UNIT IV

CLOUD DEPLOYMENY ENVIRONMENT

Google App Engine – Amazon AWS – Microsoft Azure; Cloud Software Environments – Eucalyptus – OpenStack.

GOOGLE APPLICATION ENGINE (GAE)

- Google App Engine is a PaaS cloud that provides a complete Web service environment (Platform)
- GAE provides Web application development platform for users.
- All required hardware, operating systems and software are provided to clients.
- Clients can develop their own applications, while App Engine runs the applications on
- GAE helps to easily develop an Web Application
- App Engine only supports the Java and Python programming languages.
- The Google App Engine (GAE) provides a powerful distributed data storage service.

GOOGLE CLOUD INFRASTRUCTURE

- Google has established cloud development by making use of large number of data centers.
- Eg: Google established cloud services in

❖ Gmail

❖ Google Docs

❖ Google Earth etc.

- These applications can support a large number of users simultaneously with High Availability (HA).
- In 2008, Google announced the GAE web application platform.
- GAE enables users to run their applications on a large number of data centers.
- Google App Engine environment includes the following features :

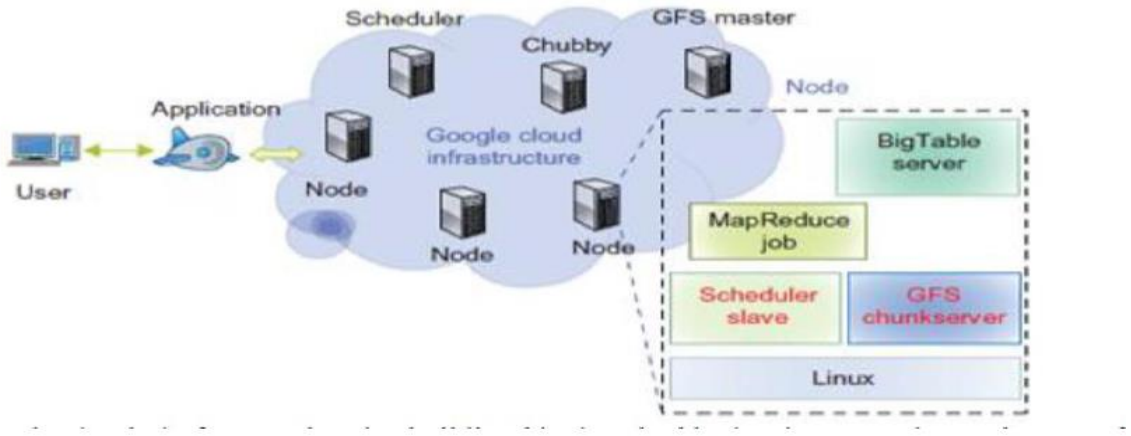
❖ Dynamic web serving

❖ Persistent(constant) storage with queries, sorting, and transactions

❖ Automatic scaling and load balancing

- Provides Application Programming Interface(API) for authenticating users.
- Send email using Google Accounts.
- Local development environment that simulates(create) Google App Engine on your computer.

GAE ARCHITECTURE



TECHNOLOGIES USED BY GOOGLE ARE

- Google File System(GFS) ->for storing large amounts of data.
- MapReduce->for application program development.
- Chubby-> for distributed application lock services.
- BigTable-> offers a storage service.
- Third-party application providers can use GAE to build cloud applications for providing

services.

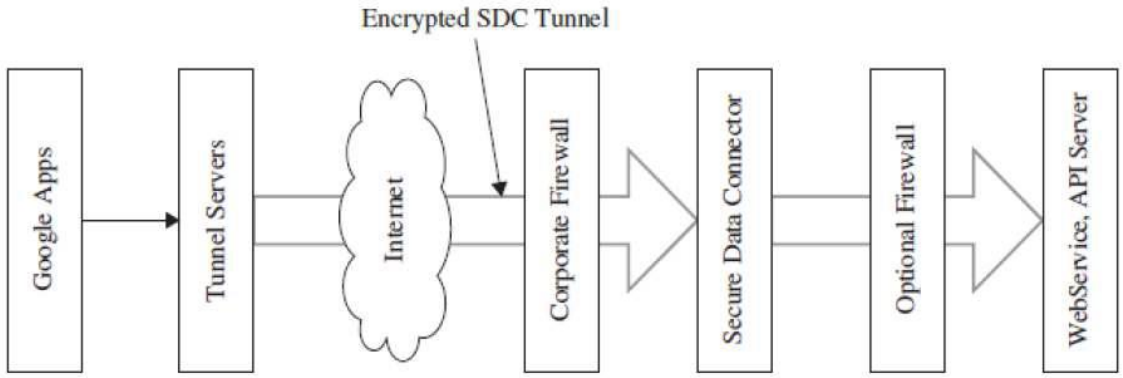
- Inside each data center, there are thousands of servers forming different clusters.
- GAE runs the user program on Google's infrastructure.
- Application developers now do not need to worry about the maintenance of servers.
- GAE can be thought of as the combination of several software components.
- GAE supports Python and Java programming environments.

FUNCTIONAL MODULES OF GAE

- The GAE platform comprises the following five major components.
- DataStore: offers data storage services based on BigTable techniques.
- The Google App Engine (GAE) provides a powerful distributed data storage service.

· This provides a secure data Storage.

