Cloud Computing [131P15C302]

~ Asst. Prof. Shivkumar Chandey
M.Sc. Computer Science

Objectives

At the end of Cloud Computing course, the students will:

1. Have an insight into the basics of cloud computing along with virtualization.
2. Have a conceptual understanding of cloud computing and will be in the position to assess their application objectives and decide how to deploy their application in the cloud with ease.
3. Understand the security aspects in cloud.

Unit II Syllabus

Unit II	Cloud Types and Models: Open-Source Cloud Implementation and Administration: Cloud Deployment Techniques:

Text Books and References

- **Textbooks:**

1. Cloud Computing Black Book, Dreamtech Press, 2014
2. Mastering Cloud Computing - Technologies and Applications Programming, 2014

- **Reference Books:**

1. Cloud Computing: Concepts, Technology & Architecture, Pearson, 2013
2. Cloud and Distributed Computing: Algorithms and Systems, Wiley

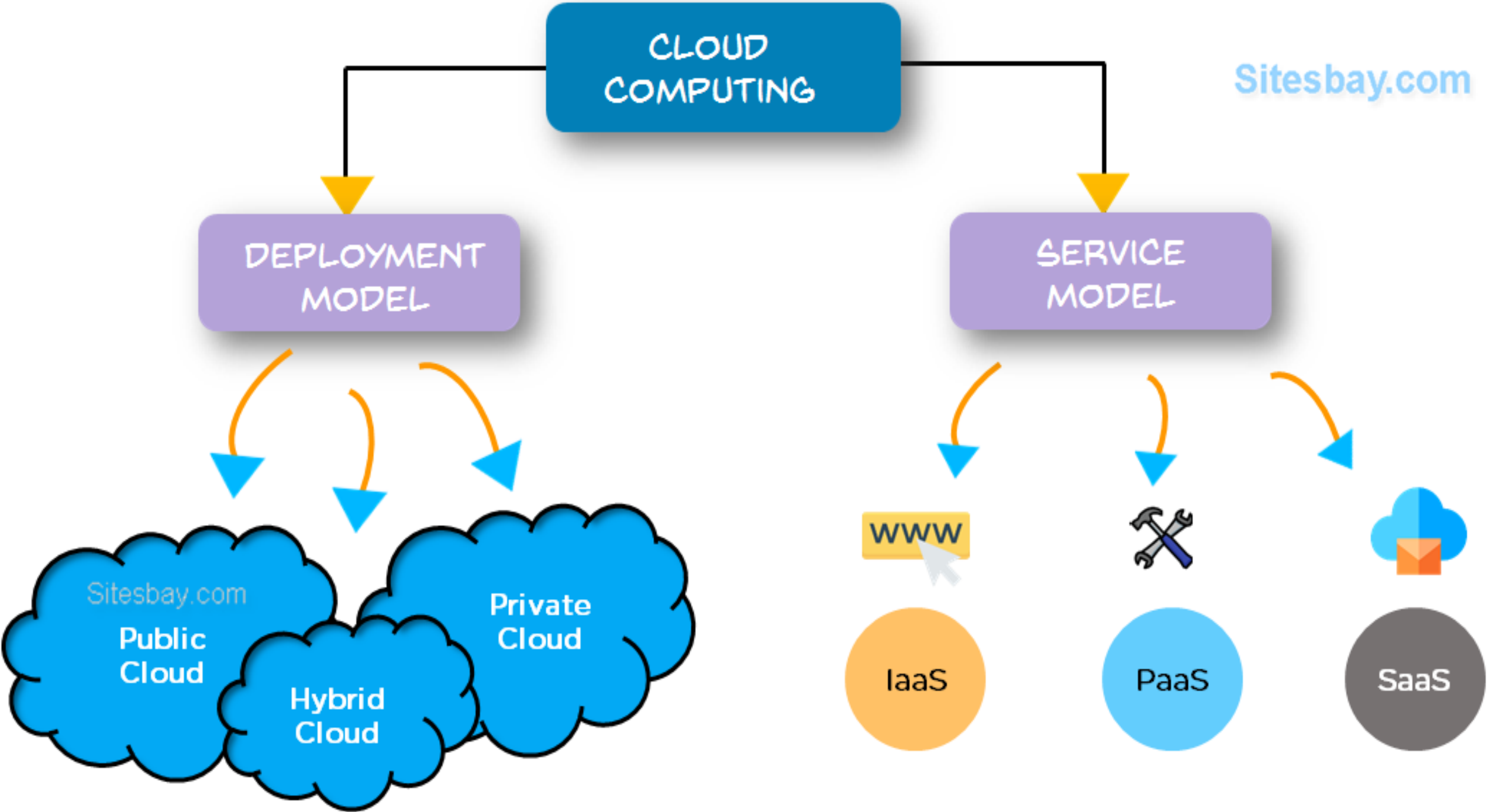
Course Learning Outcomes

1. Student will have basic understanding about cloud and virtualization along with it how one can migrate over it.
2. Students can identify the various levels of services that can be achieved by cloud
3. Student can understand the issues related to Cloud Computing.
4. Students will recognize the administrative challenges in Cloud Computing.

