

The web front-end (Fig. 2), built upon the INDIGO FutureGateway component, is accessible after authentication through INDIGO IAM and represents the entry point to the PaaS from a user's point of view. It provides the configuration interface needed to initialize and deploy any of the desired virtual hardware/Galaxy instance type combinations available. It provides distinct configuration panels for Galaxy single VM or cluster deployments.



Instance configuration

Instance description

Instance flavor, i.e. CPUs, memory size (RAM), root disk size: medium

SSH public key: Paste here your public key

Enable encryption: plain

Storage volume size: 50 GB

Delete	Parameters	Submitted	Modified	Status	Description	Output	Advanced Info
	Show	2018-10-31T11:18:01Z	2018-10-31T11:56:31Z	DONE		http://90.147.75.22/galaxy	Show

Instance configuration

Instance description

Galaxy version: release_18.05

Instance description (Galaxy brand): ELIXIR-ITALY

Galaxy administrator mail address: admin@elixir-italy.org

Galaxy flavor: galaxy-minimal

Reference data repository: elixir-italy.galaxy.refdata

Delete	Parameters	Submitted	Modified	Status	Description	Output	Advanced Info
	Show	2018-10-31T11:18:01Z	2018-10-31T11:56:31Z	DONE		http://90.147.75.22/galaxy	Show

Fig 3. The web interface is organised using different tabs for each configuration task. The “Virtual hardware” tab allows to configure the hardware, e.g. number of CPUs and quantity memory, storage size and to submit the user SSH public key that will grant access to the VM. For cluster deployments the same tab allows also to configure the number of worker nodes required and their hardware configuration. The “Galaxy” tab provides the software configuration options: Galaxy version, instance description, the mail address of the administrator, the flavour and the Reference data repository to be attached. The list of active instances and their status is available at the bottom of the page.

The instance configuration front-end is organized in two panels to guide the user through the selection of the most relevant features:

- The Virtual Hardware configuration tab exposes the array of available hardware setups, usually known as flavours, differing for the number of virtual CPUs and quantity of RAM. The user then provides its SSH public key that will grant access to the VM, once deployed, and picks the storage volume size and type, i.e. plain or encrypted. The Virtual Cluster deployment panel allows for the setup of both the front-end and worker nodes.
- The Galaxy Configuration allows to select the Galaxy release version among the supported ones, the reference data provider, the e-mail of the instance administrator and finally the *Galaxy flavour*, that is the pre-configured set of tools that will be ready to use straight away after deployment (more on this in the “Galaxy flavours” section).

The set-up defined through the interface is in turn submitted to the INDIGO Orchestrator that manages the deployment and configuration processes. Once the process comes to an end, a public IP address becomes available for the freshly minted Galaxy server and from that moment onwards the user has full administrator privileges over the instance.

The Galaxy environment

In general, the deployment of a basic Galaxy instance is a relatively simple task but setting up a Galaxy multi-user production environment is more complicated since many auxiliary software components, that in turn must be properly installed and configured, are required or recommended. Laniakea automatizes this lengthy and error-prone procedure by spawning on-the-fly virtual environments with the following configuration (Table 1): CentOS7 as the operating system (OS), PostgreSQL as the database engine, Nginx, uWSGI, and ProFTPD as the web, application and FTP servers respectively. Apart from the OS, for which there are no suggestions, this configuration is rooted in the recommendations made by the Galaxy Project itself for production environments. We chose CentOS 7 as the OS due to its well-known adherence to standards and the long span of the foreseen official support for this release (updates until June 30, 2024) however, Laniakea is compatible also with Ubuntu 16. Laniakea fully supports both virtual machines and Docker containers as virtual environments, the choice between one or the other solution is left to the preference of the IaaS administrator. The Laniakea's procedure for the on-the-fly setup of the virtual environment offers a greater degree of cloud agnosticism over pre-configured virtual machines and at the same time ensures that each new Galaxy instance makes use of the latest available release of software components. However, this approach comes with two possible drawbacks: first, the installation procedure takes time, about five hours on our test IaaS and second, there is a chance of some components failing to install due for example to the temporary unavailability of the corresponding on-line repositories. To overcome these limitations, Laniakea provides a backup procedure to instantiate fully pre-configured images that the IaaS administrator can easily obtain from the on-the-fly ones previously described. In this case, the procedure will adapt the configuration of a saved image to make it compatible with the virtual hardware and any other parameter (e.g. the instance administrator credentials) selected by the user. Apart from being more resistant to third parties' repositories unavailability, this allows faster deployments, i.e. less than an hour on our test IaaS, at the cost of a lesser degree of IaaS agnosticism.

