

Article Title

Infrastructures for High-Performance Computing: Cloud Infrastructures

Author Contact Information

Fabrizio Marozzo
c/o DIMES – University of Calabria (corresponding author)
Via P. Bucci 42c
87036 Rende (CS)
Italy
Email: fmarozzo@dimes.unical.it
Telephone: +39 0984 494782

Abstract

This article describes the main services provided by the most popular cloud infrastructures currently in use. The systems presented in this article are either public clouds - Amazon Web Services (AWS), Google Cloud Platform, Microsoft Azure and IBM Bluemix - or private clouds - OpenStack, OpenNebula and Eucalyptus. The formers provide a variety of services that implement all the three main cloud service models (IaaS, PaaS and SaaS), while the latter are focused on the IaaS model, as they provide services for the management of large pools of processing, storage, and networking resources in a data center, by exploiting a variety of virtualization techniques.

Keywords

Public Cloud, Private Cloud; Amazon Web Services (AWS); Google Cloud Platform; Microsoft Azure; IBM Bluemix; OpenStack; OpenNebula; Eucalyptus.

1. Introduction

This article describes the main services provided by the most popular cloud infrastructures currently in use. The systems presented in this article are either public clouds - *Amazon Web Services (AWS), Google Cloud Platform, Microsoft Azure and IBM Bluemix* - or private clouds - *OpenStack, OpenNebula and Eucalyptus*.

All the public cloud systems provide services according to the Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) models (*Talia, Trunfio and Marozzo 2015*). With the IaaS model, the cloud system provide services that allow customers to rent virtualized resources like storage and servers; the PaaS model allows developers to focus on deploying of applications, since cloud providers are in charge of managing the underlying infrastructure; with the SaaS model, software and data are provided to customers as ready-to-use services.

As examples of IaaS services, AWS provides the popular *Elastic Compute Cloud (EC2)* service that allows creating and running virtual servers; Google Cloud Platform allows developers to run virtual machines through its *Compute Engine* service; Microsoft Azure provides a *VM role* service to manage virtual-machine images; IBM Bluemix *Virtual servers* can be used to deploy and scale virtual machines on demand.

As examples of PaaS services, AWS provides *Elastic Beanstalk* to create, deploy, and manage applications using a large set of infrastructure services; Google Cloud Platform includes *App Engine*, a PaaS that allows developers to create and host web applications in Google-managed data centers; Microsoft Azure's PaaS

includes a *Web role* for developing Web-based applications and a *Worker role* for batch applications; IBM Bluemix includes *Mobile Foundation*, a set of APIs to manage the life cycle of mobile applications.

Finally, some examples of SaaS services are *Simple Email Service* (AWS), *Gmail* (Google Cloud Platform), *Office 365* (Microsoft Azure) and *Natural Language Understanding* (IBM Bluemix).

In general, the private cloud systems are more focused on the IaaS model, as they provide services for the management of large pools of processing, storage, and networking resources in a data center, by exploiting a variety of virtualization techniques. For example, OpenStack with its *Compute* component provides virtual servers and with its *Storage* component provides a scalable and redundant storage system; OpenNebula's *Core component* creates and controls virtual machines by interconnecting them with a virtual network environment; Eucalyptus includes a *Node Controller* that hosts the virtual machine instances and manages the virtual network endpoints, and a *Storage Controller* that provides persistent block storage and allows the creation of snapshots of volumes.

2. Amazon Web Services

Amazon offers compute and storage resources of its IT infrastructure to developers in the form of Web services. Amazon Web Services (AWS)¹ is a large set of cloud services that can be composed by users to build their SaaS applications or integrate traditional software with cloud capabilities (see **Figure 1**). It is simple to interact with these services since Amazon provides SDKs for the main programming languages and platforms (e.g., Java, .Net, PHP, Android).

AWS includes the following main services:

- Compute: *Elastic Compute Cloud* (EC2) allows creating and running virtual servers; *Amazon Elastic MapReduce* for building and executing MapReduce applications.
- Storage: *Simple Storage Service* (S3), which allows storing and retrieving data via the Internet.
- Database: *Relational Database Service* (RDS) for relational tables; *DynamoDB* for non-relational tables; *SimpleDB* for managing small datasets; *ElasticCache* for caching data.
- Networking: *Route 53*, a DNS Web service; *Virtual Private Cloud* for implementing a virtual network.
- Deployment and Management: *CloudFormation* for creating a collection of ready-to-use virtual machines with pre-installed software (e.g., Web applications); *CloudWatch* for monitoring AWS resources; *Elastic Beanstalk* to deploy and execute custom applications written in Java, PHP and other languages; *Identity and Access Management* to securely control access to AWS services and resources.
- Content delivery: *Amazon CloudFront* makes easy to distribute content via a global network of edge locations.
- App services: *Simple Notification Service* to notify users; *Simple Queue Service* that implements a message queue; *Simple Workflow Service* to implement workflow-based applications.

