What is Open Virtualization Format (OVF)?

Open Virtualization Format (OVF) is an open source standard for packaging and distributing software applications and services for virtual machines (VMs).

As the adoption of virtual infrastructure increases, there is a greater need for an open, standard, portable and platform-independent metadata format to distribute virtual systems onto and between virtualization platforms. OVF provides such a packaging and distribution format to facilitate the mobility of VMs.

The standard also describes multiple VMs with their relationships. These VMs can be wrapped up in a single virtual appliance file to enable broader distribution.

Open Virtualization Format explained

OVF is not a specification describing a virtual disk. Rather, it is a standard representation of VM metadata. This VM metadata includes the following:

- name
- configured memory
- CPU
- storage settings
- network

In addition to describing the above attributes of virtual hardware, OVF also allows virtual appliance vendors to add comments about the VM and other characteristics, such as an end-user license agreement (EULA), boot parameters and minimum requirements. They can also encrypt, compress and digitally sign their content.

OVF, which is specified by the Distributed Management Task Force (DMTF) and published by the International Organization for Standardization (ISO) as ISO 1720, is independent of any particular processor or hypervisor architecture. It leverages DMTF's Common Information Model (CIM) to allow management software to understand and map resource properties by using the OVF open standard.

As a packaging format for virtual appliances, OVF enables the mobility of virtual machines across multiple platforms by facilitating the distribution of enterprise software in a flexible, secure and efficient manner.

Consequently, both vendors and users can follow OVF specifications to deploy a VM on any virtualization platform. They can take full advantage of virtualization's benefits, including the following:

- enhanced flexibility
- portability
- verification
- version control
- signing
- better licensing terms

Features of Open Virtualization Format

Key features of OVF are as follows:

Validation support. OVF supports the validation of every VM and the complete package.

Supports single and multiple VM configurations. OVF supports both single VM packages and complex multi-tier package services involving more than one interdependent VM.

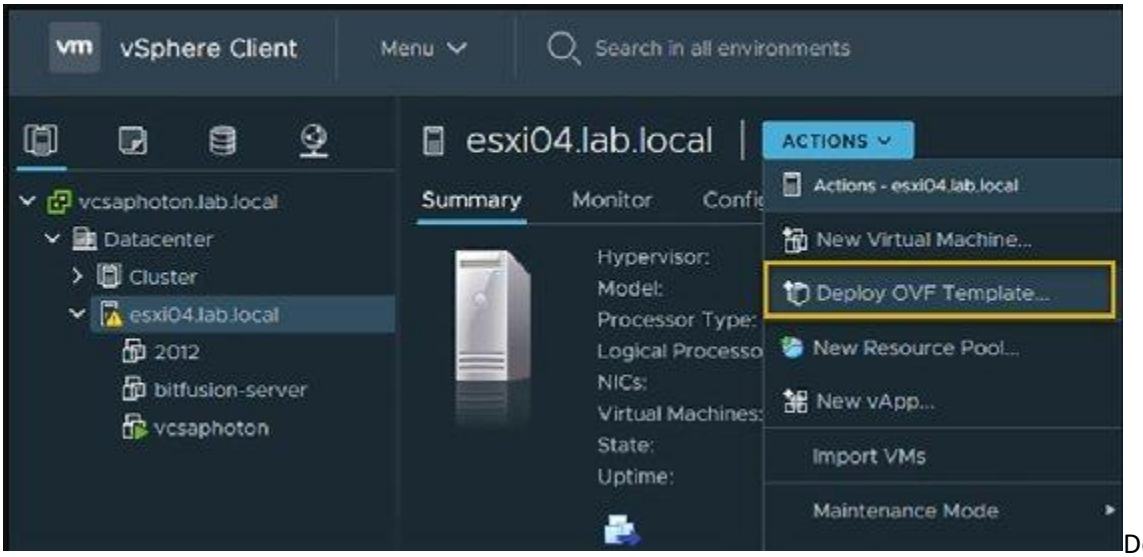
Content verification support. Depending on the industry-standard public key infrastructure (PKI), OVF enables integrity checking and content verification.