Software component	Version installed by Laniakea
Operative System	CentOS 7 (supported Ubuntu 16.04)
Galaxy	release_17.05 and release_18.05
PostgreSQL	9.6
NGINX	1.12.2
uWSGI	2.0.17.1
PROFTPD	1.3.5e

Table 1. Laniakea's Galaxy production environment components summary.

Galaxy flavours

The Galaxy Tool Shed (https://toolshed.g2.bx.psu.edu/) is the official Galaxy tools repository and provides a convenient mean for an administrator to customize an instance with tools and workflows tailored for any expected typical use. However, despite the many improvements brought by the recent adoption of Conda (https://conda.io) for the management of packages and dependencies, the concurrent installation and configuration of many tools and in particular the ones involving many dependencies or relying on reference data can still be burdensome, requiring sometimes a low-level knowledge of the Galaxy environment and of the tools to be installed. For this reason, Laniakea can provide instance administrators with a handy set of domain-specific *Galaxy flavours*, that is cured collections of tools already installed, configured, tested, organised in workflows and ready to be used out-of-the-box. To access the target Galaxy instance and install a flavour Laniakea employs the official Galaxy Project library Ephemeris (https://github.com/galaxyproject/ephemeris) while YAML recipes are used to list the required packages. Through a helper Ansible role Laniakea can also fine-tune the configuration of packages and solve dependencies that for any reason are missing or malfunction upon Conda installation. All in all, this approach allows IaaS administrators to easily prepare additional flavours just by creating new YAML recipes and eventually tweak the resulting configuration using the helper Laniakea Ansible role.

Currently we provide five proof-of-concept flavours that are named “galaxy-minimal”, “galaxy-epigen”, “rna-workbench”, “GDC_Somatic_Variant” and “CoVaCS”. The first one is a bare instance providing just the starting tools embedded in any Galaxy installation and provides a clean foundation for administrators willing to start with a blank sheet ready to be customized to their own needs or for developers of new tools and flavours. The “galaxy-epigen” flavour is based upon the layout of the Italian Epigen Project’s (http://www.epigen.it/) Galaxy server (http://www.beaconlab.it/epigalaxy) and provides a selection of tools tailored for NGS data analysis with particular emphasis on ChIP-Seq and RNA-Seq. The “rna-workbench” flavour is based on [33] and includes more than 50 tools dedicated to RNA-centric analyses covering i.e. alignment, annotation, secondary structure profiling, target prediction, etc… The “GDC_Somatic_Variant” flavour is a port of the Genomic Data Commons (GDC) pipeline for the identification of somatic variants on whole exome/genome sequencing data (https://gdc.cancer.gov/node/246), the resulting Galaxy workflow is shown in Fig. 4 and includes many tools for which we needed to develop new Galaxy wrappers. Finally, the CoVaCS flavour makes available the workflow described in [34] for genotyping and variant annotation of whole genome/exome and target-gene sequencing data. All in all, these examples provide a fine proof-of-concept of the relative ease of generating Galaxy flavours for Laniakea.

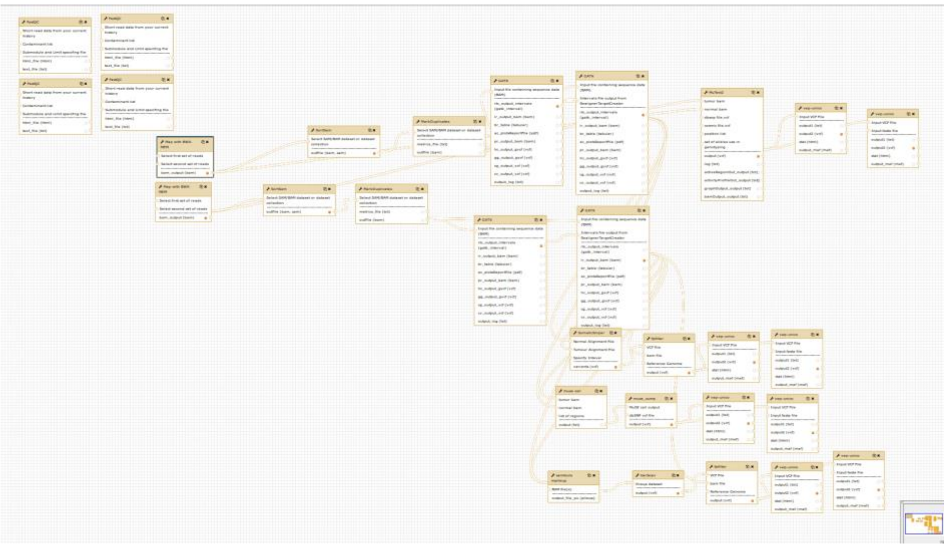


Fig.4. The GDC Somatic Variant workflow implemented as a Galaxy workflow for the corresponding flavour.

