## TECHNICAL NOTE

# Laniakea: an open solution to provide Galaxy "on-demand" instances over heterogeneous cloud infrastructures

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

[1]Institute of Biomembranes, Bioenergetics and Molecular Biotechnologies, National Research Council (CNR), Via Giovanni Amendola 122/O, 70126 Bari, Italy; [2]National Institute for Nuclear Physics (INFN), Section of Bari, Via Orabona 4, 70126 Bari, Italy; [3]Department of Biosciences, University of Milan, via Celoria 26, 20133 Milano, Italy and [4]Department of Biosciences, Biotechnologies and Biopharmaceutics, University of Bari, Via Orabona 4, 70126 Bari, Italy

[*]**Correspondence address.** Federico Zambelli, Department of Biosciences, University of Milan, via Celoria 26, 20133, Milano, Italy. E-mail: federico.zambelli@unimi.it [ID] http://orcid.org/0000-0003-3487-4331

## Abstract

**Background:** While the popular workflow manager Galaxy is currently made available through several publicly accessible servers, there are scenarios where users can be better served by full administrative control over a private Galaxy instance, including, but not limited to, concerns about data privacy, customisation needs, prioritisation of particular job types, tools development, and training activities. In such cases, a cloud-based Galaxy virtual instance represents an alternative that equips the user with complete control over the Galaxy instance itself without the burden of the hardware and software infrastructure involved in running and maintaining a Galaxy server. **Results:** We present Laniakea, a complete software solution to set up a "Galaxy on-demand" platform as a service. Building on the INDIGO-DataCloud software stack, Laniakea can be deployed over common cloud architectures usually supported both by public and private e-infrastructures. The user interacts with a Laniakea-based service through a simple front-end that allows a general setup of a Galaxy instance, and then Laniakea takes care of the automatic deployment of the virtual hardware and the software components. At the end of the process, the user gains access with full administrative privileges to a private, production-grade, fully customisable, Galaxy virtual instance and to the underlying virtual machine (VM). Laniakea features deployment of single-server or cluster-backed Galaxy instances, sharing of reference data across multiple instances, data volume encryption, and support for VM image-based, Docker-based, and Ansible recipe-based Galaxy deployments. A Laniakea-based Galaxy on-demand service, named Laniakea@ReCaS, is currently hosted at the ELIXIR-IT ReCaS cloud facility. **Conclusions:** Laniakea offers to scientific e-infrastructures a complete and easy-to-use software solution to provide a Galaxy on-demand service to their users. Laniakea-based cloud services will help in making Galaxy more accessible to a broader user base by removing most of the burdens involved in deploying and running a Galaxy service. In turn, this will facilitate the adoption of Galaxy in

scenarios where classic public instances do not represent an optimal solution. Finally, the implementation of Laniakea can be easily adapted and expanded to support different services and platforms beyond Galaxy.

*Keywords:* galaxy; on-demand; cloud; workflow; PaaS

## Background

The recent improvements in our capacity to gather vast amounts of complex, multilayered, and interconnected biomolecular data demand a parallel development and enhancement of the computational tools that are employed 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 to, among other things, a profusion of data formats, long and convoluted command lines, versioning, and the need to handle and store multiple intermediate files. In turn, frequently, this makes harnessing the information contained in the biomolecular data onerous even for expert bioinformaticians, which currently represent a limited resource [1–3]. Furthermore, the complexity of bioinformatics analyses often poses a significant obstacle to reproducibility [4] as well as being an intimidating barrier for biologists aspiring to explore their data in autonomy [5], students [1], and health care operators planning to adopt *clinical bioinformatics* approaches within their medical protocols [6]. For these reasons, in the past few years, considerable effort has been put in the development of workflow manager software platforms for bioinformatics (see [7] for a review). Usually, these platforms not only provide a more user-friendly work environment but also improve reproducibility, facilitate data sharing, and enable collaborative data processing.

### Galaxy

The Galaxy platform is one of the most successful examples of such workflow management software. Indeed, by providing a consistent, user-friendly, flexible, practical, and customisable gateway to a vast array of bioinformatics software and analysis workflows, Galaxy has attracted a vast and thriving community of users [8]. The software consists of an open-source server-side application accessible through a simple web interface that serves as a gateway to a wealth of tools for data set handling and analysis, workflow design, visualisation, and sharing of results. Although, from the user's perspective, using a Galaxy service to run bioinformatics analyses is pretty straightforward, the deployment of a production-grade Galaxy instance can require the configuration and maintenance of an extensive collection of helper components (e.g., database management system, web server, load balancer) and an even more extensive collection of bioinformatic tools and reference data. These issues, coupled with the need for an adequate IT infrastructure required to support a Galaxy service properly, have usually restricted the role of Galaxy service providers to institutions or groups with suitable IT facilities and appropriate technical know-how. At the time of writing, around 125 Public Galaxy instances are available [9], including the Galaxy Project flagships usegalaxy.org, usegalaxy.eu, and usegalaxy.org.au, serving a vast community of users [8]. These are very welcome and useful resources. However, the public nature of the service implies some hardly addressable shortcomings, such as limited user quotas for computing and storage resources, lack of customisation options (e.g., installing or implementing new Galaxy tools or linking the instance to local data repositories), and potential concerns for data security and privacy, a worry particularly noticeable when processing sensitive data. These considerations can, in turn, limit or outright interdict the use of Galaxy public instances for some specific applications, for example, analyses requiring substantial computational resources, precision medicine [10, 11], development or porting of new tools into Galaxy, and training and teaching activities [12].

### Cloud solutions to Galaxy provision

The cloud computing model [13] is rapidly gaining popularity within the life sciences [14–19] and the biomedical [6, 20, 21] communities. Among other advantages, cloud computing offers solutions and features that can overcome or mitigate the limitations of Galaxy public instances discussed in the previous section. Several efforts have already been put forward in this regard. Globus Genomics [22] provides a Galaxy-based bioinformatics workflow platform, built on Amazon cloud services, for large-scale next-generation sequencing analyses. CloudMan [23, 24] allows the deployment of personal Galaxy instances relying on arbitrarily sized compute clusters on the Amazon cloud infrastructure (and can support OpenStack [25] and OpenNebula [26] through custom deployments). In 2017, CloudMan merged forces with the Genomic Virtual Laboratory (GVL) [27], providing a cloud-based suite of genomics analysis tools, including Galaxy, that can be deployed on OpenStack-based clouds as well as on Amazon Web Services (AWS) and Azure. PhenoMeNal [28] is a recent effort to develop a Cloud Research Environment for metabolomics that includes Galaxy, among several other software. A further example is provided by Krieger and colleagues, who describe a possible configuration stack to deploy Galaxy on an OpenStack-based infrastructure as a service (IaaS) [29].

The attractiveness of Galaxy cloud solutions is made evident also in the 2016 Galaxy update [30], reporting that over 2,400 Galaxy servers were launched on the Amazon cloud in 2015 alone, pointing to strong demand for ready-to-use but private virtual Galaxy instances. The Amazon Galaxy service [31] is, however, a commercial solution that can discourage researchers or health care facilities from adopting it due to funding or budget issues, ethical concerns, or legal requirements (e.g., EU General Data Protection Regulation). Finally, an interesting point that has emerged only recently is how complex scientific user requirements, as can be considered bioinformatics workflows, can be better served by federated cloud solutions that bind together distinct and heterogeneous cloud infrastructures rather than by single cloud providers (see, e.g., [32, 33] among others).