Licensing support. OVF supports management and software licensing strategies.

Platform-independent. OVF was designed to be platform-independent, whether it's a guest OS, host platform or virtualization platform.

Extensible. OVF can support new technological advancements in virtualization and virtual appliances.

Portable packaging. OVF allows vendors to add platform-specific enhancements to their appliances and software.



Deploying an OVF template in vSphere Client.

Open Virtualization Format package

An OVF package is a group of files required to import the VM. These files are generally found in the same folder and they can be compressed, digitally signed, encrypted and archived. The package consists of metadata and file elements that describe the VMs, as well as additional information required to deploy and operate the applications in the package.

The metadata or descriptor file is one of the components of the OVF file. This extensible markup language (XML) document with the extension of .ovf includes details such as the location of virtual disks associated with the VMs, plus information about managing the VMs during import.

In addition, the OVF package typically includes the following files:

- The manifest file with a .mf extension that references the OVF files and their checksum.
- A certificate file with the .cert extension if a user chooses to digitally sign the OVF file.
- Files comprising virtual disks in the format specified by the virtualization product that exported the virtual disks.

An OVF package can be used by multiple stakeholders, including the following:

- An independent software vendor (ISV) publishing a software solution.
- A data center operator aiming to transport a software solution between data centers.
- A customer trying to archive software.

In general, any user or use case that can benefit from a standardized package for a software solution can use the OVF standard.

Uploading OVF template in vSphere Client.

Distribution of Open Virtualization Format packages

OVF packages can be distributed either as a single file or as a set of files.

Distribution as a single file

The OVF package is stored as a single file in the tar format. In OVF tar files, duplication is not allowed within the archive. The files appear in the following order inside the archive:

- .ovf descriptor
- .mf manifest (optional)
- .cert certificate (optional)

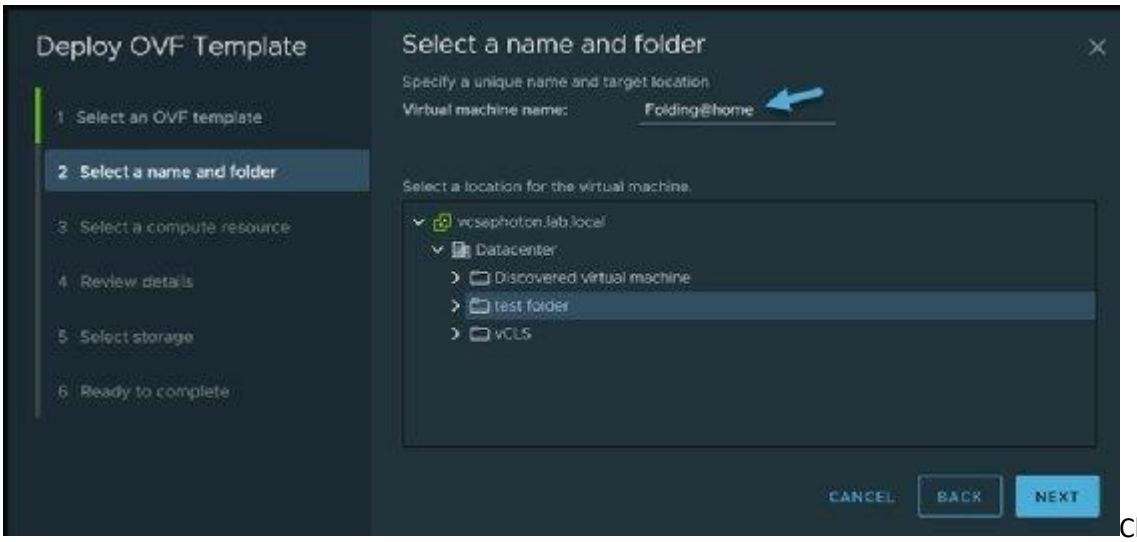
To find a file, a tar extraction tool scans the whole archive, even if the requested files are found in the beginning.