IaaS wide Reference data availability

A read-only CVMFS volume shared among all the instances on the same IaaS grants seamless access to reference data, thus avoiding useless data replication and favoring reproducibility of analyses. The IaaS manager can choose to provide its own CVMFS reference dataset, eventually by mirroring the one made available by the Galaxy Project [6], or to link a remote one. The latter solution comes the cost of transferring needed reference data from a remote service on-the-fly. To streamline the creation of new CVMFS repositories by the IaaS manager, we have made available a suitable Ansible role and documented the procedure. As a proof of concept, we provide three different repositories. The first is a basic, manually curated reference dataset maintained by us with the latest genome assemblies and corresponding indexes for Bowtie [35], Bowtie2 [36] and BWA [37,38] of human and four other model organisms (mouse, yeast, fruit fly and A. thaliana). The second is a repository tailored for variant calling in human, to use with the CoVaCS and GDC Galaxy flavours, and provides many VCF variant collections other than human genome assemblies and indexes. Finally, we also mirror the Galaxy project "by hand" reference repository. In principle, this approach could also be extended to optimize the use of storage resources and performances in case of a single user or group exploiting many geographically distributed IaaS resources. The Galaxy administrators needing additional reference data not already present in the repositories will still be able to add them using the available storage assigned to their instance.

Data protection and isolation

Unless proper countermeasures are in place, users data on a VM could be exposed to anybody having legitimate or illegitimate access to the underlying IaaS and physical hardware [39]. This naturally causes technical, ethical and legal concerns, in particular when sensible human genomic data are involved. Thus, some of the potential users of a PaaS as Laniakea, for example, health operators and researchers involved in clinical bioinformatics or similar scenarios, may be unwilling to use it wary of possible data breaches. Laniakea tackles this issue by providing to the Galaxy instance administrator a security layer that seamlessly encrypts the storage volume using file system level encryption based on the *dm-crypt* Linux kernel's subsystem coupled with LUKS encryption strategy based on aes-xts-plain64 cipher (Table S3 in Supplementary2.docx). The storage volume is encrypted employing a *key stretching* approach: a randomly generated master key is encrypted using the user passphrase through PBKDF2 key derivation. This procedure makes both brute force and *rainbow tables* [40] based attacks more computationally expensive, allows for multiple passphrases and for passphrase change or revocation without re-encryption. Finally, the LUKS *anti-forensic splitter* feature protects data against recovery after volume deletion. The resulting instance layout consists in Galaxy running on top of a standard file system but transparently using the encrypted volume for storing data as long as it is unlocked and mounted (Figure 5).

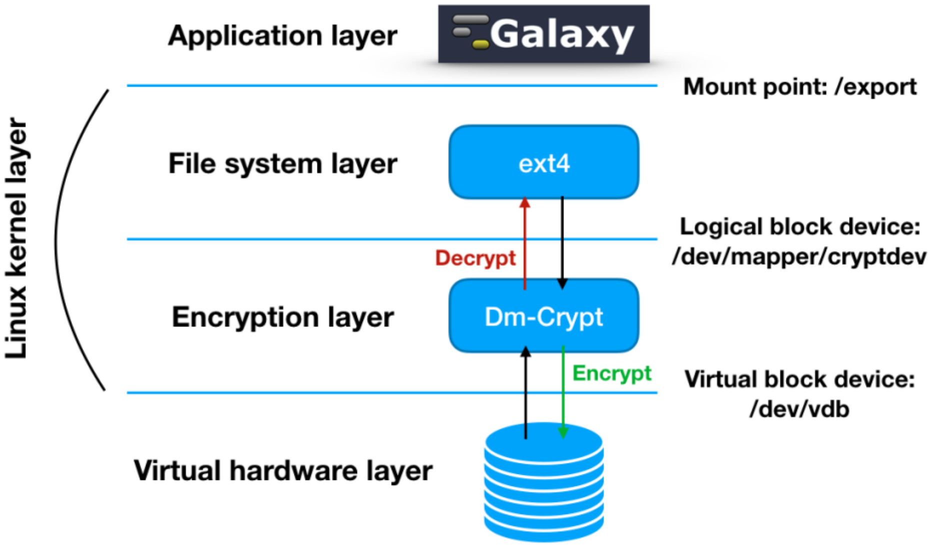


Fig. 5. The relationship between Galaxy, the file system, and dm-crypt. Data are encrypted and decrypted on-the-fly when writing and reading through dm-crypt. The underlying disk encryption layer is completely transparent to Galaxy that employs a specific mount point in order to store and retrieve files from the volume.