## Overview

Hereby we introduce Laniakea [34], a software framework for the provision of *on-demand* Galaxy instances over federated cloud e-infrastructures. Laniakea can be instantiated on existing scientific e-infrastructures leveraging the open and modular architecture developed in the context of the INDIGO-DataCloud H2020 project [35, 36] (INDIGO from now on). By hiding the technical complexity behind a user-friendly web front-end, Laniakea allows its users to configure and deploy virtual Galaxy instances

with a handful of clicks, obtaining full administrative-level access to the Galaxy instance and the underlying virtual hardware. Any Galaxy instance generated by Laniakea is accessible from a public IP address and retains all the functionalities and properties of a local Galaxy server. That is, the administrator (usually referred to as *admin user*) can install and configure new tools (e.g., through Galaxy Tool Sheds); add users, which are not required to be also users of the Laniakea service; turn the instance public; configure jobs handling; manage jobs; establish user quotas; and perform any other operation commonly available to Galaxy admin users.

Laniakea supports the deployment of Galaxy instances over three different virtual hardware layouts: single node, static cluster, and elastic cluster, which is a cluster with workload-dependent automatic scaling of the number of nodes. Galaxy jobs execution takes place within the deployed virtual environment. Laniakea can generate Galaxy instances starting from virtual machine (VM) snapshots; we refer to it as Express deployment for brevity, Docker containers [37], or using Ansible recipes [38]. We refer to the latter case as "Live Build," since the entire Galaxy software stack is installed from scratch over a bare virtual machine: all the software components, including Galaxy tools, are retrieved on-the-fly from their respective online repositories. Live Build introduces a flexible alternative compared to the somewhat more cumbersome VM snapshots and Docker containers. In fact, Ansible recipes make it possible to describe and replicate a complete software environment using just tiny text files, while VM snapshots and Docker containers usually weigh several GBs.

Laniakea integrates for the first time, to our knowledge, on a Galaxy on-demand platform, a built-in technology to encrypt storage volumes at filesystem level. Encrypted volumes allow the insulation of data from unauthorized access from malicious attackers or trusted users of the same cloud infrastructure, notably including the administrators of the cloud and hardware layers. The user can activate and manage volume encryption in a straightforward manner directly from the web front-end. Galaxy instances generated by Laniakea can also be automatically linked to local or remote CernVM-FS (CVMFS) repositories hosting reference data, meaning that all the instances created by a Laniakea cloud service can be easily linked to the same read-only storage volume hosting reference data, either local or remote, in order to avoid unnecessary redundancy resulting in waste of storage resources for the hosting cloud infrastructure.

Other functionalities of Laniakea stem from its INDIGO foundations. They include support for a wide array of cloud managers and transparent orchestration of virtual hardware at different sites over a federated cloud infrastructure depending on where resources are available to the user and compatibility with the INDIGO-IAM [39] and ELIXIR-AAI [40] Authentication and Authorisation Infrastructure (AAI) services.

Figure 1 provides an overview of the architecture of Laniakea. A step-by-step installation guide of the whole architecture is available in the Laniakea documentation. Supplementary Table S1 provides a summary of the principal Laniakea functions, comparing them to similar features of PhenoMeNal and GVL.

## Laniakea Dashboard

The Laniakea Dashboard is the web front-end of Laniakea. The home page (Fig. 2) presents the selection of the available Galaxy virtual hardware layouts (i.e., single node, cluster, elastic cluster) and deployment strategies (i.e., Express, Live Build, or Docker).

The configuration front-end is composed of two main panels (Fig. 3) that drive the user through the setup procedure of a new Galaxy instance:

- The Virtual Hardware configuration panel exposes the array of available hardware setups in terms of the number of virtual CPUs and amount of RAM. Users are asked to provide a valid SSH public key at this point, which can also be retrieved from the "SSH keys management" function of the Dashboard (Suppl. Fig. S1). The SSH key is used to grant full access to the virtual machine hosting the Galaxy instance. The data storage volume size and type (i.e., plain or encrypted) are also selected at this stage. If the user selects a cluster-backed Galaxy deployment, the panel allows for the setup of the front-end and worker nodes.
- The Galaxy Configuration panel allows the selection of the Galaxy release version among the ones supported, the reference data CVMFS volume (when more than one is available), the e-mail of the instance administrator, and the Galaxy flavour (see Galaxy flavours).

When the Galaxy deployment procedure terminates, the user is notified with an e-mail message, and a new entry is added to the "My Deployments" page of the Laniakea Dashboard (Fig. 4). The user retains full administrator rights over each created instance and can control and customise them (e.g., installing new tools, adding users, sharing data sets, setting quotas) using the standard Galaxy interface. Expert users that need access to the underlying virtual machine (e.g., to implement novel tools or to tweak advanced configuration options not available from the Galaxy administration interface) can do so using the SSH key they provided during the configuration.

## Galaxy flavours

As with any other Galaxy instance, users of Laniakea can equip their Galaxy instance with a variety of tools that can be browsed, searched, retrieved, and automatically installed from Tools Sheds using the Galaxy admin interface. However, to lighten the effort required from users and to set them in working conditions in the shortest possible time, Laniakea provides also a handy set of four domain-specific "*Galaxy flavours*," that is, Galaxy instances already configured with curated collections of tools covering some of the most common next-generation sequencing analysis pipelines. The provided tools have been extensively tested, organised in workflows, and are ready to be used out of the box.