¹ Amazon Web Services, <https://aws.amazon.com/>

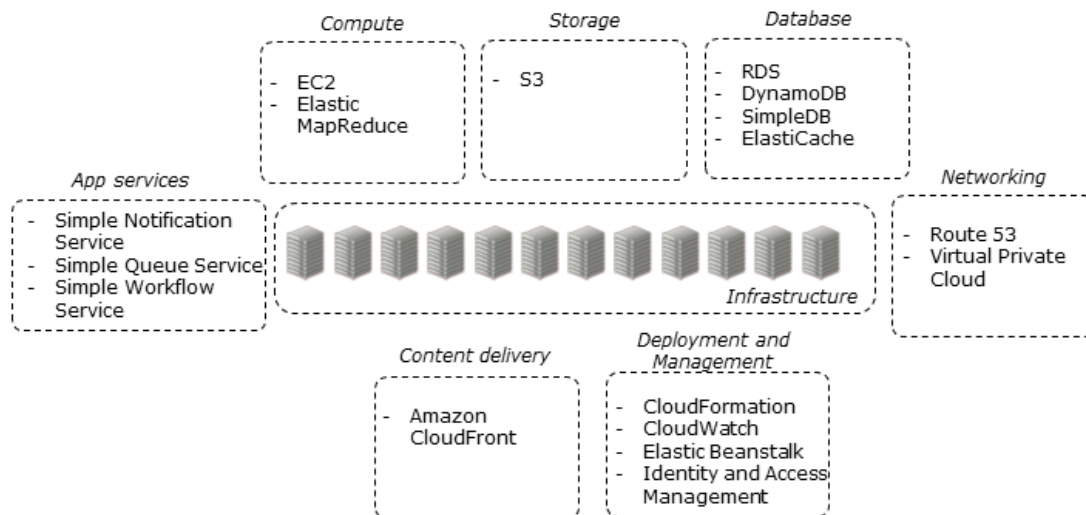


Figure 1. Amazon Web Services.

Even though Amazon is best known to be the first IaaS provider (based on its EC2 and S3 services), it is now also a PaaS and SaaS provider, with services like *Elastic Beanstalk*, which allows users to quickly create, deploy, and manage applications using a large set of AWS services, *Amazon Machine Learning*, which provides visualization tools and wizards for easily creating machine learning models, and *Simple Email Service* providing a basic email-sending service.

3. Google Cloud Platform

Google Cloud Platform² is a set of cloud computing services provided by Google. The platform allows developers to define a wide range of web applications, from simple websites to scalable multitenancy applications. The services created by users run in the same supporting infrastructure that Google uses internally to run Google Search, YouTube, Maps, and Gmail. Google provides SDKs to define applications using several languages, libraries, and frameworks (e.g., GO, PHP, Java, Python and .Net).

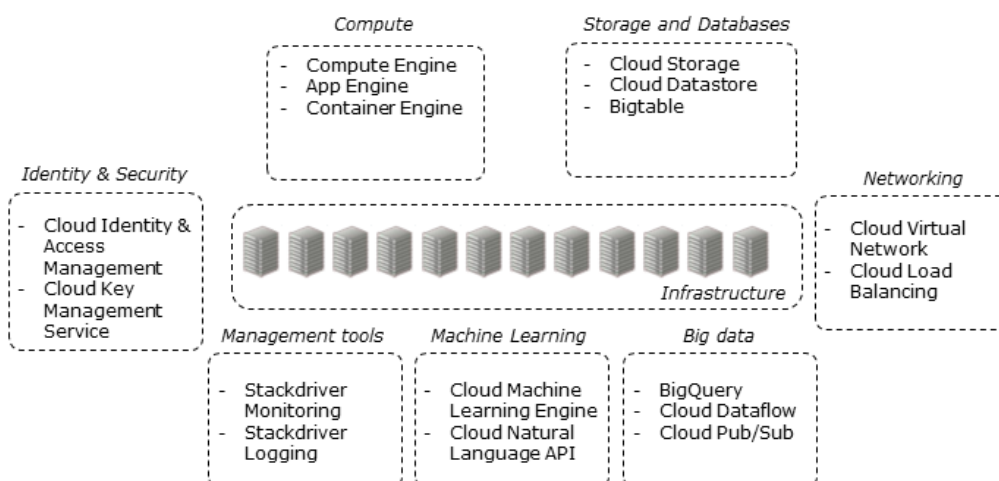


Figure 2. Google Cloud Platform.