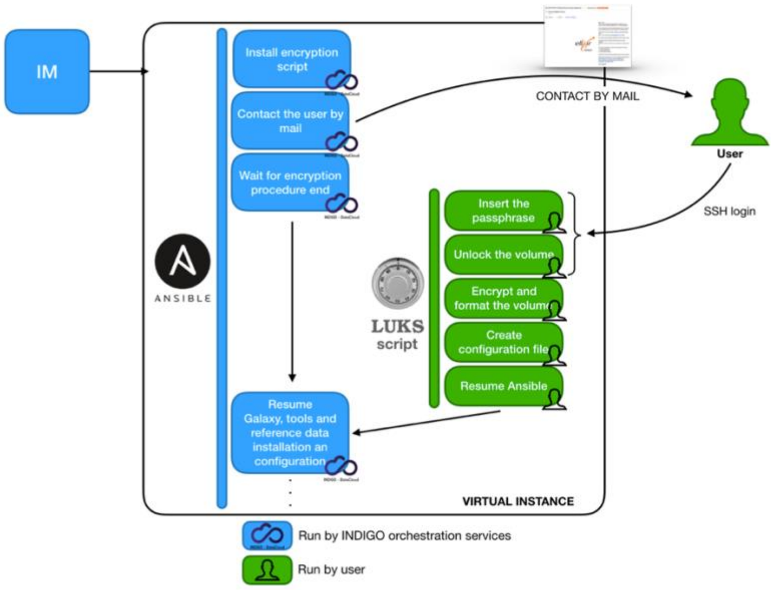

IM
Install encryption script
Contact the user by mail
Wait for encryption procedure end
ANSIBLE
CONTACT BY MAIL
User
SSH login
Insert the passphrase
Unlock the volume
Encrypt and format the volume
Create configuration file
Resume Ansible
LUKS script
Resume Galaxy, tools and reference data installation an configuration
VIRTUAL INSTANCE
Run by INDIGO orchestration services
Run by user

Fig. 6. Storage encryption workflow. The user receives an e-mail with instructions to connect through SSH to the virtual Galaxy instance being created. The user is then requested to enter a passphrase that will be used to encrypt the volume and requested each time the encrypted volume will need to be unlocked, e.g. during a reboot of the VM hosting the Galaxy instance.

The encryption procedure is coordinated by the IM which installs the encryption package and sends to the PaaS user an e-mail containing the information needed to login into the newly created Galaxy instance via SSH, together with a brief description of the encryption procedure and a detailed step by step how-to (Figure 6). This manual intervention is required to insert the passphrase for file system encryption, a similar procedure will allow mounting the encrypted volume each time the encrypted Galaxy instance is re-started. This two-steps solution separates the orchestration of services from the encryption procedure, ensuring that the encryption passphrase is never being exchanged as plain text during the deployment procedure and avoiding any interaction with the IaaS administrator(s). This results in the Galaxy instance administrator being the only one holding the passphrase to unlock the encrypted volume.

To validate our data strategy, we simulated two different scenarios where a malicious attacker tries to gain access to the data stored in the encrypted volume. In the first scenario, the attacker obtains unauthorized access to the unmounted encrypted volume, while the second simulates the use of improper use of administrator IaaS privileges when the LUKS volume is already unlocked and in use by a running Galaxy instance.

For the first scenario we compared two identical volumes, one encrypted and the other not, both attached to the same Galaxy instance, with the same set of permissions and each containing a copy of the same plain text file. Once detached, we created a binary image file of each volume and tried to access the data structure through hex dump. We were able to easily retrieve the original content of the text file from the non-encrypted volume while the hex dump of the encrypted volume did not contain the original text in any discernible form.