List of Practicals

1. Implementation of Bare-metal and hosted virtualization
2. Implementation of containerization using docker
3. Demonstration of IaaS cloud
4. Demonstration of PaaS cloud
5. Demonstration of SaaS cloud
6. Implementation of Cloud services on Open stack cloud platform
7. Implementation of Cloud services on Amazon web services
8. Demonstration of data analytics in Cloud

Cloud Types and Models


CLOUD COMPUTING
Sitesbay.com
DEPLOYMENT MODEL
SERVICE MODEL
Sitesbay.com
Public Cloud
Hybrid Cloud
Private Cloud
www
IaaS
PaaS
SaaS

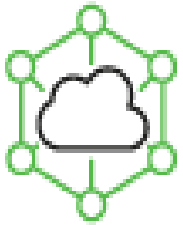
Types of Cloud Models/ Service Models

IaaS vs PaaS vs SaaS

Infrastructure as a service	Platform as a service	Software as a service
A service model in cloud computing that provides virtualized computing resources over the internet	A cloud computing model that delivers tools necessary for application development over the internet	A service model in cloud computing that hosts software and makes them available for clients over the internet
Provides access to resources such as virtual machines, virtual storage etc.	Provides runtime environments, development & deployment tools for applications	Provides software as services to the end users
Server storage network OS & Middleware packaged software	Server storage network OS & Middleware	Server storage network
Used by network architects	Used by developers	Used by end users

Types of Cloud/ Deployment Models

Types of Cloud Deployment



Public Cloud

Typically have massive amounts of available space, which translates into easy scalability. Recommended for software development and collaborative projects.

Private Cloud

Usually reside behind a firewall and are utilized by a single organization. Recommended for businesses with very tight regulatory requirements

Hybrid Cloud

Combine public clouds with private clouds to allow the two platforms to interact seamlessly. Recommended for businesses balancing big data analytics with strict data privacy regulations.

Community Cloud

A collaborative, multi-tenant platform used by several distinct organizations to share the same applications. Users are typically operating within the same industry or field.

Manufacturing organization has its own private cloud

Manufacturing organization shares cloud with general public

Combination of cloud deployment models

Manufacturing organization shares cloud with other organizations with similar interests

List of Open-Source Cloud Platforms and Tools

1] OpenStack

- OpenStack is used for cloud computing. It is used as an infrastructure as a service (Iaas). The software is meant to create private and public clouds. Users will be able to control compute, storage and networking resources throughout a data centre. This is operated from a dashboard or through OpenStack's API. Backed by the most influential companies in software development and hosting, with thousands of individual community members, many think that OpenStack is the future of cloud computing.

Key Features

- Services: Messaging, containers, clustering, compute, identity, metadata indexing as service, events, workflows, DNS, bare-metal provisioning, governance, benchmarking, optimization and deployment
- Web front-end, Big Data processing framework, container orchestration engine and NFV orchestration.

2] Eucalyptus

- Eucalyptus is open-source software for creating AWS-compatible private and hybrid clouds. It is also a Linux-based software architecture that performs scalable private and hybrid clouds within your IT infrastructure. It lets you use your collections of resources using a self-service interface whenever needed. As an infrastructure as a Service (IaaS), Eucalyptus also enables your users to provision your compute and storage resources on demand.

Key Features

- Works with various hypervisors.
- Communication within internal processes is guarded through SOAP and WS-Security.
- Administrative features like user and group management and reports.

3] OpenNebula

- OpenNebula is a flexible turnkey open-source solution to create Private Clouds and maintain Data Center virtualisation. It implements IaaS. The first open-source version was released in March 2008. It is intended to be a simple but feature-rich customisable solution to manage enterprise clouds. It is also simple to install, update and operate by the administrators; and easy to use by end-users.

Key Features

- Supports APIs like AWS EC2 and OGF OCCI.
- It supports SSH and X.509 for security and even supports token login functionality.
- Powerful UNIX based CLI for administration.

4] Apache CloudStack

- It is known to be a top-level project of the Apache Software Foundation. Apache CloudStack maintains networks of virtual machines as an Infrastructure as a Service (IaaS). CloudStack is a multi-hypervisor, high-availability cloud management platform. It is software that provides a cloud Orchestration layer, providing automation of the creation, provisioning and configuration of IaaS components.

Key Features

- Self-service user interface: AJAX console access, network virtualisation, multi-role support usage metering, virtual routers.
- LVM support: Block storage volumes, LDAP integration, OpenStack Swift integration, and domains and delegated administration.

5] VirtEngine

- VirtEngine can be utilized to build private or public clouds supporting IaaS, PaaS and SaaS. It allows customers to make use of applications in a few clicks. VirtEngine has a lot of applications, and a simple user interface for you to self-serve your needs. VirtEngine by DET.io is open as two separate solutions for the public and private cloud. The Public Cloud lets users build their own cloud and present servers to customers. The Public Cloud is open as a minified edition and a complete solution. The private cloud is an open-source and free solution for enterprises supporting HA and more enterprise features.