Distribution as a set of files

The OVF package can also be available as a set of files. This is common for applications involving a standard web browser.

The DMTF provides this example of an OVF package as a set of files:

http://xyzwebsite/virtualappliances/package.ovf
http://xyzwebsite/virtualappliances/virtualdisk1.vmdk
http://xyzwebsite/virtualappliances/virtualdisk2.vmdk
http://xyzwebsite/virtualappliances/additionalresource.iso



Choosing a location and naming for the VM appliance created using the OVF template.

Open Virtual Appliance in an OVF File

An Open Virtual Appliance (OVA) is an OVF package in a single file archive. Its file extension is .ova. An OVA package, which is in the tar format, contains the files comprising an OVF package.

The OVA package is one large file that doesn't have the flexibility of the OVF package, which is a series of uncompressed files. This is because users can access the individual disk images in the files in the OVF package -- which they can't do with the OVA package. Another drawback of OVA packages is that it takes longer to export and import them.

What is LAMP?

LAMP is an open-source Web development platform that uses ***Linux*** as the operating system, ***Apache*** as the Web server, ***MySQL*** as the relational database management system and ***PHP/Perl/Python*** as the object-oriented scripting language.

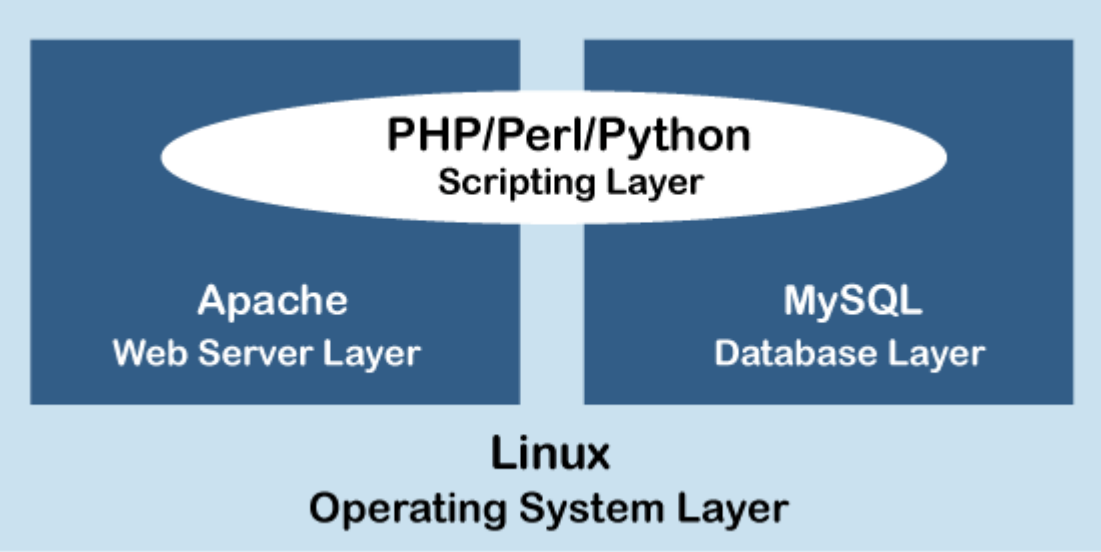
Sometimes LAMP is referred to as a LAMP stack because the platform has four layers. Stacks can be built on different operating systems.

LAMP is a example of a web service stack, named as an ***acronym***. The LAMP components are largely interchangeable and not limited to the original selection. LAMP is suitable for building dynamic web sites and web applications.

Since its creation, the LAMP model has been adapted to another component, though typically consisting of free and open-source software.

Developers that use these tools with a Windows operating system instead of Linux are said to be using ***WAMP***, with a Macintosh system ***MAMP***, and with a Solaris system ***SAMP***.

Linux, Apache, MySQL and PHP, all of them add something unique to the development of high-performance web applications. Originally popularized from the phrase Linux, Apache, MySQL, and PHP, the acronym LAMP now refers to a generic software stack model.