The platform provides several services to build and scale an application (see **Figure 2**):

² Google Cloud Platform, <https://cloud.google.com/>

- Compute: *Compute Engine*, an IaaS that allows developers to run virtual machine (VM) instances; *App Engine*, a PaaS that allows developers to create and host web applications in Google-managed data centers; *Container Engine*, a management and orchestration system for software container.
- Storage and Databases: *Cloud Storage*, a cloud storage web service for storing and accessing large and/or unstructured data sets; *Cloud Datastore*, a highly scalable NoSQL database service; *Cloud Bigtable*, a highly scalable NoSQL database service used by Google to power many core services, including Search, Analytics, Maps, and Gmail.
- Networking: *Cloud Virtual Network*, a managed networking functionality for web applications; *Cloud Load Balancing*, a load-balancer for compute resources in single or multiple regions.
- Big Data: *BigQuery*, a data warehouse enabling fast SQL queries; *Cloud Dataflow*, a data processing service for stream and batch execution of pipelines; *Cloud Pub/Sub*, a real-time messaging service to send and receive messages between applications.
- Machine Learning: *Cloud Machine Learning Engine*, a scalable machine learning service to train models and predict new data; *Cloud Natural Language API*, for extracting information mentioned in text documents, news articles or blog posts.
- Management Tools: *Stackdriver Monitoring*, to monitor cloud resources performance; *Stackdriver Logging*, for storing, monitoring, and alerting log data and events from Google services.
- Identity and Security: *Cloud Identity & Access Management*, a service to establish a policy of access to the various services; *Cloud Key Management Service*, a key management for managing encryption of cloud services.

4. Microsoft Azure

Azure³ is an environment and a set of cloud services that can be used to develop cloud-oriented applications, or to enhance existing applications with cloud-based capabilities. The platform provides on-demand compute and storage resources exploiting the computational and storage power of the Microsoft data centers. Azure is designed for supporting high availability and dynamic scaling services that match user needs with a pay-per-use pricing model. The Azure platform can be used to perform the storage of large datasets, execute large volumes of batch computations, and develop SaaS applications targeted towards end-users.

Microsoft Azure includes three basic components/services as shown in **Figure 3**:

- *Compute* is the computational environment to execute cloud applications. Each application is structured into roles: *Web role*, for Web-based applications; *Worker role*, for batch applications; *VM role*, for virtual-machine images.
- *Storage* provides scalable storage to manage: binary and text data (*Blobs*), non-relational tables (*Tables*), queues for asynchronous communication between components (*Queues*), and virtual disks (*Disks*).
- *Fabric controller* whose aim is to build a network of interconnected nodes from the physical machines of a single data center. The Compute and Storage services are built on top of this component.

³ Microsoft Azure, <https://www.microsoft.com/azure>

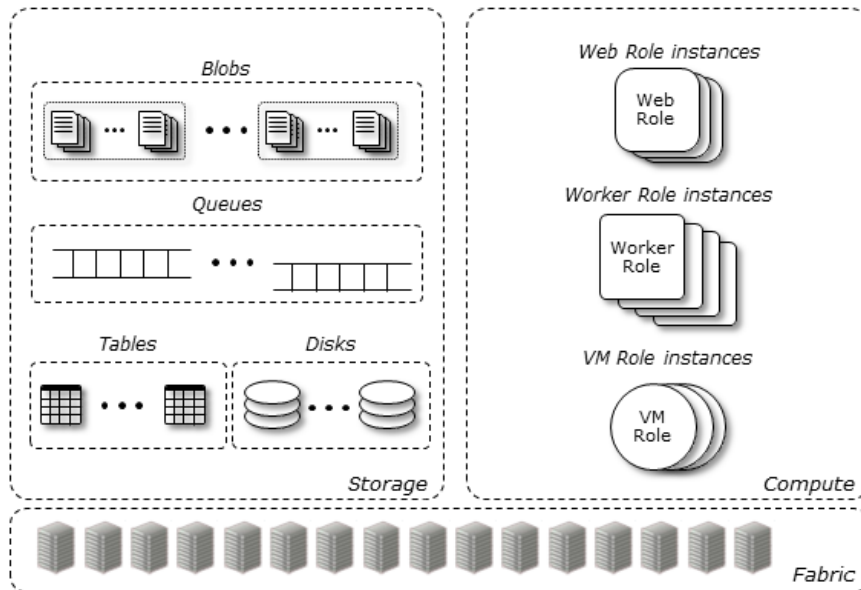


Figure 3. Microsoft Azure.