Key Features

- Access control, cost management, multi-cloud management, one-click apps, demand and supply monitoring and automatic launch.
- DNS support, cloud-native, multi-locations, Docker containers, cloud virtual machines, self-healing and migration tools.

Conclusion

- An open-source cloud platform is better than a proprietary cloud platform mainly because of the cost and the flexibility to choose from a variety of frameworks, clouds, and services.
- It is important to be sure of your personal and professional needs before selecting the deployment model. This will make your transition and maintenance much smoother and more manageable.
- If you are not satisfied with an open-source cloud platform today, you can go to another tomorrow.

Open Source Cloud Implementation and Administration

- Open-Source Eucalyptus Cloud Architecture
- Open-Source OpenStack Cloud Architecture
- Cloud Administration and Management
- Bundling or Uploading Virtual Machine Images on the Cloud Controller
- GUI Access to VM Instances over SSH

Eucalyptus

- Eucalyptus is an open source software platform for implementing Infrastructure as a Service (IaaS) in a private or hybrid cloud computing environment.
- Eucalyptus cloud computing architecture is highly scalable because of its distributed nature.

Eucalyptus

- The Eucalyptus cloud platform pools together existing virtualized infrastructure to create cloud resources for infrastructure as a service, network as a service and storage as a service.
- The name Eucalyptus is an acronym for **Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems.**

Eucalyptus

- Eucalyptus was founded out of a research project in the Computer Science Department at the University of California, Santa Barbara, and became a for-profit business called Eucalyptus Systems in 2009.
- Eucalyptus Systems announced a formal agreement with Amazon Web Services (AWS) in March 2012, allowing administrators to move instances between a Eucalyptus private cloud and the Amazon Elastic Compute Cloud (EC2) to create a hybrid cloud.
- The partnership also allows Eucalyptus to work with Amazon's product teams to develop unique AWS-compatible features.

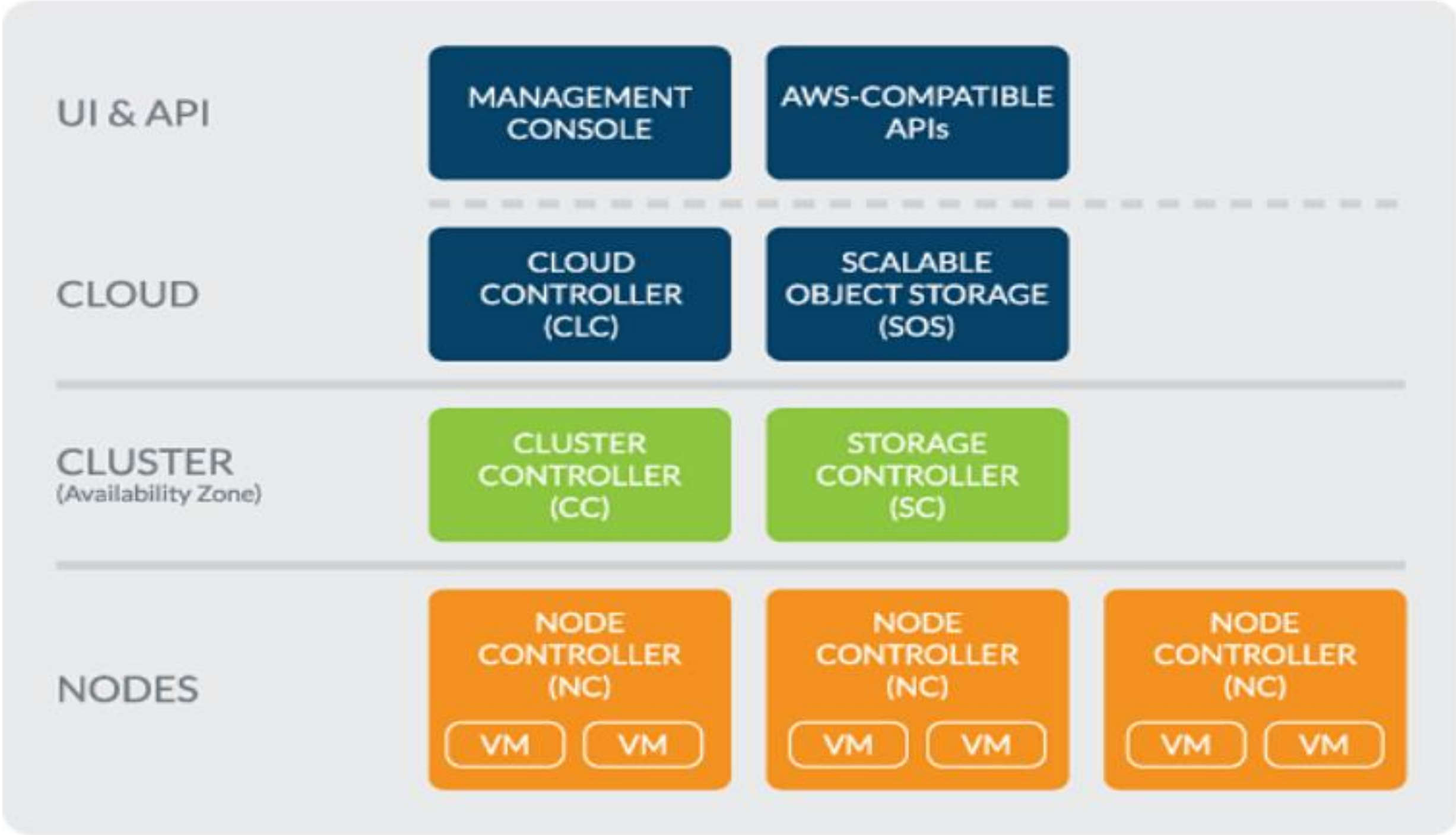
Eucalyptus features include:

- Supports both Linux and Windows virtual machines (VMs).
- Application program interface- (API) compatible with Amazon EC2
- Compatible with Amazon Web Services (AWS) and Simple Storage Service (S3).
- Works with multiple hypervisors including VMware, Xen and KVM.

Eucalyptus features include:

- Can be installed and deployed from source code or DEB and RPM
- Internal processes communications are secured through SOAP and WS-Security.
- Multiple clusters can be virtualized as a single cloud.
- Administrative features such as user and group management and reports.

Open-Source Eucalyptus Cloud Architecture


UI & API
MANAGEMENT CONSOLE
AWS-COMPATIBLE APIs
CLOUD
CLOUD CONTROLLER (CLC)
SCALABLE OBJECT STORAGE (SOS)
CLUSTER
(Availability Zone)
CLUSTER CONTROLLER (CC)
STORAGE CONTROLLER (SC)
NODES
NODE CONTROLLER (NC)
VM
VM
NODE CONTROLLER (NC)
VM
VM
NODE CONTROLLER (NC)
VM
VM

Eucalyptus Cloud Components

- Eucalyptus is made up of five distinct components that can be distributed in various cloud computing architectures.
- The five components are grouped into three separate levels.

Eucalyptus Architecture: Cloud Level

Cloud Controller (CLC)

- The Cloud Controller (CLC) is a Java program that offers EC2-compatible SOAP and Query interfaces, as well as a Web interface to the outside world, for distribution within the cloud architecture.
- In addition to handling incoming requests, the CLC acts as the administrative interface for cloud management and performs high-level resource scheduling and system accounting.
- The CLC accepts user API requests from command-line interfaces like euca2ools or GUI-based tools like the Eucalyptus Management Console and manages the underlying computer storage and network resources.
- Only one CLC can exist per cloud.

Cloud Controller (CLC)

- The CLC handles high-level:
- Authentication
- Accounting
- Reporting
- Quota management

Scalable Object Storage

- Scalable Object Storage (SOS) is the Eucalyptus service equivalent to AWS Simple Storage Service (S3).
- The SOS is pluggable service that allows infrastructure administrators the flexibility to implement scale-out storage on top of commodity resources using open source and commercial solutions that implement the S3 interface.
- Eucalyptus provides a basic storage implementation, known as Walrus, which may suit evaluation and smaller cloud deployments.

Scalable Object Storage

- For large-scale and increased performance, users are encouraged to connect the SOS to dedicated storage solutions such as RiakCS.

Eucalyptus Architecture: Cluster Level

Cluster Level (i.e., Availability Zone)

- Cluster Controller (CC)
- Storage Controller (SC)

Cluster Controller (CC)

- A cluster is equivalent to an AWS availability zone, and a single Eucalyptus cloud can have multiple clusters.
- The Cluster Controller (CC) is written in C and acts as the front end for a cluster within a Eucalyptus cloud and communicates with the Storage Controller (SC) and Node Controller (NC).
- The CC manages instance (i.e., virtual machines) execution and Service Level Agreements (SLAs) per cluster.

Storage Controller (SC)

- The Storage Controller (SC) is written in Java and is the Eucalyptus equivalent to AWS Elastic Block Store (EBS).
- The SC communicates with the Cluster Controller (CC) and Node Controller (NC) within the distributed cloud architecture and manages Eucalyptus block volumes and snapshots to the instances within its specific cluster.
- If an instance requires writing persistent data to memory outside of the cluster, it would need to write to the backend storage, which is available to any instance in any cluster.
- The SC interfaces with storage systems, including local, NFS, iSCSI, and SAN.

Eucalyptus Architecture: Node Level

Node Controller (NC)

- The Node Controller (NC) is part of the Node level of the cloud computing architecture. It is written in C and hosts the virtual machine instances and manages the virtual network endpoints. The NC downloads and caches images from Scalable Object Storage as well as creates and caches instances.