The modularity of a LAMP stack may vary. Still, this particular software combination has become popular because it is sufficient to host a wide variety of website frameworks, such as ***Joomla, Drupal***, and ***WordPress***.

The components of the LAMP stack are present in the software repositories of the most Linux distributions. The LAMP bundle can be combined with many other free and open-source software packages, such as the following:

- netsniff-ng for security testing and hardening
- intrusion prevention (IPS) system and Snort an intrusion detection (IDS)
- RRD tool for diagrams
- Nagios, Cacti, or Collectd for monitoring

LAMP Stack Components

Linux based web servers consist of four software components. These components are arranged in layers supporting one another and make up the software stack. Websites and Web Applications run on top of this underlying stack. The common software components are as follows:

1. **Linux:** Linux started in 1991. It sets the foundation for the stack model. All other layers are run on top of this layer.
   It is an open-source and free operating system. It is endured partly because it's flexible, and other operating systems are harder to configure.
2. **Apache:** The second layer consists of web server software, typically Apache Web Server. This layer resides on top of the Linux layer.
   Apache HTTP Server is a free web server software package made available under an open-source license. It used to be known as Apache Web Server when it was created in 1995. It offers a secure and extendable Web server that's in sync with current HTTP standards. Web servers are responsible for translating from web browsers to their correct website.
3. **MySQL:** MySQL is a relational database management system used to store application data. It is an open-source and keeps all the data in a format that can easily be queried with the SQL language.
   SQL works great with well-structured business domains, and a great workhorse that can

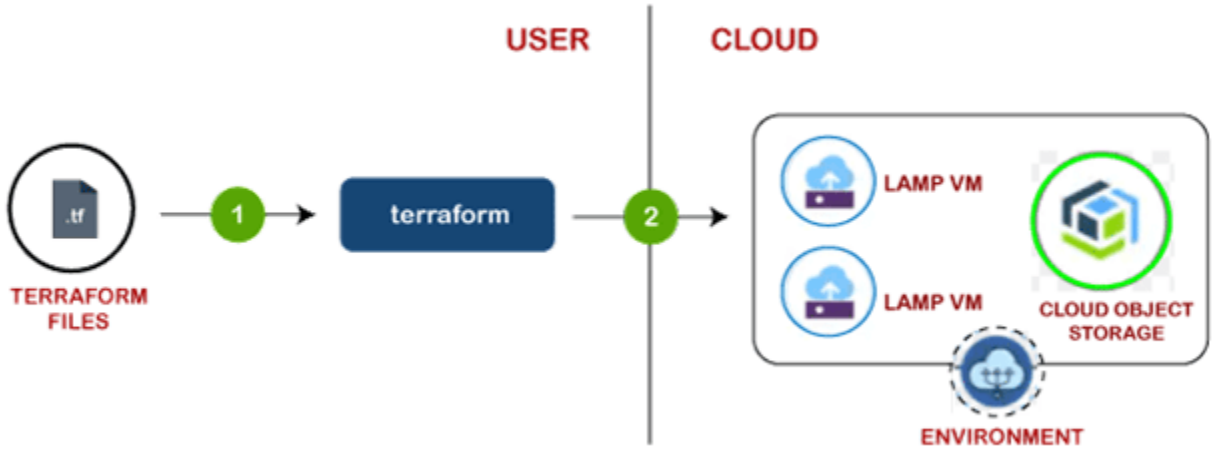
handle even the most extensive and most complicated websites with ease. MySQL stores details that can be queried by scripting to construct a website. MySQL usually sits on top of the Linux layer alongside Apache. In high-end configurations, MySQL can be offloaded to a separate host server.

4. **PHP:** The scripting layer consists of PHP and other similar web programming languages. The PHP open-source scripting language works with Apache to create dynamic web pages. We cannot use HTML to perform dynamic processes such as pulling data out of a database. To provide this type of functionality, we drop PHP code into the parts of a page that you want to be dynamic. Websites and Web Applications run within this layer. PHP is designed for efficiency. It makes programming easier and allowing to write new code, hit refresh, and immediately see the resulting changes without the need for compiling.