GOOGLE SECURE DATA CONNECTOR (SDC)



FUNCTIONAL MODULES OF GAE

· When the user wants to get the data, he/she will first send an authorized data requests to Google Apps.

· It forwards the request to the tunnel server.

· The tunnel servers validate the request identity.

· If the identity is valid, the tunnel protocol allows the SDC to set up a connection, authenticate, and encrypt the data that flows across the Internet.

· SDC also validates whether a user is authorized to access a specified resource.

· Application runtime environment offers a platform for web programming and execution.

· It supports two development languages: Python and Java.

· Software Development Kit (SDK) is used for local application development.

· The SDK allows users to execute test runs of local applications and upload application code.

· Administration console is used for easy management of user application development
cycles.

· GAE web service infrastructure provides special guarantee flexible use and management of
storage and network resources by GAE.

· Google offers essentially free GAE services to all Gmail account owners.

· We can register for a GAE account or use your Gmail account name to sign up for the

service.

· The service is free within a quota.

· If you exceed the quota, extra amount will be charged.

· Allows the user to deploy user-built applications on top of the cloud infrastructure.

· They are built using the programming languages and software tools supported by the

provider (e.g., Java, Python)

GAE APPLICATIONS

Well-known GAE applications

· Google Search Engine

· Google Docs

· Google Earth

· Gmail

· These applications can support large numbers of users simultaneously.

· Users can interact with Google applications via the web interface provided by each application.

· Applications run in the Google data centers.

· Inside each data center, there might be thousands of server nodes to form different clusters.

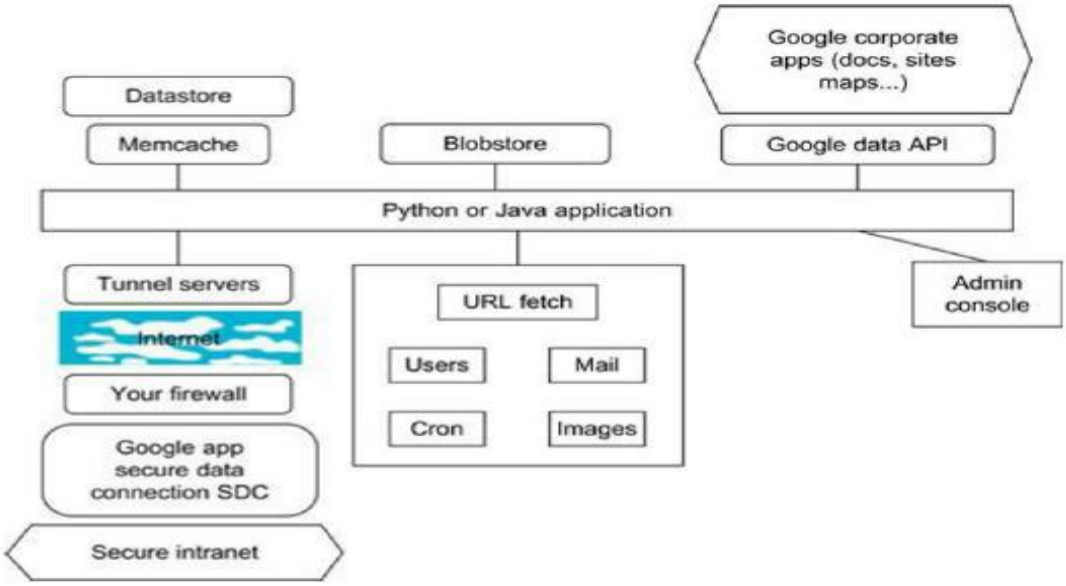
· Each cluster can run multipurpose servers.

5. 3.1 Programming Support of Google App Engine

GAE programming model for two supported languages: Java and Python. A client environment includes an Eclipse plug-in for Java allows you to debug your GAE on your local

machine. Google Web Toolkit is available for Java web application developers. Python is used

with frameworks such as Django and CherryPy, but Google also has webapp Python environment.



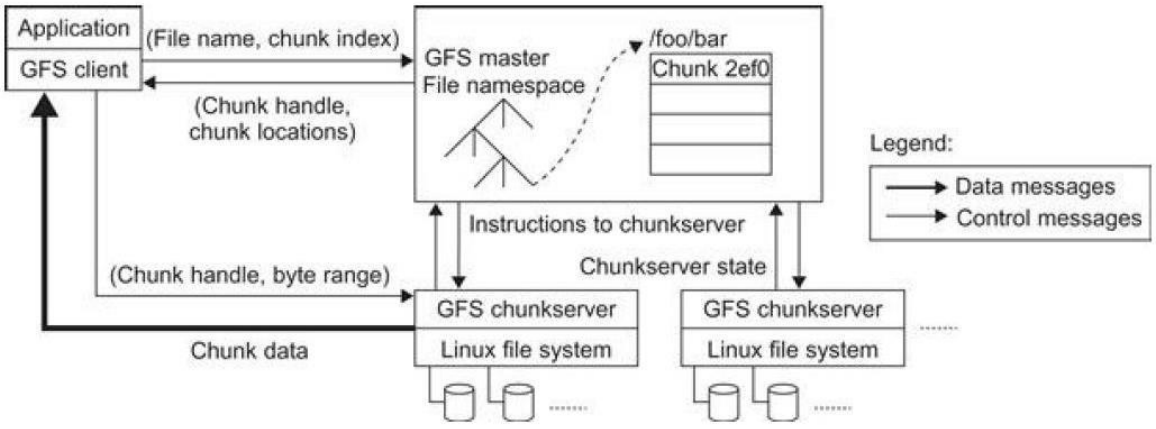
There are several powerful constructs for storing and accessing data. The data store is a NOSQL data management system for entities. Java offers Java Data Object (JDO) and Java Persistence API (JPA) interfaces implemented by the Data Nucleus Access platform, while Python has a SQL-like query language called GQL. The performance of the data store can be enhanced by in-memory caching using the memcache, which can also be used independently of the data store.