Microsoft Azure provides also a set of services that can be used in many fields, like *Machine Learning*, to build cloud-based machine learning applications, or *IOT Hub*, to connect, monitor, and control IoT devices. Azure provides standard interfaces that allow developers to interact with its services. Moreover, developers can use IDEs like Microsoft Visual Studio and Eclipse to easily design and publish Azure applications.

5. IBM Bluemix

IBM Bluemix⁴ delivers services that can easily integrate with cloud applications without users needing to know how to install or configure them. It supports several programming languages and services for building, running and managing applications on the cloud (e.g., NET, Java, Node.js, PHP, Python, Ruby, Go).

With Bluemix customers can access three types of cloud computing platforms:

- *Bluemix public*, which provides all the necessary resources for an application development available to the general public;
- *Bluemix dedicated*, which provides a own Bluemix private environment that is hosted in an isolated container managed by Bluemix;
- *Bluemix Hybrid*, which allows customers to connect their dedicated services to the Public Bluemix services by maintaining data and sensitive services on premise.

IBM Bluemix offers mainly two compute services for defining applications: *Containers* and *OpenStack VMs*. *IBM Containers* permits developers to deploy, manage and run application on IBM platform by leveraging open-source Docker container technology⁵ (Merkel 2014). Docker is an open platform used for building, shipping and running applications. *OpenStack Virtual Machines* allows running specific operating systems that users can customize.

The IBM Bluemix platform provides several services to build and scale an application (see **Figure 4**):

- *Compute: Virtual servers* to deploy and scale virtual machines on demand; *GPU Servers* to deploy high-performance GPU servers; *Containers*, to deploy, manage and run application on IBM platform by leveraging open-source Docker container technology.

⁴ IBM Bluemix, <https://www.ibm.com/cloud-computing/bluemix/>

⁵ Docker, <https://www.docker.com/>

- *Storage: Object Storage*, to access unstructured data via with RESTful APIs; *File Storage*, a full-featured file storage; *Block Storage*, a scalable block storage services.
- *Network: Load Balancing*, a load-balancer for compute resources; *Direct Link*, to transfer data between a private infrastructure and IBM Bluemix services.
- *Mobile: Swift*, to write Swift mobile applications; *Mobile Foundation*, a set of APIs to manage the app life cycle.
- *Analytics: Analytics Services*, to extract insight from data; *Natural Language Understanding*, to extract meta-data from text such as sentiment, and relations.
- *Internet of Things: IOT platform*, which allows to communicate with and consume data from IOT devices.
- *Security: Firewalls* for protecting from malicious activities; *Security Software*, an intrusion protection system for the network and server/host level.