LAMP Architecture

LAMP has classic layered architecture, with Linux at the lowest level. The next layer is Apache and MySQL, followed by PHP.

Although PHP is at the top or presentation layer, the PHP component sits inside Apache.



The LAMP stack order of execution shows how the elements interoperate. The process starts when the Apache webserver receives requests for web pages from a user's browser. If the request is for a PHP file, Apache passes the request to PHP, which loads the file and executes the code contained in the file. PHP also communicates with MySQL to fetch any data referenced in the code.

PHP then uses the code in the file and the data from the database to create the HTML that browsers require to display web pages. The LAMP stack is efficient at handling not only static web pages but also dynamic pages where the content may change each time it is loaded depending on the date, time, user identity and other factors.

After running the file code, PHP then passes the resulting data back to the Apache webserver to send to the browser. It can also store this new data in MySQL. And of course, all of these operations are enabled by the Linux operating system running at the base of the stack.

Flexibility

Although LAMP uses Linux as the OS, we can use the other components with an alternative OS to meet specific needs. For example, there is a WAMP stack, which uses Microsoft Windows.

LAMP is open source and non-proprietary so we can avoid lock-in. We have the flexibility to select the right components for specific projects or business requirements.

LAMP offers flexibility in other ways as well. Apache is modular in design, and we will find there are existing, customizable modules available for many different extensions. These modules range from support for other languages to authentication capabilities.

LAPP

The LAPP stack is an open source web platform that can be used to run dynamic web sites and servers. It is considered by many to be a powerful alternative to the more popular LAMP stack and includes Linux, Apache, PostgreSQL (instead of MySQL) and PHP, Python and Perl.

Difference Between RSS and ATOM

DifferencesTechnologyWebsite

RSS (Really Simple Syndication) and Atom are web feed technologies for distributing and syndicating content from websites or blogs. They provide similar features but differ in their implementation and characteristics.

RSS (Really Simple Syndication) is a web feed type that allows users to access updates from websites or blogs without having to visit each one separately. Atom

is a standardized web feed format that allows content producers to syndicate their information.

Read this article to find out more about RSS and ATOM and how they are different from each other.

What is RSS?

RSS (Really Simple Syndication) is a web feed type that allows users to access updates from websites or blogs without having to visit each one separately. It allows content creators to share their information in a standardized manner, making it easier for users to subscribe and automatically receive updates.

Here's a detailed explanation of RSS

Structure

- RSS feeds are written in XML (eXtensible Markup Language), a markup language that is used to encode structured data. Tags are used in XML to define elements and their attributes.
- An RSS feed consists of a root element named that contains the complete feed. There are two necessary child components within the element: and .
- The element contains the feed's metadata, such as the title, description, and link to the website or blog.
- Multiple items are represented using elements within the element. Each element represents a specific piece of content, such as a blog post or a news article, and includes details such as the title, description, link, and publication date.

Subscription and Usage

- An RSS reader or aggregator can be used to subscribe to an RSS feed. These technologies aggregate and show content from subscribed feeds in a single location.
- To subscribe, users normally require the RSS feed URL, which is usually indicated by an RSS symbol on a website or provided by a dedicated feed URL.
- Once subscribed, the RSS reader checks the feeds on a regular basis for updates and notifies users of new information or shows it in chronological order within the reader interface.
- Depending on the RSS reader's functionality, users can customize their subscriptions, organize feeds into categories, and mark articles as read or starred.

Benefits of RSS

- RSS allows users to access and consume content from various sources in one location, reducing the need to visit each website separately
- Users can subscribe to specific websites or blogs, tailoring their information consumption to their specific interests.
- RSS readers fetch updates automatically, ensuring users obtain the most recent content without having to manually check websites.
- Users can manage their subscriptions, add or remove feeds, and limit how much content they consume.

What is ATOM?