Recently, Google added the blobstore which is suitable for large files as its size limit is 2 GB. There are several mechanisms for incorporating external resources. The Google SDC Secure Data Connection can tunnel through the Internet and link your intranet to an external GAE application. The URL Fetch operation provides the ability for applications to fetch resources and communicate with other hosts over the Internet using HTTP and HTTPS requests.

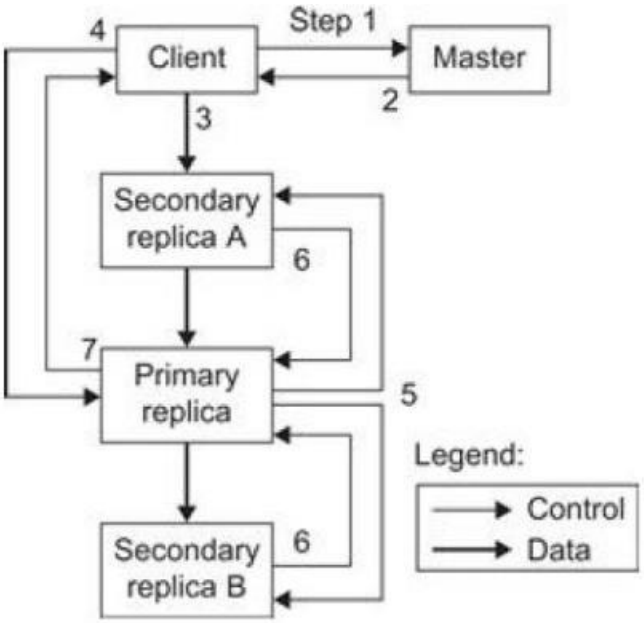
An application can use Google Accounts for user authentication. Google Accounts handles user account creation and sign-in, and a user that already has a Google account (such as a Gmail account) can use that account with your app. GAE provides the ability to manipulate image data using a dedicated Images service which can resize, rotate, flip, crop, and enhance images. A GAE application is configured to consume resources up to certain limits or quotas. With quotas, GAE ensures that your application won't exceed your budget, and that other applications running on GAE won't impact the performance of your app. In particular, GAE use is free up to certain quotas.

Google File System (GFS)

GFS is a fundamental storage service for Google's search engine. GFS was designed for Google applications, and Google applications were built for GFS. There are several concerns in GFS. rate). As servers are composed of inexpensive commodity components, it is the norm rather than the exception that concurrent failures will occur all the time. Other concerns the file size in GFS. GFS typically will hold a large number of huge files, each 100 MB or larger, with files that are multiple GB in size quite common. Thus, Google has chosen its file data block size to be 64 MB instead of the 4 KB in typical traditional file systems. The I/O pattern in the Google application is also special. Files are typically written once, and the write operations are often the appending data blocks to the end of files. Multiple appending operations might be concurrent. The customized API can simplify the problem and focus on Google applications. Figure shows the GFS architecture. It is quite obvious that there is a single master in the whole cluster. Other nodes act as the chunk servers for storing data, while the single master stores the metadata. The file system namespace and locking facilities are managed by the master. The master periodically communicates with the chunk servers to collect management information as well as give instructions to the chunk servers to do work such as load balancing or fail recovery.



The master has enough information to keep the whole cluster in a healthy state. Google uses a shadow master to replicate all the data on the master, and the design guarantees that all the data operations are performed directly between the client and the chunk server. The control messages are transferred between the master and the clients and they can be cached for future use. With the current quality of commodity servers, the single master can handle a cluster of more than 1,000 nodes.



The mutation takes the following steps:

1. The client asks the master which chunk server holds the current lease for the chunk and the locations of the other replicas. If no one has a lease, the master grants one to a replica it chooses (not shown).

2. The master replies with the identity of the primary and the locations of the other (secondary) replicas. The client caches this data for future mutations. It needs to contact the master again only when the primary becomes unreachable or replies that it no longer holds a lease.

3. The client pushes the data to all the replicas. Each chunk server will store the data in an internal LRU buffer cache until the data is used or aged out. By decoupling the data flow from the control flow, we can improve performance by scheduling the expensive data flow based on the network topology regardless of which chunk server is the primary.

4. Once all the replicas have acknowledged receiving the data, the client sends a write request to the primary. The request identifies the data pushed earlier to all the replicas. The primary assigns consecutive serial numbers to all the mutations it receives, possibly from multiple clients, which provides the necessary serialization. It applies the mutation to its own local state in serial order.

5. The primary forwards the write request to all secondary replicas. Each secondary replica applies mutations in the same serial number order assigned by the primary.

6. The secondaries all reply to the primary indicating that they have completed the operation.
7. The primary replies to the client. Any errors encountered at any replicas are reported to the client. In case of errors, the write corrects at the primary and an arbitrary subset of the secondary replicas. The client request is considered to have failed, and the modified region is left in an inconsistent state. Our client code handles such errors by retrying the failed mutation.