Apart from the basic "minimal" flavour, providing the standard set of default tools embedded in any Galaxy installation, the four available Galaxy flavours are "Epigen," "RNA workbench," "GDC Somatic Variant Calling," and "CoVaCS." The "Epigen" flavour is based upon the layout of the Epigen Project Galaxy server [41] and provides a selection of tools for the analysis of ChIP-Seq and RNA-Seq data. The "RNA workbench" flavour is based on [42] and consists of more than 50 tools dedicated to RNA-centric analyses, including, for example, alignment, annotation, secondary structure profiling, and target prediction. The "GDC Somatic Variant Calling" flavour (Fig. 5) is a porting of the Genomic Data Commons (GDC) pipeline for the identification of somatic variants on whole-exome/genome sequencing data [43]. Finally, the "CoVaCS" flavour implements the homonymous workflow (Suppl. Fig. S2) described in [44] and comprehends a set of tools for genotyping and variant annotation of whole-genome/exome and target-gene sequencing data.
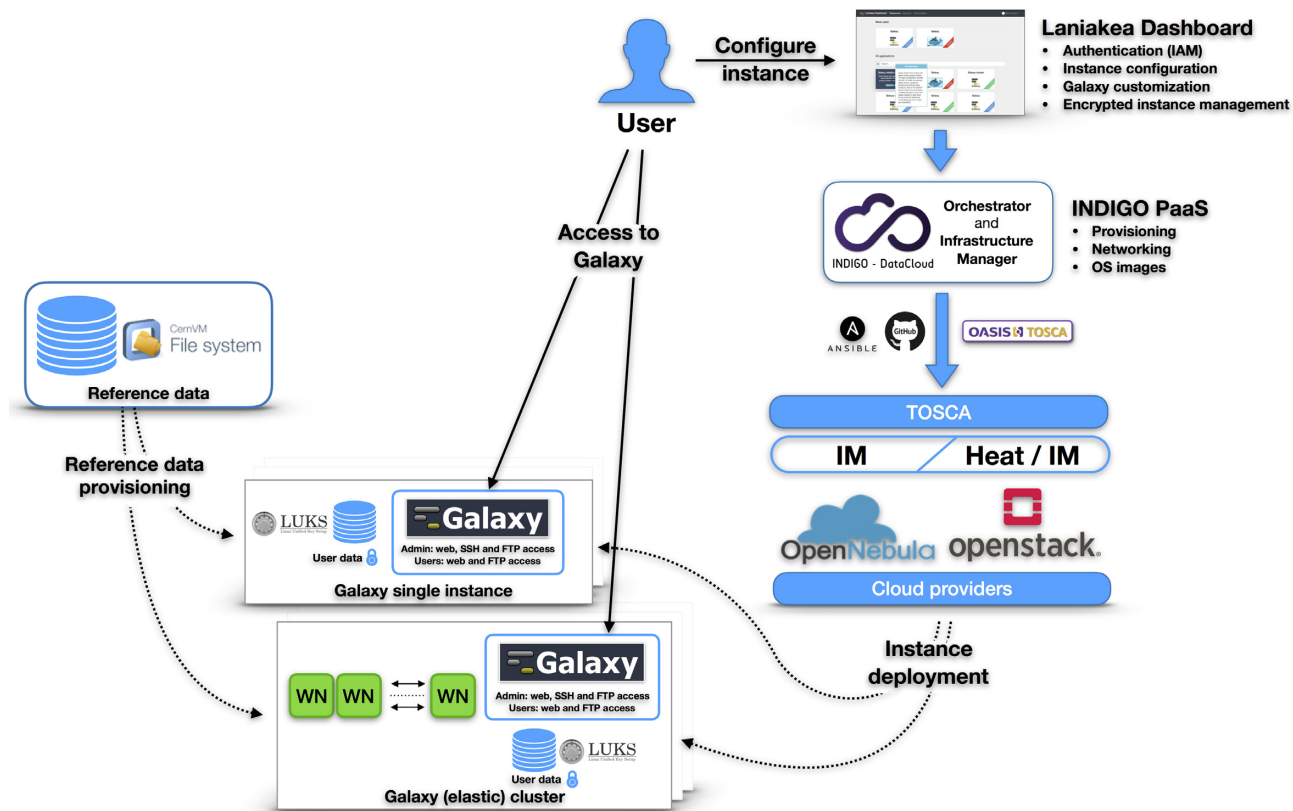
**Figure 1:** Laniakea architecture. The Laniakea Dashboard is the front-end that users access to configure and manage Galaxy instances. When a new Galaxy instance is requested by a user, the resulting TOSCA (Topology and Orchestration Specification for Cloud Orchestration) [45] template is sent to the platform as a service layer that employs INDIGO services to deploy the instance over the infrastructure as a service (IaaS), retrieving the needed virtual hardware, storage, and networking resources. Finally, the Galaxy instance is configured with the requested set of tools (flavour) and attached to a plain or Linux Unified Key Setup (LUKS) encrypted storage volume and the CernVM-FS shared volume hosting reference data. At the end of the process, a public IP address is assigned to the freshly minted Galaxy instance and made available to the user.

New flavours can be implemented easily in Laniakea by providing the corresponding virtual machine snapshot, Docker container, and Ansible recipe needed for the corresponding deployment strategies (Express, Docker, and Live Build). A simple and documented procedure, requiring only the list of desired Galaxy tools, is in place to allow users to design customized flavours to be made available in Laniakea.

### Reference data volume

To avoid useless replication of reference data and facilitate the reproducibility of analyses, Laniakea supports linking Galaxy to read-only CVMFS volumes that can be shared among multiple instances. As proof of concept, the Laniakea@ReCaS service (see "Laniakea@ReCaS testing and production service") currently provides access to three different reference data repositories. The first, named "ELIXIR-IT Galaxy reference data," is a basic, manually curated, reference data set containing the latest releases of the human, mouse, yeast, fruit fly, and *Arabidopsis thaliana* genomes and the corresponding indexes for Bowtie [46], Bowtie2 [47], and BWA [48, 49]. The second, named "ELIXIR-IT Galaxy CoVaCS reference data," is a repository tailored for variant calling in humans, to be used with the CoVaCS flavour, with a collection of publicly available data sets of human genetic variants derived from a selection of large-scale resequencing projects [50–52], along with curated human genome assemblies and indexes obtained from the GATK bundle repository [53].

Finally, a mirror of the Galaxy project "by hand" reference repository, named "usegalaxy.org Galaxy reference data," is provided.

### Data volume encryption

Unless proper countermeasures are in place, data stored on a data volume linked to a virtual machine can potentially be exposed to anyone with legitimate or illegitimate access to the underlying IaaS and physical hardware [54]. These considerations are exceptionally relevant for health operators and researchers involved in clinical bioinformatics or with the analysis of sensible data in general. We tackle this issue by providing Laniakea users with the option to attach to any Galaxy instance a secure data storage volume with filesystem-level encryption. The storage volume is encrypted using a *key stretching* approach: a randomly generated master key is encrypted using an instance-specific passphrase through PBKDF2 key derivation. This approach makes both brute force and *rainbow tables* [55] based attacks more computationally expensive and, at the same time, allows for multiple passphrases and passphrase change or revocation without reencryption. A randomly generated alphanumerical key is univocally assigned to each user and safely stored by Laniakea using Hashicorp Vault [56] to make the system even more robust and prevent users from using weak passphrases or losing them. In this way, the passphrase can, at the same time, be seen and used only by its legitimate holder, and the need to remember it is removed. Finally, the Linux Unified Key Setup

**Figure 2:** Laniakea Dashboard home page. Each tile provides a quick explanation of the corresponding application and links to the configuration panels (see Fig. 3). Deployments using virtual machine snapshots correspond to the tiles labelled as "Express." Deployments using Docker containers correspond to the tiles labelled as "Docker." Finally, deployments using Ansible recipes correspond to the tiles labelled as "Live Build."



**Figure 3:** Laniakea Dashboard configuration panels. The "Virtual hardware" tab (left) allows the selection of the virtual hardware in terms of number of virtual CPUs, amount of RAM, size of the data volume (encrypted or not), and number and hardware configuration of the worker nodes (only for cluster deployments), and it requires the public SSH key of the user. The "Galaxy" tab (right) is used to tweak the software configuration: Galaxy version, description of the instance, the e-mail address of the administrator, Galaxy flavour (see Galaxy flavours), and reference data repository.