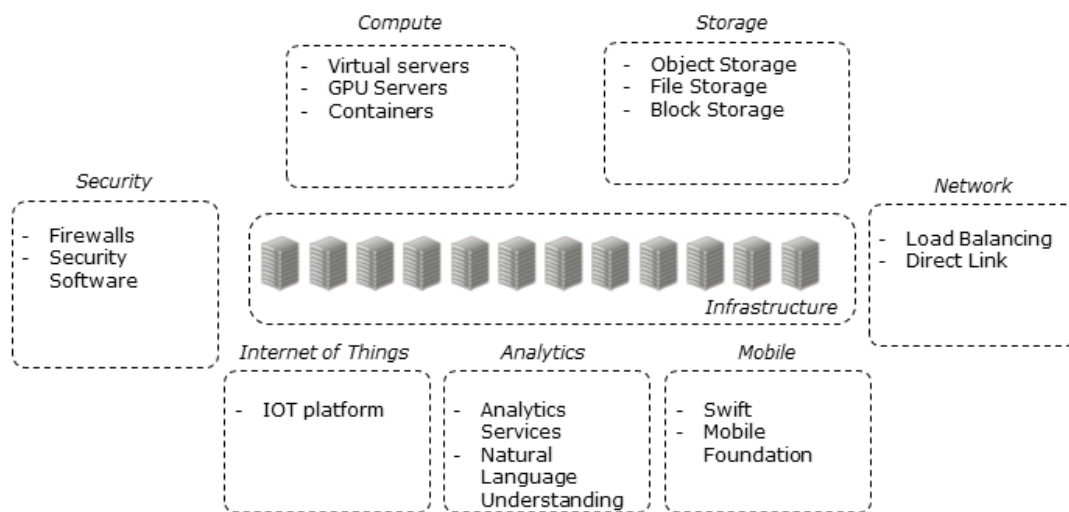


Figure 4. IBM Bluemix.

6. OpenStack

OpenStack⁶ is a cloud system that allows the management of large pools of processing, storage, and networking resources in a data center through a Web-based interface (Wen, Gu, Li, Gao, Zhang, 2012). The system has been designed, developed and released following four open principles:

- *Open source:* OpenStack is released under the terms of the Apache License 2.0;
- *Open design:* Every six months there is a design summit to gather requirements and define new specifications for the upcoming release;
- *Open development:* A publicly available source code repository is maintained for the entire development process;
- *Open Community:* Most decisions are made by the OpenStack community using a lazy consensus model.

The modular architecture of OpenStack is composed by four main components, as shown in **Figure 5**.

⁶ OpenStack, <https://www.openstack.org/>

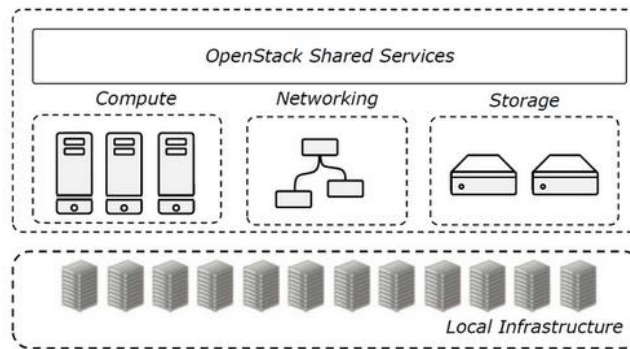


Figure 5. OpenStack.

OpenStack *Compute* provides virtual servers upon demand by managing the pool of processing resources available in the data center. It supports different virtualization technologies (e.g., VMware, KVM) and is designed to scale horizontally. OpenStack *Storage* provides a scalable and redundant storage system. It supports Object Storage and Block Storage: the former allows storing and retrieving objects and files in the data center; the latter allows creating, attaching and detaching of block devices to servers. OpenStack *Networking* manages the networks and IP addresses. Finally, OpenStack *Shared Services* are additional services provided to ease the use of the data center. For instance, Identity Service maps users and services, Image Service manages server images, Database Service provides a relational database.

7. OpenNebula

OpenNebula⁷ (Sotomayor, Montero, Llorente and Foster 2009) is an open-source framework mainly used to build private and hybrid clouds. The main component of the OpenNebula architecture (see Figure 6) is the *Core*, which creates and controls virtual machines by interconnecting them with a virtual network environment. Moreover, the *Core* interacts with specific storage, network and virtualization operations through pluggable components called *Drivers*. In this way, OpenNebula is independent from the underlying infrastructure and offers a uniform management environment.

The *Core* also supports the deployment of *Services*, which are a set of linked components (e.g., Web server, database) executed on several virtual machines. Another component is the *Scheduler*, which is responsible for allocating the virtual machines on the physical servers. To this end, the *Scheduler* interacts with the *Core* component through appropriate deployment commands.

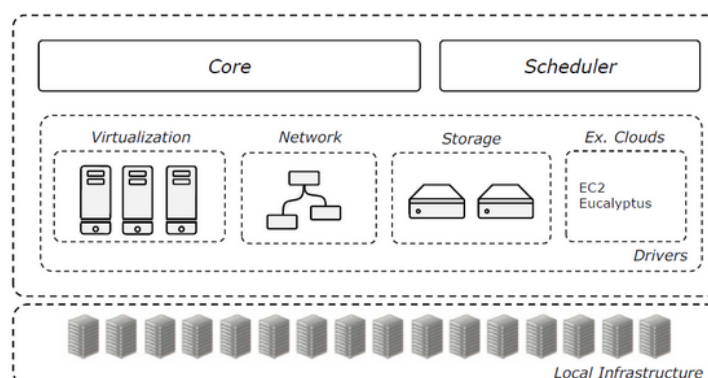


Figure 6. OpenNebula.