AMAZON WEB SERVICES

AWS tutorial provides basic and advanced concepts. Our AWS tutorial is designed for beginners and professionals.

AWS stands for Amazon Web Services which uses distributed IT infrastructure to provide different IT resources on demand.

Our AWS tutorial includes all the topics such as introduction, history of aws, global infrastructure, features of aws, IAM, Storage services, Database services, etc.

What is AWS?

- AWS stands for **Amazon Web Services**.
- The AWS service is provided by the Amazon that uses distributed IT infrastructure to provide different IT resources available on demand. It provides different services such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS).

- Amazon launched AWS, a cloud computing platform to allow the different organizations to take advantage of reliable IT infrastructure.

Uses of AWS

- A small manufacturing organization uses their expertise to expand their business by leaving their IT management to the AWS.
- A large enterprise spread across the globe can utilize the AWS to deliver the training to the distributed workforce.
- An architecture consulting company can use AWS to get the high-compute rendering of construction prototype.
- A media company can use the AWS to provide different types of content such as ebox or audio files to the worldwide files.

Pay-As-You-Go

Based on the concept of Pay-As-You-Go, AWS provides the services to the customers AWS provides services to customers when required without any prior commitment or upfront investment. Pay-As-You-Go enables the customers to procure services from AWS.

- Computing
- Programming models
- Database storage
- Networking

Advantages of AWS

1) Flexibility

- We can get more time for core business tasks due to the instant availability of new features and services in AWS.
- It provides effortless hosting of legacy applications. AWS does not require learning new technologies and migration of applications to the AWS provides the advanced computing and efficient storage.
- AWS also offers a choice that whether we want to run the applications and services together or not. We can also choose to run a part of the IT infrastructure in AWS and the remaining part in data centres.

2) Cost-effectiveness

AWS requires no upfront investment, long-term commitment, and minimum expense when compared to traditional IT infrastructure that requires a huge investment.

3) Scalability/Elasticity

Through AWS, autoscaling and elastic load balancing techniques are automatically scaled up or down, when demand increases or decreases respectively. AWS techniques are ideal for handling unpredictable or very high loads. Due to this reason, organizations enjoy the benefits of reduced cost and increased user satisfaction.

4) Security

- AWS provides end-to-end security and privacy to customers.
- AWS has a virtual infrastructure that offers optimum availability while managing full privacy and isolation of their operations.
- Customers can expect high-level of physical security because of Amazon's several years of experience in designing, developing and maintaining large-scale IT operation centers.
- AWS ensures the three aspects of security, i.e., Confidentiality, integrity, and availability of user's data.

MICROSOFT AZURE

Microsoft Azure is a cloud computing platform that provides a wide variety of services that we can use without purchasing and arranging our hardware. It enables the fast development of solutions and provides the resources to complete tasks that may not be achievable in an on-premises environment. Azure Services like compute, storage, network, and application services allow us to put our effort into building great solutions without worrying about the assembly of physical infrastructure.

This tutorial covers the fundamentals of Azure, which will provide us the idea about all the Azure key services that we are most likely required to know to start developing solutions. After completing this tutorial, we can crack job interviews or able to get different Microsoft Azure certifications.

What is Azure

Microsoft Azure is a growing set of cloud computing services created by Microsoft that hosts your existing applications, streamline the development of a new application, and also enhances our on-premises applications. It helps the organizations in building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

Azure Services

Compute services: It includes the Microsoft Azure Cloud Services, Azure Virtual Machines, Azure Website, and Azure Mobile Services, which processes the data on the cloud with the help of powerful processors.

- **Data services:** This service is used to store data over the cloud that can be scaled according to the requirements. It includes Microsoft Azure Storage (Blob, Queue Table, and Azure File services), Azure SQL Database, and the Redis Cache.
- **Application services:** It includes services, which help us to build and operate our application, like the Azure Active Directory, Service Bus for connecting distributed systems, HDInsight for processing big data, the Azure Scheduler, and the Azure Media Services.
- **Network services:** It helps you to connect with the cloud and on-premises infrastructure, which includes Virtual Networks, Azure Content Delivery Network, and the Azure Traffic Manager.