For the second scenario, we tried to read data from the volume when already active, that is mounted on a running Galaxy instance, using the OpenStack cloud controller. It turned out to be impossible reading the LUKS device without providing the correct passphrase. All in all, we think that the described approach positively helps in insulating any data uploaded to an encrypted Galaxy instance from malicious access as long as the instance itself and the encryption keys remain uncompromised.

Cluster support

From the point of view of the IaaS administrator, the option to offer static and/or elastic cluster (Fig. 7) support to users provides the alternative to guarantee a constant pool of resources to those instances attached to a static cluster, or greater control over the efficient usage of the available resources for those instances that instead rely on an elastic cluster. In fact, the latter solution dynamically scales the number of cluster nodes available to a Galaxy instance depending on its workload. From the user's perspective, both solutions enable straightforward access to computational resources beyond the ones assigned to the Galaxy instance virtual hardware, enabling a greater number of simultaneous users, faster job execution and more room for computationally demanding analysis tools.

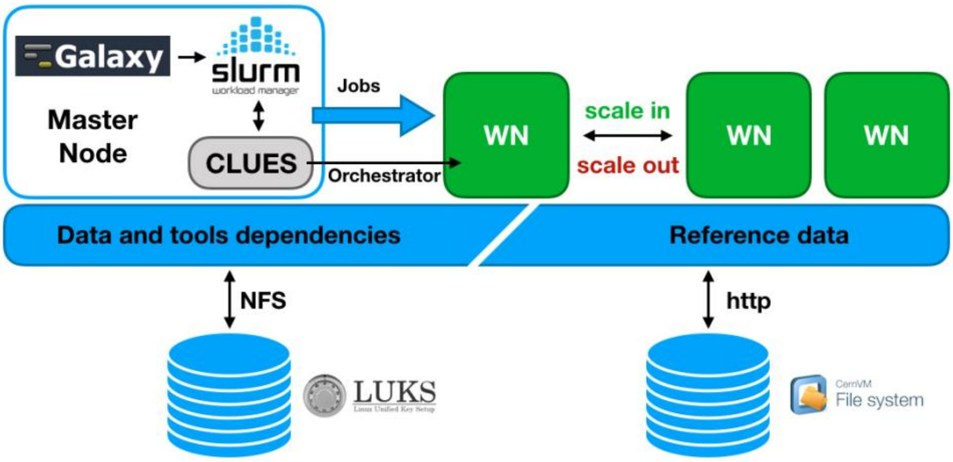


Fig. 7. Galaxy elastic cluster architecture. Initially only the master node, that hosts Galaxy, SLURM and CLUES, is deployed. The SLURM queue is monitored by CLUES and new worker nodes are deployed to process pending jobs up to the maximum number set during cluster configuration, thus adapting resources availability to the current workload. The user home directory and persistent storage are shared among master and worker nodes through Network File System (NFS) enabling the sharing of CONDA tools dependencies. If and when the dependencies are not satisfied by CONDA, the needed packages are installed during deployment on each worker node. The CVMFS shared volume is also mounted on each worker node so to ensure that tools have access to reference data.

Laniakea pilot instance

In order to test and demonstrate Laniakea, we have deployed a pilot service over an OpenStack (Mitaka release) IaaS hosted at the ELIXIR-IT ReCaS-Bari facility [41], of which we report the current INDIGO PaaS layer configuration in Table S4 (Supplementary2.docx). During the first closed beta program, starting Dec 2018, we will reserve to end users up to 128 CPUs, 256 GB of RAM and 10 TB of disk storage. We will gradually increase those resources over the next few months in order to make this prototype instance grow into a full-fledged service provided by the Italian Node of ELIXIR (https://www.elixir-europe.org/). The service front-end is available at elixir-italy-laniakea.cloud.ba.infn.it.

Discussion

Laniakea provides a quite complete solution to easily add a Galaxy on-demand service to the portfolio of public and private cloud providers. This is achieved leveraging INDIGO middleware that, having been designed to support a vast array of scientific services, may already be present and supported within the same cloud infrastructure or, as an alternative, can also be used to pilot locally available computational resources from a remote deployment. Laniakea's scalability features encompass a variety of Galaxy setups that span from small instances to serve e.g. small research groups, developers and didactic purposes to production grade instances with (elastic) cluster support supporting multiple concurrent users and demanding analyses. New Galaxy flavours, implementing and making available future or existing data analysis pipelines, can be quickly deployed and shared through Ansible recipes to ease the error-prone and lengthy routines of tools installation. We are confident that the Galaxy PaaS delivered with Laniakea can effectively mitigate the need to host and maintain local hardware and software infrastructures in many different scenarios, enabling at the same time the more efficient use of the available resources, the strong reliability offered by cloud environments and also helping to improve reproducibility. Finally, the embedded data security features provide an insulated work environment that goes in the direction of addressing some of the burdens typical of research and clinical settings involving sensitive genomic data. In future developments aimed at improving Laniakea's compatibility with existing cloud setups, we will extend cluster support to other resource managers (e.g. TORQUE [30], HTCondor [42], etc...). Next, we plan to add the possibility to instantiate Galaxy Docker containers as long running services, exploiting the wide number of dockerized Galaxy flavours hosted at Docker Hub (https://hub.docker.com/) and maintained by the Galaxy community and finally to provide seamless integration with "dockerized" tools.