**Figure 4:** Laniakea Dashboard information and management interface. It reports the name, current status, creation time, initial Galaxy flavour, virtual hardware setup (virtual machine flavour), and the URL (endpoint) of each Galaxy instance generated by the user. Galaxy instances that are needed no more can be deleted using the "Delete" button.
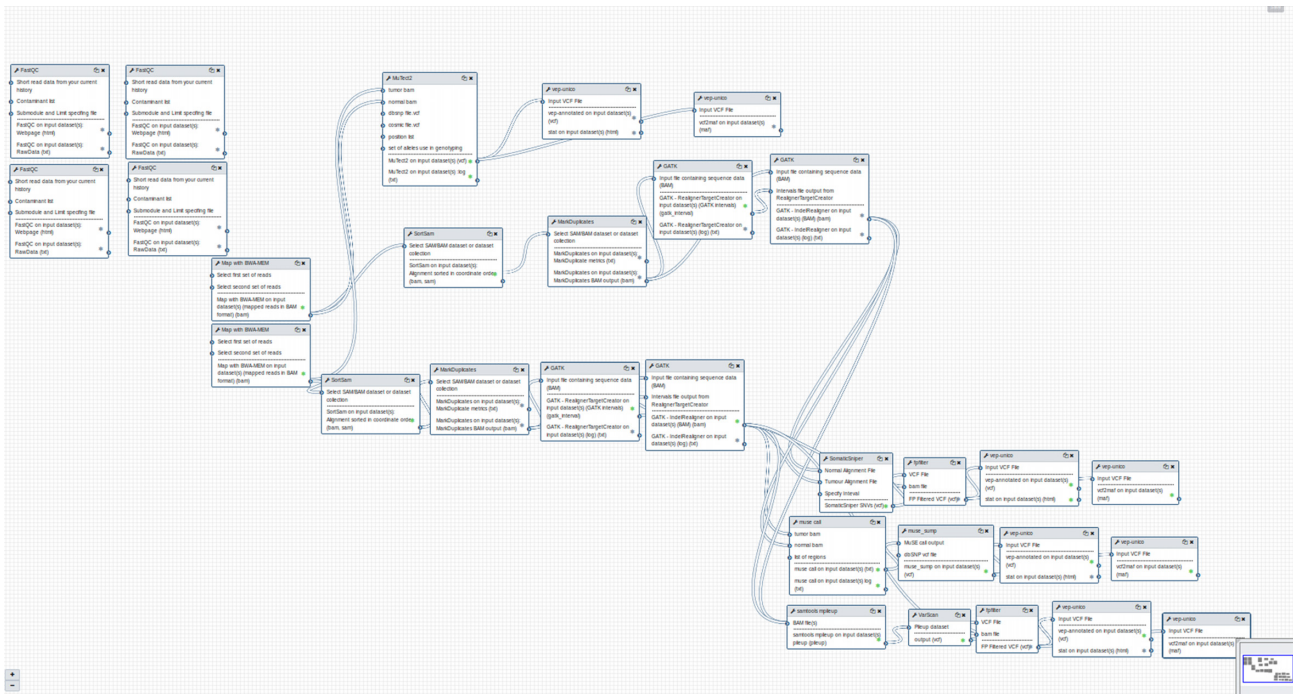


**Figure 5:** The GDC Somatic Variant Calling analysis pipeline implemented as a Galaxy workflow for the corresponding flavour. The workflow design interface of Galaxy is a powerful instrument to elaborate complex workflows chaining together the output and the input of different tools in an intuitive fashion.

(LUKS) *antiforensic splitter* feature protects data against recovery attempts after volume deletion. The resulting layout consists of Galaxy running transparently on top of the encrypted volume (Fig. 6). This means that any Galaxy instance attached to an encrypted volume retains for its users all the functionalities, data sharing included, of any other instance without filesystem-level encryption.

The encryption procedure (Fig. 7) is completely automated and can be enabled by the user through the Laniakea Dashboard during the configuration process of a new Galaxy instance. A similar procedure allows the user to remount the data volume directly from the Dashboard if an encrypted Galaxy instance is rebooted.

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

In 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 a *hex dump*. We were able to quickly 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. In the second scenario, we tried to read data from the volume already mounted on a running Galaxy instance using the OpenStack cloud controller. We were not able to gain access to the LUKS encrypted device by any means without providing the correct passphrase.
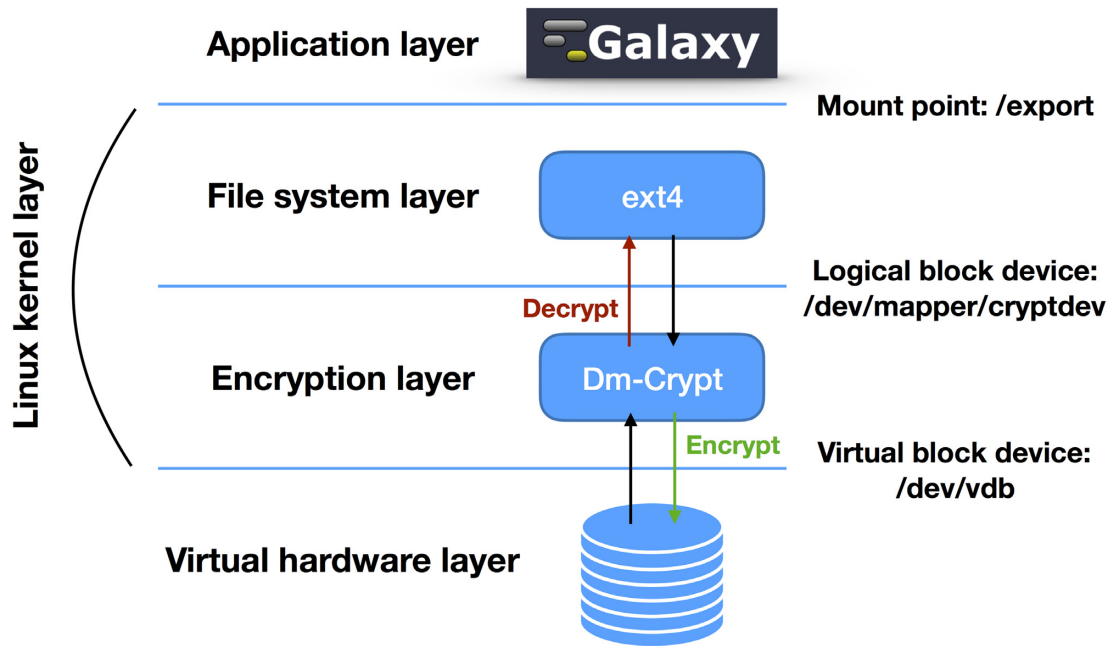
**Figure 6:** The relationship between Galaxy, the filesystem, and dm-crypt. Data are encrypted and decrypted on-the-fly when writing and reading through dm-crypt. The underlying disk encryption layer is entirely transparent for Galaxy.
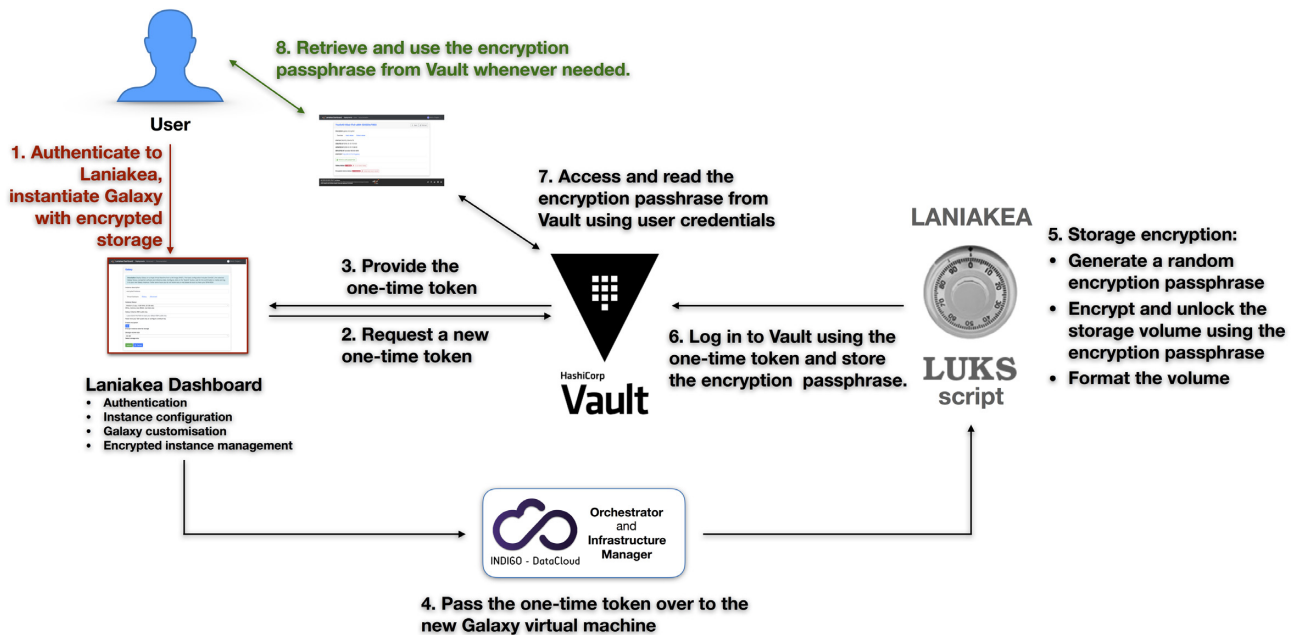