Microsoft Azure Services

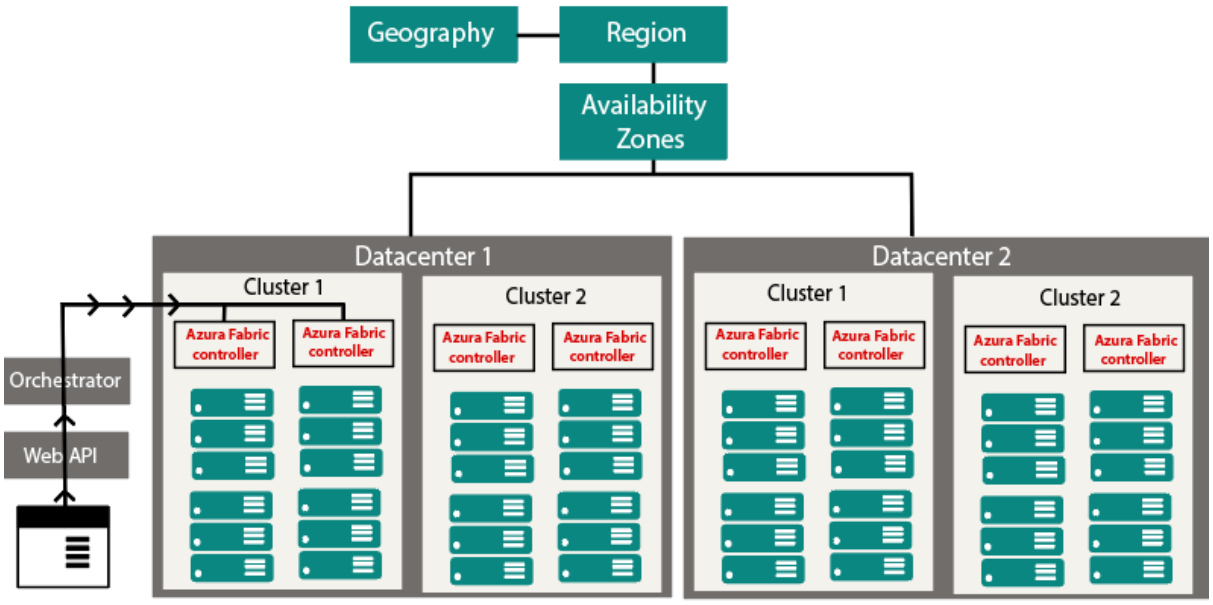
Following are some of the services Microsoft Azure offers:

- **Compute:** Includes Virtual Machines, Virtual Machine Scale Sets, Functions for serverless computing, Batch for containerized batch workloads, Service Fabric for microservices and container orchestration, and Cloud Services for building cloud-based apps and APIs.
- **Networking:** With Azure, you can use a variety of networking tools, like the Virtual Network, which can connect to on-premise data centers; Load Balancer; Application Gateway; VPN Gateway; Azure DNS for domain hosting, Content Delivery Network, Traffic Manager, ExpressRoute dedicated private network fiber connections; and Network Watcher monitoring and diagnostics
- **Storage:** Includes Blob, Queue, File, and Disk Storage, as well as a Data Lake Store, Backup, and Site Recovery, among others.
- **Web + Mobile:** Creating Web + Mobile applications is very easy as it includes several services for building and deploying applications.
- **Containers:** Azure has a property that includes Container Service, which supports Kubernetes, DC/OS or Docker Swarm, and Container Registry, as well as tools for microservices.

- **Databases:** Azure also included several SQL-based databases and related tools.
- **Data + Analytics:** Azure has some big data tools like HDInsight for Hadoop Spark, R Server, HBase, and Storm clusters
- **AI + Cognitive Services:** With Azure developing applications with artificial intelligence capabilities, like the Computer Vision API, Face API, Bing Web Search, Video Indexer, and Language Understanding Intelligent.
- **Internet of Things:** Includes IoT Hub and IoT Edge services that can be combined with a variety of machine learning, analytics, and communications services.
- **Security + Identity:** Includes Security Center, Azure Active Directory, Key Vault, and Multi-Factor Authentication Services.
- **Developer Tools:** Includes cloud development services like Visual Studio Team Services, Azure DevTest Labs, HockeyApp mobile app deployment and monitoring, Xamarin cross-platform mobile development, and more.

How Azure works

It is essential to understand the internal workings of Azure so that we can design our applications on Azure effectively with high availability, data residency, resilience, etc.



Microsoft Azure is completely based on the concept of virtualization. So, similar to other virtualized data center, it also contains *racks*. Each rack has a separate power unit and network switch, and also each rack is integrated with a software called *Fabric-Controller*. This *Fabric-controller* is a distributed application, which is responsible for managing and monitoring servers within the rack. In case of any server failure, the Fabric-controller recognizes it and recovers it. And Each of these Fabric-Controller is, in turn, connected to a piece of software called *Orchestrator*. This *Orchestrator* includes web-services, Rest API to create, update, and delete resources.

When a request is made by the user either using PowerShell or Azure portal. First, it will go to the Orchestrator, where it will fundamentally do three things:

1. Authenticate the User
2. It will Authorize the user, i.e., it will check whether the user is allowed to do the requested task.
3. It will look into the database for the availability of space based on the resources and pass the request to an appropriate Azure Fabric controller to execute the request.

Combinations of racks form a cluster. We have multiple clusters within a data center, and we can have multiple Data Centers within an Availability zone, multiple Availability zones within a Region, and multiple Regions within a Geography.

- **Geographies:** It is a discrete market, typically contains two or more regions, that preserves data residency and compliance boundaries.
- **Azure regions:** A region is a collection of data centers deployed within a defined perimeter and interconnected through a dedicated regional low-latency network.

Azure covers more global regions than any other cloud provider, which offers the scalability needed to bring applications and users closer around the world. It is globally available in 50 regions around the world. Due to its availability over many regions, it

helps in preserving data residency and offers comprehensive compliance and flexible options to the customers.

- **Availability Zones:** These are the physically separated location within an Azure region. Each one of them is made up of one or more data centers, independent configuration.

Azure Pricing

It is one of the main reasons to learn Microsoft Azure. Because Microsoft is providing free Credits in the Azure account to access Azure services for free for a short duration. This credit is sufficient for people who are new at Microsoft Azure and want to use the services.