Conclusions

Laniakea offers a clear example of the current trend of services virtualization, following the direction set forth e.g. by the European Open Science Initiative (EOSC) Declaration and enabling researchers and scientific services providers to implement many of the recommendations therein outlined. In fact, Laniakea offers a platform-agnostic, cloud-based service that can be almost effortlessly kept up to date; this in turn facilitates the provision of software and services for the Life Science field and beyond, contributes to the efficient employment of existing and future computational resources and, by easing the barriers posed by the software and hardware requirements needed to deploy and maintain a Galaxy instance, enables access to cutting-edge technology for a wide array of researchers and other stakeholders.

Acknowledgements

This work has been supported by the European Commission H2020 research and innovation program under grant agreement RIA 653549.

References

1. Piccolo SR, Frampton MB. Tools and techniques for computational reproducibility. Gigascience. 2016;5:1–13.

2. Kumar S, Dudley J. Bioinformatics software for biologists in the genomics era. Bioinformatics. 2007;23:1713–7.

3. Attwood TK, Blackford S, Brazas MD, Davies A, Schneider MV. A global perspective on evolving bioinformatics and data science training needs. Brief Bioinform [Internet]. 2017;1–7. Available from: http://academic.oup.com/bib/article/doi/10.1093/bib/bbx100/4096809/A-global-perspective-on-evolving-bioinformatics

4. Beckmann JS, Lew D. Reconciling evidence-based medicine and precision medicine in the era of big data: Challenges and opportunities. Genome Med [Internet]. Genome Medicine; 2016;8:1–11. Available from: http://dx.doi.org/10.1186/s13073-016-0388-7

5. Cohen-Boulakia S, Belhajjame K, Collin O, Chopard J, Froidevaux C, Gaignard A, et al. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. Futur Gener Comput Syst [Internet]. Elsevier B.V.; 2017;75:284–98. Available from: http://dx.doi.org/10.1016/j.future.2017.01.012

6. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res [Internet]. 2018;46:W537–44. Available from: http://dx.doi.org/10.1093/nar/gky379

7. Mell P, Grance T. The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. Nist Spec Publ. 2011;145:7.

8. Langmead B, Nellore A. Cloud computing for genomic data analysis and collaboration. Nat Rev Genet [Internet]. Nature Publishing Group; 2018;19:208–19. Available from: http://www.nature.com/doifinder/10.1038/nrg.2017.113

9. Karim R, Michel A, Zappa A, Baranov P, Sahay R, Rebholz-schuhmann D. Improving data workflow systems with cloud services and use of open data for bioinformatics research. Brief Bioinform [Internet]. 2017;1–16. Available from: http://fdslive.oup.com/www.oup.com/pdf/production_in_progress.pdf

10. Pavlovich M. Computing in Biotechnology: Omics and Beyond. Trends Biotechnol [Internet]. Elsevier Ltd; 2017;35:479–80. Available from: http://dx.doi.org/10.1016/j.tibtech.2017.03.011

11. Warth B, Levin N, Rinehart D, Teijaro J, Benton HP, Siuzdak G. Metabolizing Data in the Cloud. Trends Biotechnol [Internet]. Elsevier Ltd; 2017;35:481–3. Available from: http://dx.doi.org/10.1016/j.tibtech.2016.12.010

12. Griebel L, Prokosch HU, Köpcke F, Toddenroth D, Christoph J, Leb I, et al. A scoping review of cloud computing in healthcare. BMC Med Inform Decis Mak. 2015;15:1–16.

13. Bellazzi R. Big Data and Biomedical Informatics : A Challenging Opportunity Big Data : Why Bother ? Big Data : Must-have or. Yearb Med Inform. 2014;9:8–13.