**Figure 7:** Storage encryption workflow. (1) The user logs into the Laniakea Dashboard and enables data encryption when configuring a new Galaxy instance. (2–3) The Dashboard contacts Hashicorp Vault, using an Identity and Access Management token, to retrieve a one-time Vault token. The one-time token is used to avoid transmitting user credentials over the infrastructure and limit any potential damage from a malicious attacker intercepting it (4). The one-time token is passed to the encryption script on the virtual machine through the INDIGO Orchestrator service. (5) A random passphrase is generated, and the data volume is encrypted by Linux Unified Key Setup (LUKS), unlocked, and formatted. It is now ready to be attached to the new Galaxy instance. (6) The encryption script logs into Vault using the one-time token and stores the encryption passphrase that will be accessible only by the Laniakea user that requested the encrypted volume (7–8). The user can retrieve the passphrase at any time using the Dashboard. For example, if the encrypted volume needs to be remounted (usually after a reboot of the Galaxy virtual machine), the user can retrieve the passphrase from Vault and unlock the volume using the Dashboard.

This approach should go a long way to reasonably insulate any data uploaded to an encrypted Laniakea Galaxy instance from malicious access as long as the Galaxy instance itself and the encryption passphrase remain uncompromised.

## Cluster support

For an IaaS administrator, the option to offer static or elastic cluster support to users provides the alternative between guaranteeing a constant pool of resources to those instances at-

tached 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. The INDIGO platform as a service (PaaS) layer used by Laniakea supports both solutions and can deploy static or elastic clusters on top of existing OpenStack/OpenNebula, Amazon AWS, and Google Cloud. Elastic clusters (Fig. 8) dynamically scale the number of nodes available to a Galaxy instance, depending on its workload. From the user point of view, both solutions enable straightforward access to computational resources beyond those assigned to single-node Galaxy instances, enabling a higher number of simultaneous Galaxy users, quicker execution of jobs, and additional room for computationally intensive analyses.

### Laniakea@ReCaS testing and production service

We ran a testing programme for the prototypal Laniakea service, named Laniakea@ReCaS, starting in December 2018 and ending in November 2019. In total, 128 CPUs, 256 GB of RAM, and 10 TB of disk storage were reserved for the service at the ReCaS cloud facility [57, 58]. The programme involved the participation of 15 users from several research institutions and scientific backgrounds that were either directly invited by us or asked to join the programme after being introduced to Laniakea during workshops or other dissemination events. The users were requested to stress-test the service by deploying, deleting, and extensively using one or more virtual Galaxy instances for their daily research activities. During this programme, we collected feedback from users and worked in fixing the juvenile issues of the service, also prioritizing a list of features for future development.

Access to the production service is offered to researchers and other stakeholders on a per-project basis through an open-ended call in a fashion similar to the ongoing ELIXIR-IT HPC@CINECA service (in press). In brief, each project proposal is evaluated by a scientific and technical board. Successful proposals are granted a standard package of computational resources to be used with Laniakea for an amount of time compatible with the project requirements. The service has been added to the catalogue of the EOSC Marketplace [59].

## Discussion and Conclusions

Laniakea provides a solution to easily include a Galaxy on-demand service within the portfolio of public and private scientific cloud providers. This result is achieved by leveraging the INDIGO middleware that, having been designed to support a vast array of scientific services, is extensively used and supported across several European cloud infrastructures. Furthermore, support for federated cloud infrastructures allows piloting computational resources from remote INDIGO services, bypassing the need to have the INDIGO software stack installed on each member of the federated cloud infrastructure.

Laniakea supports a variety of Galaxy setups and deployment strategies that can be useful in multiple employment scenarios (e.g., small research groups, developers, didactic purposes, and even production-grade instances with static or elastic cluster support enabling multiple concurrent users and computationally demanding analyses). Helping in bypassing the need to host and maintain local hardware and software infrastructures in those scenarios, Laniakea favours a more efficient use of the available resources, harnessing the improved reliability offered by cloud environments and enhancing the reproducibility of bioinformatics analyses through Galaxy.

New Laniakea Galaxy flavours can be quickly developed and shared in the form of VM snapshots, Docker containers, or Ansible recipes. These preconfigured Galaxy instances save users from lengthy routines of tools installation and provide a means to nimbly develop and make available data analysis pipelines as we did with the CoVaCS and GDC Somatic Variant flavours. Domain-specific public Galaxy instances have already been developed using Laniakea as a platform to implement and put into production novel services [60, 61]. At the same time, Laniakea users keep the ability to customise their Galaxy instances with all the options and instruments commonly available to any other Galaxy admin (e.g., Tool Sheds, user quotas, roles, groups, data libraries, jobs managing, API keys). Finally, the data security layer of Laniakea represents a significant step in the direction of addressing the common issue posed by the analysis of sensitive data in public cloud infrastructures.

Future development directions for Laniakea will aim at improving the compatibility of Laniakea with a broader array of existing cloud setups, at extending cluster support to other resource managers (e.g., TORQUE [62], HTCondor [63]), and at widening the selection of tools to be made available on-demand beyond Galaxy.

## Methods

### INDIGO-DataCloud middleware

Laniakea builds on the INDIGO software catalogue [35, 64]. In particular:

- The Identity and Access Management (IAM) is an Authentication and Authorisation Infrastructure (AAI) service that manages user identities, attributes (e.g., affiliation and group membership), and authorization policies to access federated PaaS and manage heterogeneous and distributed resources through a single user account.
- The INDIGO PaaS [35, 65, 66] serves as the abstraction layer for the definition and provision of the resources required by users, managing the transparent deployment of virtual machines on OpenStack, OpenNebula, and commercial cloud providers. INDIGO PaaS processes requests in the form of TOSCA [45] omation templates. These are modular documents that use YAML syntax and Ansible roles to describe the properties and configuration of the virtual hardware and software components and the sequence of actions needed to achieve the deployment of a virtual environment.
- CLUES [67] is an elasticity manager for High Performance Computing (HPC) clusters that enables dynamic cluster resources scaling, deploying, and powering on of new working nodes depending on the current workload of the cluster and powering off and removal when they are no longer required.