Atom is a standardized web feed format that allows content producers to syndicate their information. Atom, like RSS (Really Simple Syndication), allows users to subscribe to updates from websites or blogs without visiting them individually. Atom was created as an alternative to RSS with the goal of improving extensibility while adopting a more modern approach to web distribution

Here's a detailed explanation of Atom:

Structure

- Atom feeds are written in XML (eXtensible Markup Language) and must follow strict XML grammar. This ensures that different implementations are consistent and compatible.
- An atom feed begins with a root element that contains the whole feed. There are several essential and optional elements within the element that reflect various characteristics of the feed.
- The element often contains metadata such as the feed title, subtitle, and links to the feed's connected website or blog
- Each item in an Atom feed, like RSS, is represented by an element. The element includes details such as the title, content, author, publication date, and links to the entry's web page.

Extensibility and Flexibility

- Atom's extensibility is one of its primary advantages. It provides a framework for adding new elements, properties, and namespaces to handle different forms of content or custom metadata.
- Content providers can build their own extensions by using XML namespaces to add specialised components or characteristics to the Atom feed, allowing for customization and richer content representations

- Atom's versatility makes it suitable for a wide range of content types, including articles, blog postings, podcasts, and more, because it can adapt to the specific requirements of different domains.

Subscription and Usage

- Similar to RSS, users can subscribe to Atom feeds using feed readers or aggregators.
- Subscribing to an Atom feed involves providing the reader with the feed's URL or clicking on an Atom icon or link on a website to initiate the subscription process.
- Once subscribed, the feed reader retrieves the Atom feed on a regular basis and displays the most recent content to the user. Within the feed reader interface, users may browse, organize, and manage their subscriptions

Difference between RSS and ATOM

The following table highlights the major differences between RSS and ATOM:

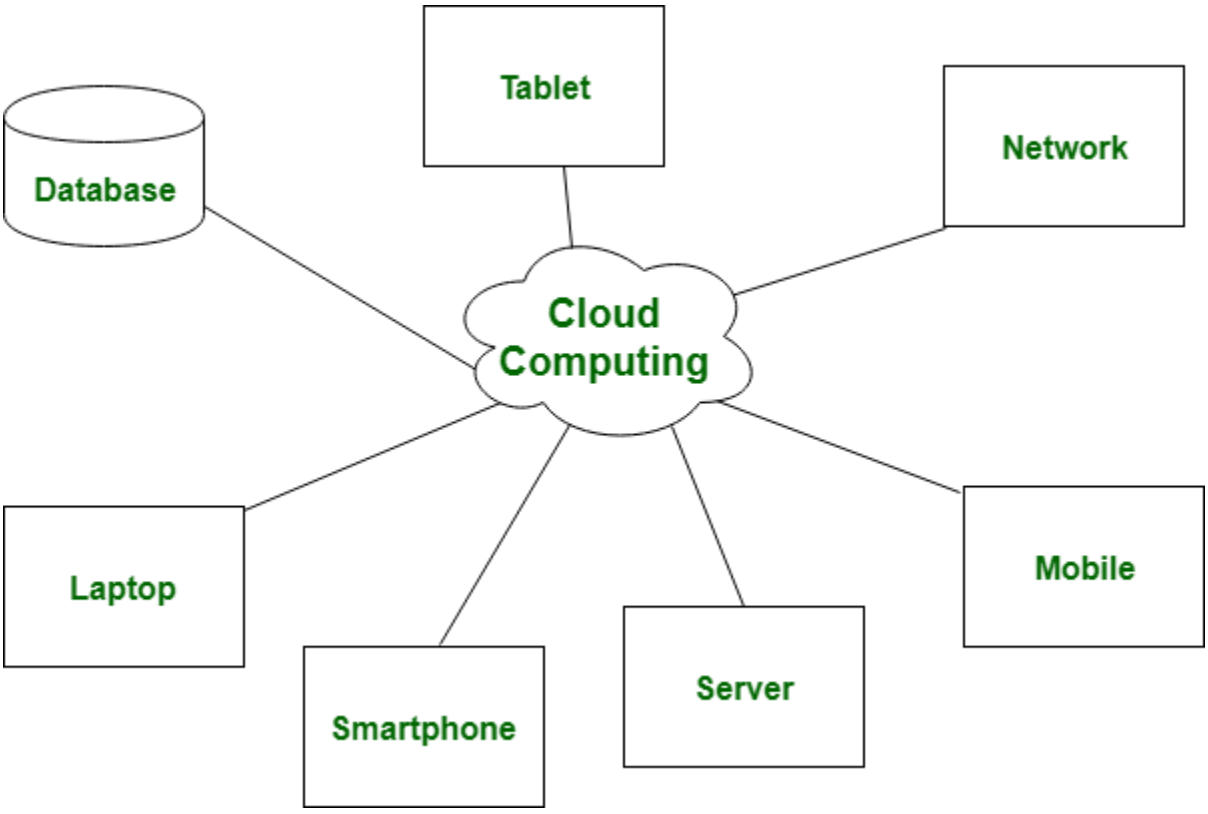
Characteristics	**RSS**	**ATOM**
Version	Various versions (RSS 0.9, RSS 1.0, and RSS 2.0)	The core format has remained relatively stable.
Syntax	Less strict XML syntax	Strict XML syntax
Compatibility	Well-established support across platforms, CMS, and readers	Increasing support across platforms, CMS, and readers
Ecosystem	Established ecosystem with a wide range of tools and resources	Growing ecosystem with increasing adoption and support
Standardization	No formal standard exists; multiple versions exist.	IETF standard with a formalized specification
Extensibility	Limited extensibility	More flexible and extensible, it allows custom elements and attributes.
Format	XML-based	XML-based