OpenStack Architecture

Introduction to OpenStack

- OpenStack is a cloud OS that is used to control the large pools of computing, storage, and networking resources within a data center.
- OpenStack is an open-source and free software platform. This is essentially used and implemented as an IaaS for cloud computing.
- We can call the OpenStack a software platform that uses pooled virtual resources to create and manage private and public cloud.

Introduction to OpenStack

- OpenStack offers many cloud-related services (such as networking, storage, image services, identity, etc.) by default.
- This can be handled by users through a web-based dashboard, a RESTful API, or command-line tools.
- OpenStack manages a lot of virtual machines; this permits the usage of physical resources to be reduced.

Introduction to OpenStack

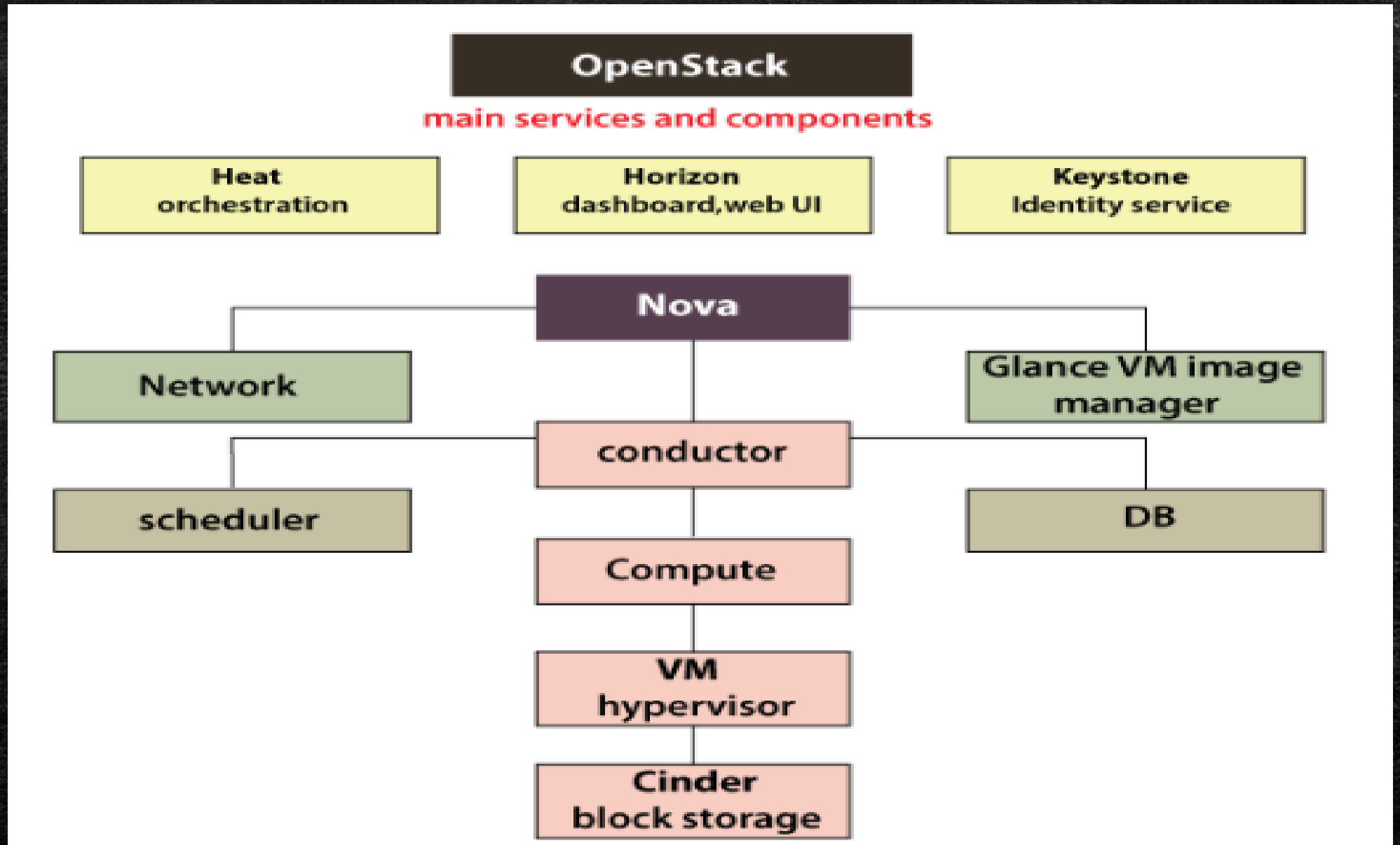
- In 2010, OpenStack began as the joint project of NASA and Rackspace Hosting.
- It was handled by the OpenStack Foundation which is a non-profit collective entity developed in 2012 September for promoting the OpenStack community and software.
- 50+ enterprises have joined this project.