We developed the set of TOSCA templates required to automate the installation and configuration of software and virtual hardware for Laniakea Galaxy instances [68–70].

### The Galaxy environment

- Galaxy production environment standard software stack: CentOS 7, PostgreSQL, Nginx, uWSGI, and ProFTPD. Apart from the operative system, for which there are no official recommendations, this configuration is rooted in the guidelines for production environments issued by the Galaxy Project [71]. We implemented the Ansible role required to orches-
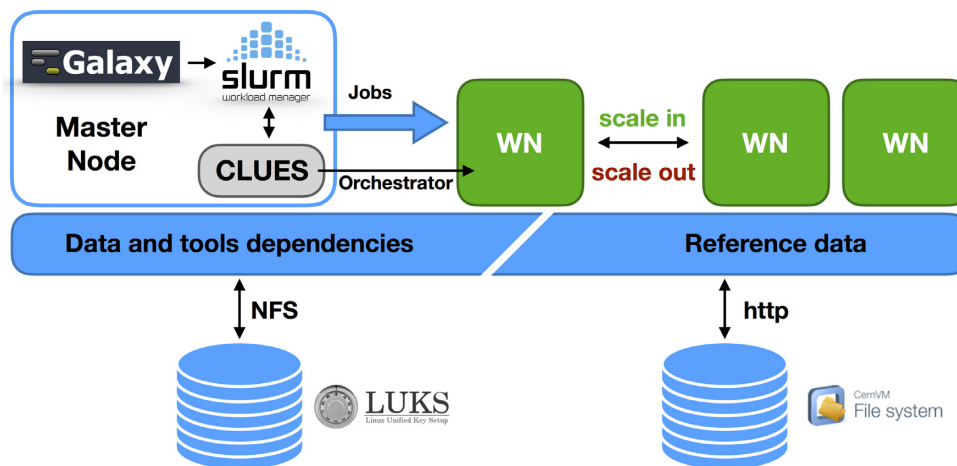
**Figure 8:** Galaxy elastic cluster architecture. Initially, only the master node, which hosts Galaxy, SLURM, and CLUES, is deployed. The SLURM queue is monitored by CLUES, and new worker nodes are deployed to process pending jobs up to the maximum number set during the cluster configuration, thus adapting resource availability to the current workload. The user home directory and persistent storage are shared among master and worker nodes through the Network File System (NFS), enabling the sharing of CONDA tool dependencies. The CVMFS shared volume is also mounted on each worker node to ensure that tools have access to reference data.

trate the automatic installation and configuration of the environment [72].

- Tools: Laniakea employs the official Galaxy Project library *Ephemeris* [73] to manage Galaxy tools installation. We developed an Ansible role [74] to use Ephemeris with Laniakea and the YAML recipes corresponding to the Galaxy flavours [75] currently available in Laniakea. Several wrappers have been developed for the GDC Somatic Variant Calling and CoVaCS Galaxy flavours and made available on the official Galaxy Test Tool Shed [76].
- CVMFS [77] is used by Laniakea to provide reference data to multiple Galaxy instances. Galaxy instances can both be linked to and use the official usegalaxy.org CVMFS repository [78] or use any other custom CVMFS repository. As a proof of concept, we set up a local CVMFS repository for those tools in the GDC Somatic Variant Calling and CoVaCS flavours requiring reference data not available in the main repository. We did automatize both the configuration of Galaxy to make use of a CVMFS server and the creation of new CVMFS repositories on the IaaS [79–81]. Finally, the *galaxyctl* python API [82] has been developed to manage and monitor the status of Galaxy instances from the dashboard of Laniakea.

### Docker containers

Galaxy dockers used by Laniakea are based on the official Galaxy Docker port [83]. We developed an Ansible role [84] to modify the official Galaxy Docker container and make readily available Laniakea Galaxy flavours also through this medium. All Laniakea Docker containers are available on DockerHub [85].

### Storage encryption

The encryption layer is based on LUKS [86], 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 reencryption of the whole device. We developed a bash script [87] to perform storage encryption in

Laniakea, an Ansible role [88] to automate the whole procedure, and the *luksctl* python package and API [89, 90] to let users easily create and manage encrypted volumes from the Dashboard.

### Hashicorp Vault

To let users securely store and access encryption passphrases and SSH private keys, Laniakea relies on Hashicorp Vault [56] secrets management software. Data stored on Vault are encrypted with a 256-bit Advanced Encryption Standard (AES) cipher in the Galois Counter Mode with a randomly generated nonce. The Laniakea's Hashicorp Vault configuration is available at [91].

### Laniakea dashboard

The Laniakea dashboard is based on the Orchestrator dashboard developed in the framework of the DEEP-HybridDataCloud H2020 project [92], using the Flask web micro-framework [93] and the Bootstrap 4 toolkit [94]. Laniakea dashboard code is available at [95].

## Availability of supporting source code and requirements

Project name: Laniakea
Project home page: https://laniakea-elixir-it.github.io
Operating system: Platform independent
Programming languages: JavaScript, Python, Shell, XML, YAML
Other requirements: Linux, Docker, INDIGO-Datacloud PaaS services, INDIGO IAM, Ansible.
License: All software developed for Laniakea is licensed under GPLv3, with the exception of Ansible roles that, being part of the INDIGO software catalogue, are released under the Apache-2.0 license.
Biotools ID: biotools: Laniakea
RRID:SCR_018146
The Laniakea web portal is available at https://laniakea-elixir-it.github.io. Source code and service configuration files are hosted on GitHub at https://github.com/Laniakea-elixir-it. Laniakea ansible roles, being part of the INDIGO source code, are hosted on the INDIGO GitHub repository at https://github.com/indigo-dc.

Complete documentation for Laniakea, comprising a step-by-step guide of Laniakea installation, is available at https://laniakea.readthedocs.io/en/latest/.

Laniakea docker containers are hosted on DockerHub at https://hub.docker.com/u/laniakeacloud.

## Supporting data

Snapshots of the code and supporting data are available in the *GigaScience* repository GigaDB [96].

## Additional files

Fila name: Laniakea_Supplementary.docx
Title: Additional Table S1 and Figures S1 and S2
Description: Supplementary Table S1 compares Galaxy-related features of Laniakea, GVL and Phenomenal. Supplementary Figure S1 shows the SSH Keys management interface. Supplementary Figure S2 displays the Galaxy implementation of the CoVaCS workflow.

## Abbreviations

AAI: Authentication and Authorisation Infrastructure; AES: Advanced Encryption Standard; AWS: Amazon Web Services; CVMFS: CernVM-FS; GDC: Genomic Data Commons; GVL: Genomic Virtual Laboratory; HPC: High Performance Computing; IaaS: Infrastructure as a Service; IAM: Identity and Access Management; INDIGO: INtegrating Distributed data Infrastructures for Global ExplOitation; LUKS: Linux Unified Key Setup; PaaS: Platform as a Service; SSH: Secure SHell; TOSCA: Topology and Orchestration Service for Cloud Applications; VM: Virtual Machine.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

## References

1. Attwood TK, Blackford S, Brazas MD, et al. A global perspective on evolving bioinformatics and data science training needs. Brief Bioinform 2019;**20**:398–404.