14. Liu B, Madduri RK, Sotomayor B, Chard K, Lacinski L, Dave UJ, et al. Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses. J Biomed Inform [Internet]. Elsevier Inc.; 2014;49:119–33. Available from: http://dx.doi.org/10.1016/j.jbi.2014.01.005

15. Afgan E, Baker D, Coraor N, Chapman B, Nekrutenko A, Taylor J. Galaxy CloudMan: Delivering cloud compute clusters. BMC Bioinformatics. 2010;11:2–7.

16. Afgan E, Chapman B, Taylor J. CloudMan as a platform for tool, data, and analysis distribution. BMC Bioinformatics [Internet]. BMC Bioinformatics; 2012;13:1. Available from: BMC Bioinformatics

17. Afgan E, Sloggett C, Goonasekera N, Makunin I, Benson D, Crowe M, et al. Genomics Virtual Laboratory: A practical bioinformatics workbench for the cloud. PLoS One. 2015;10:1–20.

18. Krieger MT, Torreno O, Trelles O, Kranzlmüller D. Building an open source cloud environment with auto-scaling resources for executing bioinformatics and biomedical workflows. Futur Gener Comput Syst [Internet]. Elsevier B.V.; 2017;67:329–40. Available from: http://dx.doi.org/10.1016/j.future.2016.02.008

19. Afgan E, Baker D, van den Beek M, Blankenberg D, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic Acids Res. 2016;44:W3–10.

20. Afgan E, Baker D, Coraor N, Goto H, Paul IM, Makova KD, et al. Harnessing cloud computing with Galaxy Cloud. Nat Biotechnol. 2011;29:972–4.

21. Campos DSI, Marco LGJ, Solagna DLP, Matyska JGL, Hardt PFM, Dutka GDL, et al. INDIGO-DataCloud : a Platform to Facilitate Seamless Access to E-Infrastructures. J Grid Comput [Internet]. 2018; Available from: https://link.springer.com/article/10.1007%2Fs10723-018-9453-3

22. Salomoni D, Campos I, Gaido L, Donvito G, Antonacci M, Fuhrman P, et al. INDIGO-Datacloud: foundations and architectural description of a Platform as a Service oriented to scientific computing. 2016;1–31. Available from: http://arxiv.org/abs/1603.09536

23. Lipton P (Ca T, Moser S (Ibm), Palma D (Vnomic), Spatzier T (Ibm). Topology and Orchestration Specification for Cloud Applications - PRIMER. 2013;1–114. Available from: http://docs.oasis-open.org/tosca/TOSCA/v1.0/cs01/TOSCA-v1.0-cs01.html

24. OASIS. TOSCA Simple Profile in YAML Version 1 . 0 Committee Specification Draft 04 / Public Review Draft 01. 2015; Available from: http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/csprd01/TOSCA-Simple-Profile-YAML-v1.0-csprd01.pdf

25. Ben-Kiki O, Evans C, Ingerson B. YAML Ain't Markup Language (YAML^{TM}) Version 1.2. Language (Baltim) [Internet]. 2009;1–100. Available from: http://www.yaml.org/spec/1.2/spec.html

26. Housley R, Polk W, Ford W, Solo D. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. United States: RFC Editor; 2002.

27. OpenID Foundation. OpenID Connect Discovery 1.0 incorporating errata set 1. 2014;311376. Available from: http://openid.net/specs/openid-connect-discovery-1_0.html

28. Cantor S, Hodges J, Hirsch F, Philpott R, Security RS a, Hughes J, et al. Profiles for the OASIS

Security Assertion Markup Language (SAML). Language (Baltim) [Internet]. 2005;16:66. Available from: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Profiles+for+the+OASIS+Security+Assertion+Markup+Language+(SAML)+V2.0#0

29. Yoo AB, Jette MA, Grondona M. SLURM: Simple Linux Utility for Resource Management. In: Feitelson D, Rudolph L, Schwiegelshohn U, editors. Job Sched Strateg Parallel Process. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003. p. 44–60.

30. Staples G. TORQUE Resource Manager. Proc 2006 ACM/IEEE Conf Supercomput [Internet]. New York, NY, USA: ACM; 2006. Available from: http://doi.acm.org/10.1145/1188455.1188464

31. Fruhwirth C. New methods in hard disk encryption. Inst Comput Lang Theory Log … [Internet]. 2005; Available from: http://git.dyne.org/tomb/plain/doc/New_methods_in_HD_encryption.pdf

32. Buncic P, Aguado Sanchez C, Blomer J, Franco L, Harutyunian A, Mato P, et al. CernVM - A virtual software appliance for LHC applications. J Phys Conf Ser. 2010;219.