Basic Principles of OpenStack

- Open Source: Under the Apache 2.0 license, OpenStack is coded and published. Apache allows the community to use it for free.
- Open Design: For the forthcoming update, the development group holds a Design Summit every 6 months.
- Open Development: The developers maintain a source code repository that is freely accessible through projects like the Ubuntu Linux distribution via entig100s.
- Open Community: OpenStack allows open and transparent documentation for the community.

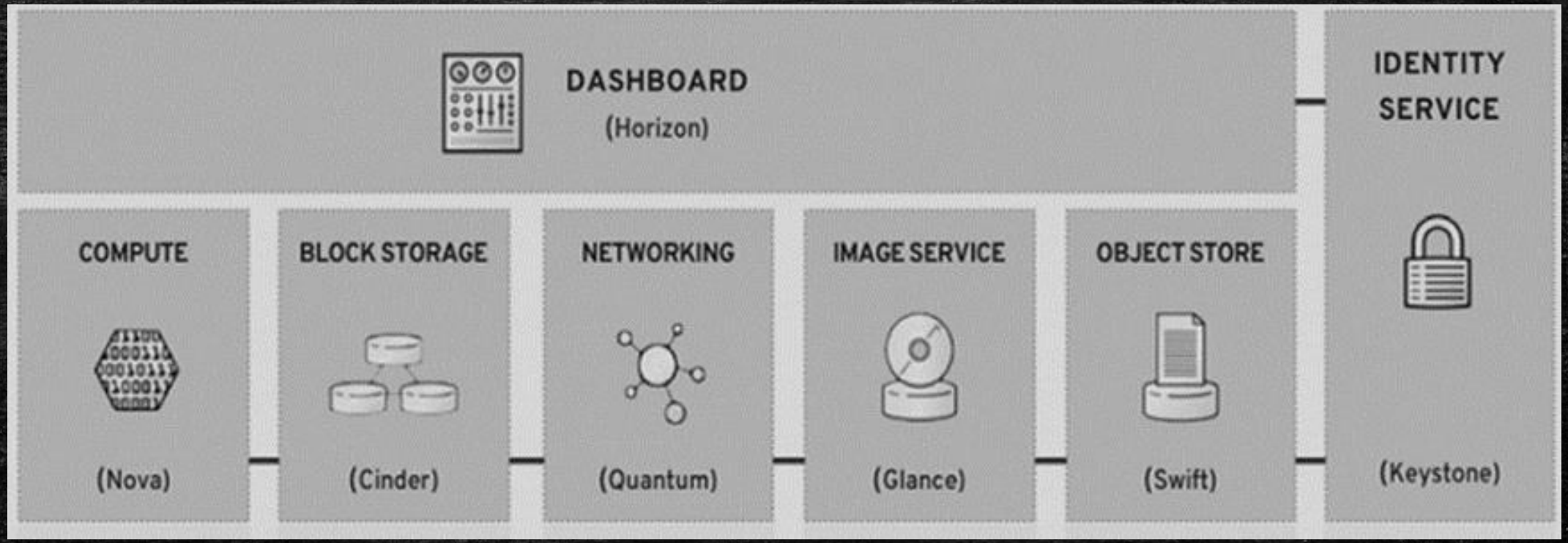
Architecture of OpenStack

- OpenStack contains a modular architecture along with several code names for the components.


OpenStack
main services and components
Heat
orchestration
Horizon
dashboard,web UI
Keystone
Identity service
Nova
Network
Glance VM image
manager
conductor
scheduler
DB
Compute
VM
hypervisor
Cinder
block storage

Components of OpenStack

Components of OpenStack



Compute (Nova)

- Compute is a controller that is used to manage resources in virtualized environments.
- It handles several virtual machines and other instances that perform computing tasks.

Object Storage (Swift)

- To store and retrieve arbitrary data in the cloud, object storage is used.
- In Swift, it is possible to store the files, objects, backups, images, videos, virtual machines, and other unstructured data.
- Developers may use a special identifier for referring the file and objects in place of the path, which directly points to a file and allows the OpenStack to manage where to store the files.

Block Storage (Cinder)

- This works in the traditional way of attaching and detaching an external hard drive to the OS for its local use.
- Cinder manages to add, remove, create new disk space in the server.
- This component provides the virtual storage for the virtual machines in the system.

Networking (Neutron)

- This component is used for networking in OpenStack.
- Neutron manages all the network-related queries, such as IP address management, routers, subnets, firewalls, VPNs, etc.
- It confirms that all the other components are well connected with the OpenStack.

Dashboard (Horizon)

- This is the first component that the user sees in the OpenStack.
- Horizon is the web UI (user interface) component used to access the other back-end services.
- Through individual API (Application programming interface), developers can access the OpenStack's components, but through the dashboard, system administrators can look at what is going on in the cloud and manage it as per their need.

Identity Service (Keystone)

- It is the central repository of all the users and their permissions for the OpenStack services they use.
- This component is used to manage identity services like authorization, authentication, AWS Styles (Amazon Web Services) logins, token-based systems, and checking the other credentials (username & password).