Structure	Hierarchical structure with predefined elements	Hierarchical structure with predefined elements but allows custom extensions
Development	Developed by Netscape in the late 1990s	Developed as an alternative to RSS, it was standardized by the IETF in 2005.

Difference between Cloud Computing and Grid Computing

Cloud Computing:

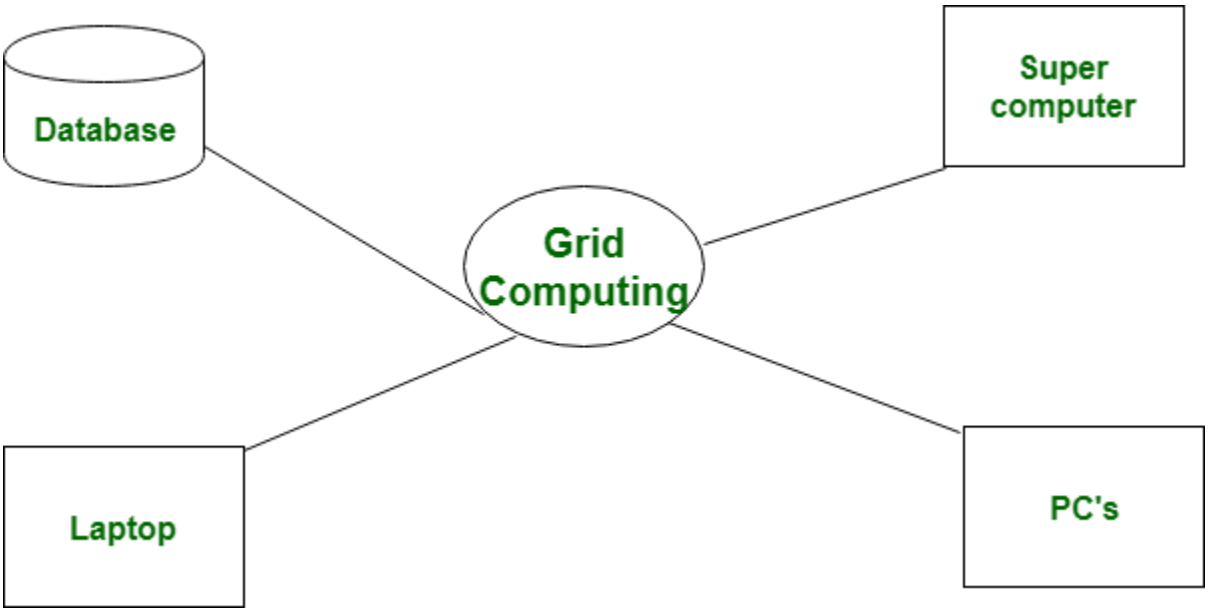
Cloud Computing is a Client-server computing architecture. In cloud computing, resources are used in centralized pattern and cloud computing is a high accessible service. It is a pay and use business means, in cloud computing, the users pay for the use