33. Grüning BA, Fallmann J, Yusuf D, Will S, Erxleben A, Eggenhofer F, et al. The RNA workbench: Best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. Nucleic Acids Res. 2017;45:W560–6.

34. Chiara M, Gioiosa S, Chillemi G, D'Antonio M, Flati T, Picardi E, et al. CoVaCS: a consensus variant calling system. BMC Genomics [Internet]. BMC Genomics; 2018;19:120. Available from: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-018-4508-1

35. Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009;10.

36. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9:357–9.

37. Li H, Li H, Durbin R, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics [Internet]. 2009;25:1754–60. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2705234%5C&tool=pmcentrez%5C&rendertype=abstract%5Cnpapers2://publication/doi/10.1093/bioinformatics/btp324

38. Li H, Durbin R. Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics. 2010;26:589–95.

39. Yuchi X, Shetty S. Enabling security-aware virtual machine placement in IaaS clouds. Proc - IEEE Mil Commun Conf MILCOM. 2015;2015–December:1554–9.

40. Oechslin P. Making a Faster Cryptanalytic Time-Memory Trade-Off. 2003;617–30. Available from: http://link.springer.com/10.1007/978-3-540-45146-4_36

41. Antonacci M, Bellotti R, Cafagna F, de Palma M, Diacono D, Donvito G, et al. The ReCaS Project: The Bari Infrastructure. High Perform Sci Comput Using Distrib Infrastructures Results Sci Appl Deriv from Ital PON ReCaS Proj. World Scientific; 2017. p. 17–33.

42. Thain D, Tannenbaum T, Livny M. Distributed computing in practice: The Condor experience. Concurr Comput Pract Exp. 2005;17:323–56.

[indigo1] Salomoni D, Campos I, Gaidet L al. INDIGO-Datacloud: foundations and architectural description of a Platform as a Service oriented to scientific computing. arXiv:1603.09536

[indigo2] Salomoni D, Campos I, Gaidet L al. INDIGO-DataCloud: A data and computing platform to facilitate seamless access to e-infrastructures. arXiv:1711.01981v4

[tosca] Lipton, P.C.T., Moser, S.I., Palma, D.V., Spatzier, T.I. Topology and Orchestration Specification for Cloud Applications. Tech. rep., OASIS Standard (2013)

[openstack] OpenStack Foundation: OpenStack. https://www.openstack.org (2017)

[opennebula] OpenNebula Project: OpenNebula. https://www.opennebula.org (2017) [scheduling] Alvaro Lopez Garcia et al. Improved Cloud resource allocation: how INDIGO-DataCloud is overcoming the current limitations in Cloud schedulers. arXiv:1707.06403v1

[iam] Ceccanti A., et al. The INDIGO-Datacloud Authentication and Authorization Infrastructure. 2017 J. Phys.: Conf. Ser. 898 102016

[fgw] Marcin Płóciennik et al. Two-level Dynamic Workflow Orchestration in the INDIGO DataCloud for Large-scale, Climate Change Data Analytics Experiments. https://doi.org/10.1016/j.procs.2016.05.359

[LUKS_web] https://gitlab.com/cryptsetup/cryptsetup

[LUKS_spec] Clemens Fruhwirth, https://mirrors.edge.kernel.org/pub/linux/utils/cryptsetup/LUKS_docs/on-disk-format.pdf

[GVL] Afgan E, Sloggett C, Goonasekera N, Makunin I, Benson D, Crowe M, Gladman S, Kowsar Y, Pheasant M, Horst R, Lonie A., Genomics Virtual Laboratory: A Practical Bioinformatics Workbench for the Cloud., PLoS One. 2015 Oct 26;10(10):e0140829.

[cloudman1] Afgan E., Baker D., Coraor N., Goto H., Paul I.M., Makova K.D., Nekrutenko A., Taylor J., "Harnessing cloud computing with Galaxy Cloud," Nature Biotechnology, Vol 29, Issue 11, 2011.

[cloudman2] https://galaxyproject.org/news/2018-01-22-gvl430/

[WP29] Advice paper on special categories of data ("sensitive data"). http://ec.europa.eu/justice/article-29/documentation/other-document/files/2011/2011_04_20_letter_artwp_mme_le_bail_directive_9546ec_annex1_en.pdf