Image Service (Glance)

- The glance component is used to provide the image services to OpenStack. Here, image service means the images or virtual copies of hard disks.
- When we plan to deploy a new virtual machine instance, then glance allows us to use these images as templates.
- Glance allows virtual box (VDI), VMware (VMDK, OVF), Raw, Hyper-V (VHD) and KVM (qcow2) virtual images.

Telementry (Ceilometer)

- It is used to meter the usage and report it to OpenStack's individual users.
- So basically, Telementry provides billing services to OpenStack's individual users.

Orchestration (Heat)

- It allows the developers to store the cloud application's necessities as a file so that all-important resources are available in handy.
- This component organizes many complex applications of the cloud through the templates, via both the local OpenStack REST API and Query API.

Shared File System (Manila)

- It offers storage of the file to a virtual machine.
- This component gives an infrastructure for managing and provisioning file shares.

Elastic Map-reduce (Sahara)

- The Sahara component offers a simple method to the users to preplanned Hadoop clusters by referring to the multiple options such as the Hadoop version, cluster topology and hardware details of nodes and some more.

The release versions of Openstack

Release Name	Release Date	Components Included
Austin	October 2010	Nova, Swift
Bexar	February 2011	Nova, Glance, Swift
Cactus	April 2011	Nova, Glance, Swift
Diablo	September 2011	Nova, Glance, Swift
Essex	April 2012	Nova, Glance, Swift, Horizon, Keystone
Folsom	September 2012	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Grizzly	April 2013	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Havana	October 2013	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder, Ceilometer, Heat
Icehouse	April 2014	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder, Ceilometer, Heat, Trove
Juno	October 2014	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder, Ceilometer, Heat, Trove, Sahara
Kilo	April 2015	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder, Ceilometer, Heat, Trove, Sahara, Ironic

OpenStack command-line interface

- Here is a list of common commands for reference.

Identity (keystone)

- List all users

```
$ openstack user list
```

- List Identity service catalog

```
$ openstack catalog list
```

Images (glance)

- List images you can access

  $ openstack image list

- Delete specified image

  $ openstack image delete IMAGE

- Describe a specific image

  $ openstack image show IMAGE

- Update image

  $ openstack image set IMAGE

Compute (nova)

- List instances, check status of instance

  $ openstack server list

- List images

  $ openstack image list

- Create a flavor named m1.tiny

  $ openstack flavor create --ram 512 --disk 1 --vcpus 1 m1.tiny

- List flavors

  $ openstack flavor list

Cloud management

- Cloud management refers to the exercise of control over public, private or hybrid cloud infrastructure resources and services.
- A well-designed cloud management strategy can help IT pros to control those dynamic and scalable computing environments.

Cloud management goals

- Cloud management can also help organizations achieve three goals:

1. Self-service refers to the flexibility achieved when IT pros access cloud resources, create new ones, monitor usage and cost, and adjust resource allocations.
2. Workflow automation lets operations teams manage cloud instances without human intervention.
3. Cloud analysis helps track cloud workloads and user experiences.

Cloud management Components

Automation and orchestration
- Application migration
- VM images/instances
- Configuration management

Security
- IAM
- Encryption
- Mobile/endpoint security

Governance and compliance
- Risk assessment/threat analysis
- Audits
- Service and resource governance

Performance Monitoring
- Storage
- Networks
- Applications
- Compute

Cost Management
- Cloud instance right sizing
- User chargeback and billing

Cloud Deployment Techniques

- Potential Network Problems and their Mitigation

Network - Node Latency

- Reducing the latency between network nodes is critical to improving cloud performance .
- Using an optimized network fabric for the cloud will serve to minimize transport latency and delays .

Transport - Protocol Latency

- To mitigate the impact of Transmission Control Protocol (TCP) latency , reduce congestion and data loss and improve performance .
- It is best to design and deploy an optimized Ethernet fabric for the cloud .

Number - of - nodes Traversed

- In traditional three - tier architecture (web front - end , application , and database) multiple hops are needed for data to traverse between servers and the end - users .
- Cloud providers must reduce the latency between nodes within a multi - tier topology so that data packets traverse shorter distances .

TCP Congestion

- TCP is normally used for transmission of data packets on the Internet .
- During network congestion or packet transmission errors , TCP uses smaller windows , which negatively impacts throughput rates and reliability .
- The work - around is to design and implement a congestion - free network fabric . This , in turn , enables TCP to use larger windows , thus enabling a higher throughput .

Cloud Network Topologies

- Cloud network topology characterizes the manner in which consumers ' access public or private cloud resources over the Internet or over corporate intranets .
- The cloud network can be viewed as comprising the following three components :

Cloud Network Topologies

- The front - end or user - access layer , which helps users initiate a connection to cloud services .
- The compute layer , which comprises cloud servers , storage , load - balancers , and security devices .
- The underlying network layer , which can be a Layer 2 or Layer 3 network topology.

Automation for Cloud Deployments

- It is imperative for cloud providers to automate cloud operations .
- Automation for cloud is important for all cloud deployment , that is , IaaS , SaaS and PaaS .