⁷ OpenNebula, <https://opennebula.org/>

OpenNebula can implement a hybrid cloud using specific Cloud Drivers that allow interacting with external clouds. In this way, the local infrastructure can be supplemented with computing and storage resources from public clouds. Currently, OpenNebula includes drivers for using resources from Amazon EC2 and Eucalyptus.

8. Eucalyptus

Eucalyptus⁸ (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) is an open source and paid software platform for implementing IaaS in a private or hybrid cloud computing environment (Endo, Gonçalves, Kelner, Sadok 2010). Eucalyptus enables developers to use compute, storage, and network resources that can be dynamically scaled and can be used to build web-based applications. Companies can use AWS-compatible services, images, and scripts on their own on-premises IaaS environments, without using a public Cloud environment like AWS. It is simple to interact with Eucalyptus since it provides a console to self-service provision and configure compute, network, and storage resources.

Eucalyptus is based on five basic components as shown in **Figure 7**. The *Node Controller (NC)* hosts the virtual machine instances and manages the virtual network endpoints. The *Cluster Controller (CC)* acts as the front end for a cluster and is responsible for deploying and managing instances on NCs. The *Storage Controller (SC)* provides persistent block storage and allows the creation of snapshots of volumes. The *Cloud Controller (CLC)* is the front-end for managing cloud services and performs high-level resource scheduling and system accounting. CLC also allows interacting with the components of the Eucalyptus infrastructure providing an AWS-compatible web services interface to customers. *Walrus Storage Controller (WS3)* offers a persistent storage to all of the virtual machines in the Eucalyptus cloud and can be used to store machine images and snapshots. WS3 can be used with its AWS S3 APIs.

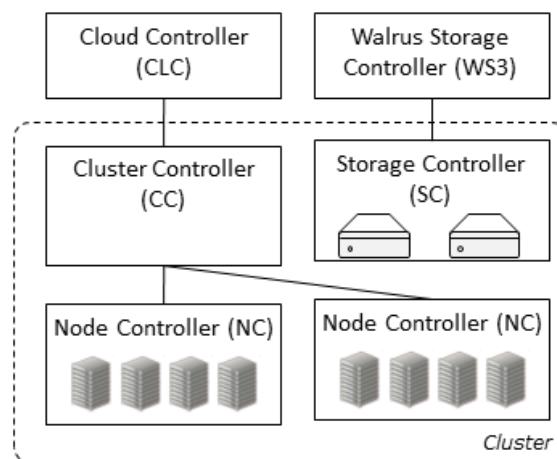


Figure 7. Eucalyptus.

9. Closing Remarks

This article described the main services provided by the most popular cloud infrastructures currently in use. The systems presented in this article were either public clouds - Amazon Web Services (AWS), Google Cloud Platform, Microsoft Azure and IBM Bluemix - or private clouds - OpenStack, OpenNebula and Eucalyptus. As discussed above, the public cloud systems provide a variety of services that implement all the three main cloud service models, i.e., Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). On the other hand, the private cloud systems are more focused on the IaaS model, as they provide services for the management of large pools of processing, storage, and networking resources in a data center, by exploiting a variety of virtualization techniques.

Cross References

⁸ Eucalyptus , <http://www.eucalyptus.com/>

Infrastructures for High-Performance Computing: Cloud Computing
Infrastructures for High-Performance Computing: Cloud Computing Development Environments

References

Endo, P. T., Gonçalves, G. E., Kelner, J., & Sadok, D. (2010). A survey on open-source cloud computing solutions. In Brazilian Symposium on Computer Networks and Distributed Systems (Vol. 71).

Merkel, D. (2014) "Docker: lightweight linux containers for consistent development and deployment." Linux Journal 2014.239.

Sotomayor, B., Montero, R. S., Llorente, I. M. and Foster, I. (2009), "Virtual Infrastructure Management in Private and Hybrid Clouds", IEEE Internet Computing, vol. 13, 14-22.

Talia, D., Trunfio P. and Marozzo, F. (2015). Data analysis in the Cloud. Elsevier, The Netherlands.

Wen, X., Gu, G., Li, Q., Gao, Y., & Zhang, X. (2012). Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In Fuzzy Systems and Knowledge Discovery (FSKD), 9th International Conference on (pp. 2457-2461).