Microsoft offers the **pay-as-you-go** approach that helps organizations to serve their needs. Typically the cloud services will be charged based on the usage. The flexible pricing option helps in up-scaling and down-scaling the architecture as per our requirements**.**

Azure Certification

Microsoft Azure helps to fill the gap between the industry requirement and the resource available. Microsoft provides Azure Certification into three major categories, which are:

- **Azure Administrator:** Those who implement, monitor, and maintain Microsoft Azure solutions, including major services.

- **Azure Developer:** Those who design, build, test, and maintain cloud solutions, such as applications and services, partnering with cloud solution architects, cloud DBAs, cloud administrators, and clients to implement these solutions.

- **Azure Solution Architect:** Those who have expertise in compute, network, storage, and security so that they can design the solutions that run on Azure.

All these certifications are divided into different levels. If anyone is planning to get certified, then he/she first has to get an associate-level certification and then go for the advanced level.

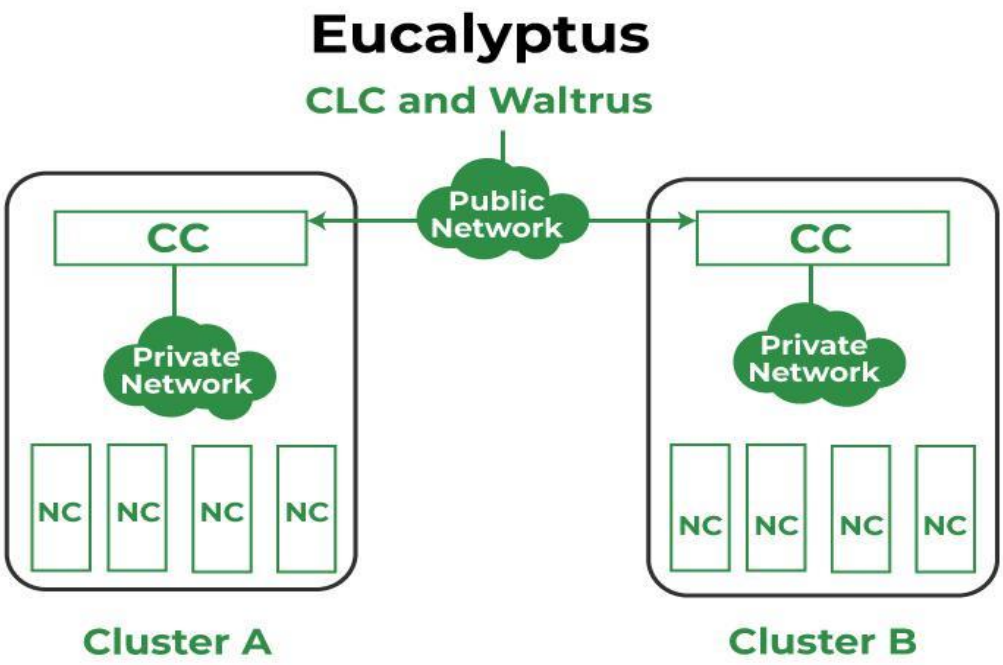
	AWS	Google Cloud	Azure
Technology	EC2 (Elastic Compute Cloud)	Google Compute Engine (GCE)	VHD (Virtual Hard Disk)
Databases Supported	AWS fully supports relational and NoSQL databases and Big Data.	Technologies pioneered by Google, like Big Query, Big Table, and Hadoop, are naturally fully supported.	Azure supports both relational and NoSQL databases, and Big Data, through Windows Azure Table and HDInsight.
Pricing	Per hour – rounded up	Per minute – rounded up (minimum 10 minutes)	Per minute – rounded up commitments (pre-paid or monthly)
Models	On demand, reserved, spot	On demand – sustained use	On demand – short term commitments (pre-paid or monthly)
Difficulties	Many enterprises find it difficult to understand the company's cost structure	Fewer features and services.	Less "enterprise-ready"
Storage Services	• Simple Storage Service (S3) • Elastic Block Storage (EBS) • Elastic Block Storage (EBS)	• Blob Storage • Queue Storage • File Storage • Disk Storage • Data Lake Store	• Cloud Storage • Persistent Disk • Transfer Appliance
Machine Learning	• Sage Maker • Lex • Polly • And many more	• Machine Learning • Azure Bot Service • Cognitive Service	• Cloud Speech API • Cloud Video Intelligence • Cloud Machine Learning Engine • And many more.

CLOUD SOFTWARE ENVIRONMENTS:

Introduction to Eucalyptus

The open-source cloud refers to software or applications publicly available for the users in the cloud to set up for their own purpose or for their organization.

Eucalyptus is a Linux-based open-source software architecture for cloud computing and also a storage platform that implements Infrastructure a Service (IaaS). It provides quick and efficient computing services. Eucalyptus was designed to provide services compatible with Amazon's EC2 cloud and Simple Storage Service(S3).



Eucalyptus Architecture

Eucalyptus CLIs can handle Amazon Web Services and their own private instances. Clients have the independence to transfer cases from Eucalyptus to Amazon Elastic Cloud. The virtualization layer oversees the Network, storage, and Computing. Occurrences are isolated by hardware virtualization.

Important Features are:-

1. **Images**: A good example is the Eucalyptus Machine Image which is a module software bundled and uploaded to the Cloud.
2. **Instances**: When we run the picture and utilize it, it turns into an instance.
3. **Networking**: It can be further subdivided into three modes: Static mode(allocates IP address to instances), System mode (assigns a MAC address and imputes the instance's network interface to the physical network via NC), and Managed mode (achieves local network of instances).
4. **Access Control:** It is utilized to give limitations to clients.