Automation for Cloud Deployments

- For IaaS , automation must be used to implement centralized policies for various issues such as security , authorization to access resources , etc.
- It can be used to automatically allocate resources such as bandwidth , memory , storage , etc. based on real - time workload .

Automation for Cloud Deployments

- For PaaS , automation must be used to build a highly flexible platform that provides real - time workload optimization , security , compliance , and metering of utilization during the entire lifecycle of an application

Automation for Cloud Deployments

- For SaaS , automation must be built - in to improve user experience with the applications , improve performance by dynamic resource allocation , and protect from Internet malware .

Federated Cloud Deployments

- A federation is an integration of various smaller units , put together to accomplish a range of tasks .
- A federation of clouds is a mechanism used by a cloud provider , where it rents and integrates resources , applications , or services from various cloud vendors to meet extensive and large - scale customer needs .

Federated Cloud Deployments

- The resources are distributed via many providers and hence the utilization percentage is high , which helps reduce the cost for consumers .
- The performance is better due to caching and having multiple data copies at various provider locations .
- These copies can be accessed by users globally , who benefit from having copies close to them .
- Data availability within a federated cloud is better due to data being replicated to multiple sites .
- Security and compliance in a multi - vendor federated cloud is a primary concern .
- The data is physically present in various datacenters around the world .

Cloud Performance

- Cloud performance refers to the performance of cloud applications and servers , as well as access speeds of network and storage I / O .
- It is measured primarily by the round - trip response time , which is the time interval between a user issued command and the receipt of the result from the cloud Performance (besides service uptime) is an important part of the cloud SLA .
- It can be quantified in terms of the maximum response time experienced by the end user .
- This must be a key metric for the performance of applications and an important SLA criterion .

Cloud Performance Monitoring and Tuning

- There are various issues related to monitoring and tuning cloud performance .
- The performance of virtual machines is difficult to track since the resources are dynamic and based on the work load .
- All cloud aspects are not in the control of a particular organization .
- The division of control depends on the cloud service offered .
- For PaaS , for example , the provider controls the hardware , network , security , servers , operating system , patches , development environment , database configuration , and compilers

Impact of Memory on Cloud Performance

- In cloud computing , memory performance and utilization is fundamental for overall performance.
- Large database transactions require massive amounts of memory to meet the various expected performance levels.
- Moreover , multi - tenancy and simultaneous user tasks put a lot of demand on memory.
- The coordination between different cloud services to meet a particular demand requires in memory tasks.

Impact of Memory on Cloud Performance

- Jobs need to be split and assembled after processing , which increases overhead cost.
- Another problem in cloud relates to memory leaks.
- It is a situation where a user job , database , or application does not return back the temporarily - allocated memory to the operating system even after it has been cleaned up and is no longer in use.
- This can be due to a bug , malware , or a deliberate user job that wants to consume all memory.

Improving Cloud Database Performance

- Cloud databases offer noteworthy benefits over traditionally - hosted internal databases.
- Moreover , cloud vendors continue to add and improve their database offerings to make it a convincing option for enterprises.
- Cloud databases have higher ease of accessibility , better replication to remote datacenters alongside automation and better elasticity.
- Sharding a cloud database is another technique to improve performance.

Improving Cloud Database Performance

- Sharding is a process of splitting a large database into a number of smaller databases , each being hosted on a separate server.
- It helps to boost the performance of applications that require frequent and large database transactions.
- Sharding also helps reduce the size of the database index , thus decreasing the time needed for searches within the database.
- To further improve performance and availability , providers offer a horizontally - scaled server environment , where it is quick and easy to bring up more virtual machines to meet higher workloads .

Improving Cloud Database Performance

- Besides performance , providers focus upon improving database integrity by using database profilers.
- Sharding analyzes the source database for inconsistencies in index , table relationships , or data structure.
- By examining the data quality and utilization pattern , it is able to point out the potential problems , if any , within a database.
- This improves the performance of the database.

Cloud Services Brokerage (CSB)

- A Cloud Services Brokerage (CSB) is an organization that plays a role as a facilitator or inter - mediator for delivering cloud services.
- The CSB is usually a telecommunication or datacenter hosting service provider with a large number of customers.
- In the cloud provider - consumer relationship , CSBs are an optional entity involved with mediating between the two.
- However in the process , the CSBs also offer additional value to the provider and consumer.

Cloud Services Brokerage (CSB)

- They help providers by relieving them of acquiring customers , billing , and enabling integrated access to multiple cloud services.
- The cloud consumers get integrated access to one or more cloud and value - added services such as cloud backups , SaaS , and Identity Management (IdM).

Google Classroom code for "Cloud Computing"

- Join the Google Classroom by Using following Code:

2t6xpa5

Thank You!!!Any Query?

asktoshivsir@gmail.com

Shivkumar Chandey

(+91 9987389441)

Scan QR Code to connect on LinkedIn