2. Via A, Attwood TK, Fernandes PL, et al. A new pan-European Train-the-Trainer programme for bioinformatics: Pilot results on feasibility, utility and sustainability of learning. Brief Bioinform 2019;**20**:405–15.

3. McGrath A, Champ K, Shang CA, et al. From trainees to trainers to instructors: Sustainably building a national capacity in bioinformatics training. PLoS Comput Biol 2019;**15**:1–12.

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

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

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

7. Cohen-Boulakia S, Belhajjame K, Collin O, et al. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. Futur Gener Comput Syst 2017;**75**:284–98. http://dx.doi.org/10.1016/j.future.2017.01.012

8. Afgan E, Baker D, Batut B, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res 2018;**46**:W537–44. http://dx.doi.org/10.1093/nar/gky379

9. Galaxy Project: Servers, clouds, and deployable resources. https://galaxyproject.org/use. Accessed 10 Mar 2020.

10. Euan A. Towards precision medicine. Nat Rev Genet 2016;**17**:507–22.

11. Lightbody G, Haberland V, Browne F, et al. Review of applications of high-throughput sequencing in personalized medicine: barriers and facilitators of future progress in research and clinical application. Brief Bioinform 2018;**20**:1795–811.

12. Batut B, Hiltemann S, Bagnacani A, et al. Community-driven data analysis training for biology. Cell Syst 2018;**6**:752–758.e1.

13. 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.

14. Langmead B, Nellore A. Cloud computing for genomic data analysis and collaboration. Nat Rev Genet 2018;**19**:208–19. http://www.nature.com/doifinder/10.1038/nrg.2017.113

15. Karim R, Michel A, Zappa A, et al. Improving data workflow systems with cloud services and use of open data for bioinformatics research. Brief Bioinform 2017;**19**:1035–50. http://fdslive.oup.com/www.oup.com/pdf/production_in_progress.pdf

16. Pavlovich M. Computing in biotechnology: omics and beyond. Trends Biotechnol 2017;**35**:479–80. http://dx.doi.org/10.1016/j.tibtech.2017.03.011

17. Warth B, Levin N, Rinehart D, et al. Metabolizing data in the cloud. Trends Biotechnol 2017;**35**:481–3. http://dx.doi.org/10.1016/j.tibtech.2016.12.010

18. Emami Khoonsari P, Moreno P, Bergmann S, et al. Interoperable and scalable data analysis with microservices: Applications in metabolomics. Bioinformatics 2019;**35**:3752–60.

19. Verderame L, Merelli I, Morganti L, et al. A secure cloud-edges computing architecture for metagenomics analysis. Futur Gener Comput Syst 2019; https://doi.org/10.1016/j.future.2019.09.013

20. Griebel L, Prokosch HU, Köpcke F, et al. A scoping review of cloud computing in healthcare. BMC Med Inform Decis Mak 2015;**15**:1–16.

21. Bellazzi R. Big data and biomedical informatics: A challenging opportunity. Yearb Med Inform 2014;**9**:8–13.

22. Liu B, Madduri RK, Sotomayor B, et al. Cloud-based

bioinformatics workflow platform for large-scale next-generation sequencing analyses. J Biomed Inform 2014;**49**:119–33. http://dx.doi.org/10.1016/j.jbi.2014.01.005

23. Afgan E, Baker D, Coraor N, et al. Galaxy CloudMan: Delivering cloud compute clusters. BMC Bioinformatics 2010;**11**:2–7.

24. Afgan E, Chapman B, Taylor J. CloudMan as a platform for tool, data, and analysis distribution. BMC Bioinformatics 2012;**13**:1.

25. Sefraoui O, Aissaoui M, Eleuldj M. OpenStack: Toward an open-source solution for cloud computing. Int J Comput Appl 2012;**55**:38–42. http://research.ijcaonline.org/volume55/number3/pxc3882991.pdf

26. Sotomayor B, Montero RS, Llorente IM, et al. Virtual infrastructure management in private and hybrid clouds. IEEE Internet Comput 2009;**13**:14–22. http://ieeexplore.ieee.org/document/5233608/

27. Afgan E, Sloggett C, Goonasekera N, et al. Genomics Virtual Laboratory: A practical bioinformatics workbench for the cloud. PLoS One 2015;**10**:1–20.

28. Peters K, Bradbury J, Bergmann S, et al. PhenoMeNal: Processing and analysis of metabolomics data in the cloud. Gigascience 2019;**8**:pii: giy149.

29. Krieger MT, Torreno O, Trelles O, et al. Building an open source cloud environment with auto-scaling resources for executing bioinformatics and biomedical workflows. Futur Gener Comput Syst 2017;**67**:329–40. http://dx.doi.org/10.1016/j.future.2016.02.008

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

31. Afgan E, Baker D, Coraor N, et al. Harnessing cloud computing with Galaxy Cloud. Nat Biotechnol 2011;**29**:972–4.

32. Caballer M, Zala S, García ÁL, et al. Orchestrating complex application architectures in heterogeneous clouds. J Grid Comput 2017;**16**:3–18.

33. Attardi G, DI Martino B, Esposito A, et al. Using federated cloud platform to implement academia services for research and administration. 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, 2018;413–8.

34. Laniakea Home Page. https://laniakea-elixir-it.github.io. Accessed 10 Mar 2020.

35. Campos DSI, Marco LGJ, Solagna DLP, et al. INDIGO-DataCloud: A platform to facilitate seamless access to e-infrastructures. J Grid Comput 2018;**16**:381–408. https://link.springer.com/article/10.1007%2Fs10723-018-9453-3

36. Salomoni D, Campos I, Gaido L, et al. INDIGO-Datacloud: foundations and architectural description of a platform as a service oriented to scientific computing. 2016;1–31. http://arxiv.org/abs/1603.09536

37. Merkel D. Docker: Lightweight Linux containers for consistent development and deployment. Linux J 2014;**2014**:2. http://dl.acm.org/ft_gateway.cfm?id=2600241&type=html%5Cnhttp://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment

38. Ansible documentation. https://docs.ansible.com/ansible/latest/index.html. Accessed 10 Mar 2020.

39. Ceccanti A, Hardt M, Wegh B, et al. The INDIGO-Datacloud authentication and authorization infrastructure. J Phys Conf Ser 2017;**898**:102016.

40. Linden M, Prochazka M, Lappalainen I, et al. Common elixir service for researcher authentication and authorisation. F1000Res 2018;**7**:1199.

41. EPIGEN Galaxy server. http://www.beaconlab.it/epigalaxy. Accessed 10 Mar 2020.

42. Grüning BA, Fallmann J, Yusuf D, et al. The RNA workbench: Best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. Nucleic Acids Res 2017;**45**:W560–6.

43. National Cancer Institute. Genomic Data Commons. DNA-Seq Somatic Variation. https://gdc.cancer.gov/node/246. Accessed 10 Mar 2020.

44. Chiara M, Gioiosa S, Chillemi G, et al. CoVaCS: A consensus variant calling system. BMC Genomics 2018;**19**:120. https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-018-4508-1

45. Binz T, Breitenbücher U, Kopp O, Leymann F. TOSCA: Portable Automated Deployment and Management of Cloud Applications. In: Bouguettaya A, Sheng Q, Daniel F , (eds). Advanced Web Services. New York, NY: Springer; 2014.