5. **Elastic Block Storage**: It gives block-level storage volumes to connect to an instance.
6. **Auto-scaling and Load Adjusting**: It is utilized to make or obliterate cases or administrations dependent on necessities.

Components of Architecture

- **Node Controller** is the lifecycle of instances running on each node. Interacts with the operating system, hypervisor, and Cluster Controller. It controls the working of VM instances on the host machine.
- **Cluster Controller** manages one or more Node Controller and Cloud Controller simultaneously. It gathers information and schedules VM execution.
- **Storage Controller (Walrus)** Allows the creation of snapshots of volumes. Persistent block storage over VM instances. Walrus Storage Controller is a simple file storage system. It stores images and snapshots. Stores and serves files using S3(Simple Storage Service) APIs.
- **Cloud Controller** Front-end for the entire architecture. It acts as a Complaint Web Services to client tools on one side and interacts with the rest of the components on the other side.

Operation Modes Of Eucalyptus

- **Managed Mode**: Numerous security groups to users as the network is large. Each security group is assigned a set or a subset of IP addresses. Ingress rules are applied through the security groups specified by the user. The network is isolated by VLAN between Cluster Controller and Node Controller. Assigns two IP addresses on each virtual machine.
- **Managed (No VLAN) Node:** The root user on the virtual machine can snoop into other virtual machines running on the same network layer. It does not provide VM network isolation.

- **System Mode:** Simplest of all modes, least number of features. A MAC address is assigned to a virtual machine instance and attached to Node Controller's bridge Ethernet device.
- **Static Mode**: Similar to system mode but has more control over the assignment of IP address. MAC address/IP address pair is mapped to static entry within the DHCP server. The next set of MAC/IP addresses is mapped.

Advantages Of The Eucalyptus Cloud

1. Eucalyptus can be utilized to benefit both the eucalyptus private cloud and the eucalyptus public cloud.
2. Examples of Amazon or Eucalyptus machine pictures can be run on both clouds.
3. Its API is completely similar to all the Amazon Web Services.
4. Eucalyptus can be utilized with DevOps apparatuses like Chef and Puppet.
5. Although it isn't as popular yet but has the potential to be an alternative to OpenStack and CloudStack.
6. It is used to gather hybrid, public and private clouds.
7. It allows users to deliver their own data centers into a private cloud and hence, extend the services to other organizations.

Introduction to OpenStack

It is a free open standard cloud computing platform that first came into existence on July 21' 2010. It was a joint project of Rackspace Hosting and NASA to make cloud computing more ubiquitous in nature. It is deployed as Infrastructure-as-a-service(IaaS) in both public and private clouds where virtual resources are made available to the users. The software platform contains interrelated components that control multi-vendor hardware pools of processing, storage, networking resources through a data center. In OpenStack, the tools which are used to build this platform are referred to as "projects". These projects handle a large number of services including computing, networking, and storage services. Unlike virtualization, in which resources such as RAM, CPU, etc are abstracted from the hardware using hypervisors,

OpenStack uses a number of APIs to abstract those resources so that users and the administrators are able to directly interact with the cloud services.

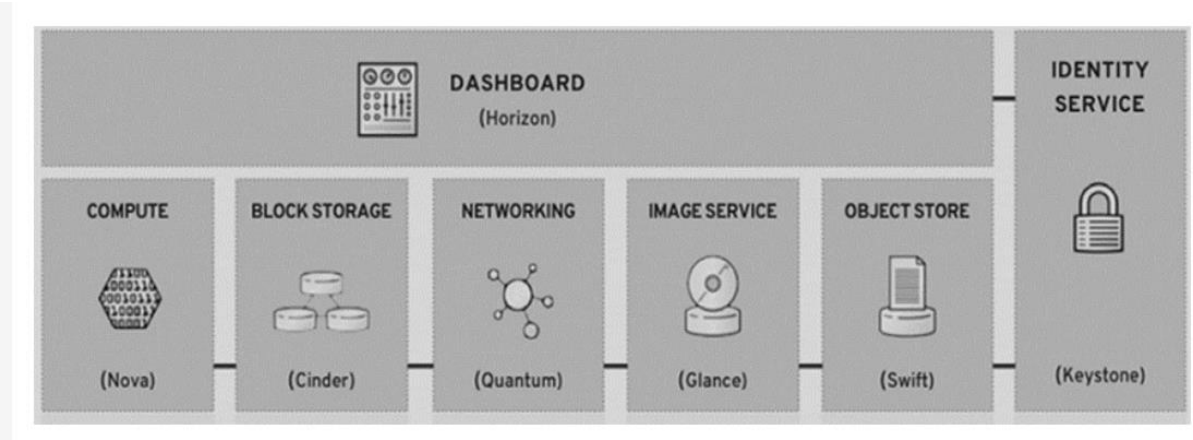
OpenStack components

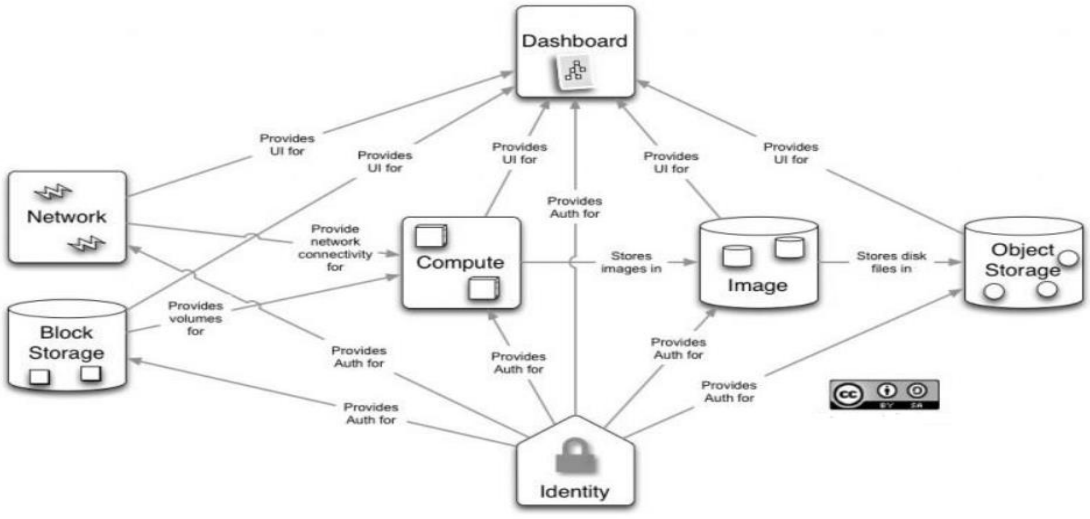
Apart from various projects which constitute the OpenStack platform, there are nine major services namely Nova, Neutron, Swift, Cinder, Keystone, Horizon, Ceilometer, and Heat. Here is the basic definition of all the components which will give us a basic idea about these components.

1. **Nova (compute service):** It manages the compute resources like creating, deleting, and handling the scheduling. It can be seen as a program dedicated to the automation of resources that are responsible for the virtualization of services and high-performance computing.
2. **Neutron (networking service):** It is responsible for connecting all the networks across OpenStack. It is an API driven service that manages all networks and IP addresses.
3. **Swift (object storage):** It is an object storage service with high fault tolerance capabilities and it used to retrieve unstructured data objects with the help of Restful API. Being a distributed platform, it is also used to provide redundant storage within servers that are clustered together. It is able to successfully manage petabytes of data.
4. **Cinder (block storage):** It is responsible for providing persistent block storage that is made accessible using an API (self- service). Consequently, it allows users to define and manage the amount of cloud storage required.
5. **Keystone (identity service provider):** It is responsible for all types of authentications and authorizations in the OpenStack services. It is a directory-based service that uses a central repository to map the correct services with the correct user.
6. **Glance (image service provider):** It is responsible for registering, storing, and retrieving virtual disk images from the complete network. These images are stored in a wide range of back-end systems.

7. **Horizon (dashboard):** It is responsible for providing a web-based interface for OpenStack services. It is used to manage, provision, and monitor cloud resources.
8. **Ceilometer (telemetry):** It is responsible for metering and billing of services used. Also, it is used to generate alarms when a certain threshold is exceeded.
9. **Heat (orchestration):** It is used for on-demand service provisioning with auto-scaling of cloud resources. It works in coordination with the ceilometer.

These are the services around which this platform revolves around. These services individually handle storage, compute, networking, identity, etc. These services are the base on which the rest of the projects rely on and are able to orchestrate services, allow bare-metal provisioning, handle dashboards, etc.





Features of OpenStack

- **Modular architecture:** OpenStack is designed with a modular architecture that enables users to deploy only the components they need. This makes it easier to customize and scale the platform to meet specific business requirements.
- **Multi-tenancy support:** OpenStack provides multi-tenancy support, which enables multiple users to access the same cloud infrastructure while maintaining security and isolation between them. This is particularly important for cloud service providers who need to offer services to multiple customers.
- **Open-source software:** OpenStack is an open-source software platform that is free to use and modify. This enables users to customize the platform to meet their specific requirements, without the need for expensive proprietary software licenses.
- **Distributed architecture:** OpenStack is designed with a distributed architecture that enables users to scale their cloud infrastructure horizontally across multiple physical servers. This makes it easier to handle large workloads and improve system performance.
- **API-driven:** OpenStack is API-driven, which means that all components can be accessed and controlled through a set of APIs. This makes it easier to automate and integrate with other tools and services.
- **Comprehensive dashboard:** OpenStack provides a comprehensive dashboard that enables users to manage their cloud infrastructure and resources through a user-friendly web interface. This makes it easier to monitor and manage cloud resources without the need for specialized technical skills.
- **Resource pooling:** OpenStack enables users to pool computing, storage, and networking resources, which can be dynamically allocated and de-allocated based on demand. This enables users to optimize resource utilization and reduce waste.

Advantages of using OpenStack

- It boosts rapid provisioning of resources due to which orchestration and scaling up and down of resources becomes easy.
- Deployment of applications using OpenStack does not consume a large amount of time.
- Since resources are scalable therefore they are used more wisely and efficiently.
- The regulatory compliances associated with its usage are manageable.

Disadvantages of using OpenStack

- OpenStack is not very robust when orchestration is considered.
- Even today, the APIs provided and supported by OpenStack are not compatible with many of the hybrid cloud providers, thus integrating solutions becomes difficult.
- Like all cloud service providers OpenStack services also come with the risk of security breaches.