46. Langmead B, Trapnell C, Pop M, et al. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 2009;**10**:R25.

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

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

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

50. Sherry ST. dbSNP: The NCBI database of genetic variation. Nucleic Acids Res 2001;**29**:308–11. https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/29.1.308

51. Mills RE, Pittard WS, Mullaney JM, et al. Natural genetic variation caused by small insertions and deletions in the human genome. Genome Res 2011;**21**:830–9.

52. Auton A, Abecasis GR, Altshuler DM, et al. A global reference for human genetic variation. Nature 2015;**526**:68–74.

53. McKenna A, Hanna M, Banks E, et al. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome ResGenome Res 2010;**20**:1297–303.

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

55. Oechslin P. Making a faster cryptanalytic time-memory trade-off. In: Boneh D. (eds) Advances in Cryptology - CRYPTO 2003.Lecture Notes in Computer Science, vol 2729. Springer, Berlin, Heidelberg; 2003;617–30.

56. HashiCorp Vault. https://www.vaultproject.io. Accessed 10 Mar 2020.

57. ReCaS Bari Data Center. https://www.recas-bari.it/index.php/en. Accessed 10 Mar 2020.

58. Antonacci M, Bellotti R, Cafagna F, et al. The ReCaS Project: The Bari Infrastructure. In High Performance Scientific Computing Using Distributed Infrastructures; World Scientific: Singapore; 2017;17–33, ISBN 978-981-4759-70-0.

59. EOSC Marketplace: Laniakea@ReCaS https://marketplace.eosc-portal.eu/services/laniakea-recas. Accessed 10 Mar 2020.

60. Zanardi N, Morini M, Tangaro MA, et al. PIPE-T: A new Galaxy tool for the analysis of RT-qPCR expression data. Sci Rep 2019;**9**(1):17550.

61. Chiara M, Mandreoli P, Tangaro MA, et al. VINYL: Variant prIoritizatioN bY survivaL analysis. bioRxiv 2020.01.23.917229;

62. Staples G. TORQUE Resource Manager. Proc 2006 ACM/IEEE Conf Supercomput. New York, NY: ACM; 2006. http://doi.acm.org/10.1145/1188455.1188464

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

64. INDIGO DataCloud. INDIGO software catalogue https://github.com/indigo-dc. Accessed 10 Mar 2020.

65. Caballer M, Blanquer I, Moltó G, et al. Dynamic management of virtual infrastructures. J Grid Comput 2015;**13**:53–70.

66. EOSC Marketplace: INDIGO PaaS. https://marketplace.eosc-portal.eu/services/paas-orchestrator. Accessed 10 Mar 2020.

67. de Alfonso C, Caballer M, Alvarruiz F, et al. An energy management system for cluster infrastructures. Comput Electr Eng 2013;**39**:2579–90. https://linkinghub.elsevier.com/retrieve/pii/S0045790613001365

68. GitHub. INDIGO-DataCloud Tosca types. https://github.com/indigo-dc/tosca-types/blob/master/custom_types.yaml. Accessed 10 Mar 2020.

69. GitHub. INDIGO-DataCloud Tosca templates. https://github.com/indigo-dc/tosca-templates. Accessed 10 Mar 2020.

70. GitHub. Laniakea dashboard Tosca templates. https://github.com/Laniakea-elixir-it/laniakea-dashboard-config. Accessed 10 Mar 2020.

71. Galaxy Project: Galaxy production environment guidelines. https://docs.galaxyproject.org/en/latest/admin/production.html. Accessed 10 Mar 2020.

72. GitHub. Laniakea Ansible for Galaxy installation. https://github.com/indigo-dc/ansible-role-galaxycloud. Accessed 10 Mar 2020.

73. Galaxy Project: Ephemeris libraries. https://github.com/galaxyproject/ephemeris. Accessed 10 Mar 2020.

74. GitHub. Laniakea Ansible role for Ephemeris. https://github.com/indigo-dc/ansible-role-galaxycloud-tools. Accessed 10 Mar 2020.

75. GitHub. Laniakea Galaxy flavours. https://github.com/indigo-dc/Galaxy-flavors-recipes. Accessed 10 Mar 2020.

76. Laniakea Galaxy Test Tool Shed repository. https://tinyurl.com/elixir-it-tts. Accessed 10 Mar 2020.

77. Buncic P, Aguado Sanchez C, Blomer J, et al. CernVM: A virtual software appliance for LHC applications. J Phys Conf Ser 2010;**219**:042003.

78. Galaxy Project: usegalaxy.org reference data. https://galaxyproject.org/admin/reference-data-repo. Accessed 10 Mar 2020.

79. GitHub. Laniakea Ansible role for CVMFS configuration. https://github.com/indigo-dc/ansible-role-galaxycloud-refdata. Accessed 10 Mar 2020.

80. GitHub. Laniakea Ansible role for CVMFS server. https://github.com/indigo-dc/ansible-role-cvmfs-server. Accessed 10 Mar 2020.

81. GitHub. Laniakea Ansible role for CVMFS client. https://github.com/indigo-dc/ansible-role-cvmfs-client. Accessed 10 Mar 2020.

82. GitHub. Laniakea galaxyctl Python API. https://github.com/Laniakea-elixir-it/galaxyctl. Accessed 10 Mar 2020.

83. GitHub. Official Galaxy Docker port. https://github.com/bgruening/docker-galaxy-stable. Accessed 10 Mar 2020.

84. GitHub. Laniakea Ansible role for Galaxy Docker. https://github.com/indigo-dc/ansible-role-galaxycloud-docker. Accessed 10 Mar 2020.

85. DockerHub. Laniakea repository. https://hub.docker.com/u/laniakeacloud. Accessed 10 Mar 2020.

86. Fruhwirth C. New methods in hard disk encryption. Master's thesis, Vienna University of Technology 2005.

87. GitHub. Laniakea storage encryption script. https://github.com/Laniakea-elixir-it/fast-luks. Accessed 10 Mar 2020.

88. Github. Laniakea Ansible role for storage encryption. https://github.com/indigo-dc/ansible-role-galaxycloud-os. Accessed 10 Mar 2020.

89. GitHub. Laniakea luksctl management. https://github.com/Laniakea-elixir-it/luksctl. Accessed 10 Mar 2020.

90. GitHub. Laniakea lucksctl management API. https://github.com/Laniakea-elixir-it/luksctl_api. Accessed 10 Mar 2020.

91. GitHub. HashiCorp Vault configuration for Laniakea. https://github.com/Laniakea-elixir-it/vault_config. Accessed 10 Mar 2020.

92. Deep-Hybrid-DataCloud Project. https://deep-hybrid-datacloud.eu. Accessed 10 Mar 2020.

93. Flask. https://flask.palletsprojects.com/en/1.1.x. Accessed 10 Mar 2020.

94. Bootstrap 4. https://bootstrapcreative.com/shop/bootstrap-4-toolkit. Accessed 10 Mar 2020.

95. GitHub. Laniakea dashboard. https://github.com/Laniakea-elixir-it/orchestrator-dashboard/tree/laniakea-stable. Accessed 10 Mar 2020.

96. Tangaro MA, Donvito G, Antonacci M, et al. Supporting data for "Laniakea: an open solution to provide Galaxy "on-demand" instances over heterogeneous cloud infrastructures" GigaScience Database 2020. http://dx.doi.org/10.5524/100718.