Grid Computing:

Grid Computing is a Distributed computing architecture. In grid computing,

resources are used in collaborative pattern, and also in grid computing, the users do not pay for use.



Let's see the difference between cloud and grid computing which are given below:

S.NO	Cloud Computing	Grid Computing
1.	Cloud computing is a Client-server computing architecture.	While it is a Distributed computing architecture.
2.	Cloud computing is a centralized executive.	While grid computing is a decentralized executive.
3.	In cloud computing, resources are used in centralized pattern.	While in grid computing, resources are used in collaborative pattern.
4.	It is more flexible than grid computing.	While it is less flexible than cloud computing.
5.	In cloud computing, the users pay for the use.	While in grid computing, the users do not pay for use.

S.NO	Cloud Computing	Grid Computing
6.	Cloud computing is a high accessible service.	While grid computing is a low accessible service.
7.	It is highly scalable as compared to grid computing.	While grid computing is low scalable in comparison to cloud computing.
8.	It can be accessed through standard web protocols.	While it is accessible through grid middleware.
9.	Cloud computing is based on service-oriented.	Grid computing is based on application-oriented.
10.	Cloud computing uses service like IAAS, PAAS, SAAS.	Grid computing uses service like distributed computing, distributed pervasive, distributed information.

What is Google App Engine

A scalable runtime environment, Google App Engine is mostly used to run Web applications. These dynamic scales as demand change over time because of Google's vast computing infrastructure. Because it offers a secure execution environment in addition to a number of services, App Engine makes it easier to develop scalable and high-performance Web apps. Google's applications will scale up and down in response to shifting demand. Croon tasks, communications, scalable data stores, work queues, and in-memory caching are some of these services.

The App Engine SDK facilitates the testing and professionalization of applications by emulating the production runtime environment and allowing developers to design and test applications on their own PCs. When an application is finished being produced, developers can quickly migrate it to App Engine, put in place quotas to control the cost that is generated, and make the programmer available to everyone. Python, Java, and Go are among the languages that are currently supported.

The development and hosting platform Google App Engine, which powers anything from web programming for huge enterprises to mobile apps, uses the same infrastructure as Google's large-scale internet services. It is a fully managed PaaS (platform as a service) cloud computing platform that uses in-built services to run your apps. You can start creating almost immediately after receiving the software development kit (SDK). You may immediately access the Google app developer's manual once you've chosen the language you wish to use to build your app.

After creating a Cloud account, you may Start Building your App

- Using the Go template/HTML package
- Python-based webapp2 with Jinja2
- PHP and Cloud SQL
- using Java's Maven

The app engine runs the programmers on various servers while "sandboxing" them. The app engine allows the program to use more resources in order to handle increased demands. **The app engine powers programs like Snapchat, Rovio, and Khan Academy.**

Features of App Engine

Runtimes and Languages

To create an application for an app engine, you can use Go, Java, PHP, or Python. You can develop and test an app locally using the SDK's deployment toolkit. Each language's SDK and nun time are unique. Your program is run in a:

- Java Run Time Environment version 7
- Python Run Time environment version 2.7
- PHP runtime's PHP 5.4 environment
- Go runtime 1.2 environment

Generally Usable Features

These are protected by the service-level agreement and depreciation policy of the app engine. The implementation of such a feature is often stable, and any changes made to it are backward-compatible. These include communications, process management, computing, data storage, retrieval, and search, as well as app configuration and management. Features like the HRD migration tool,

Google Cloud SQL, logs, datastore, dedicated Memcached, blob store, Memcached, and search are included in the categories of data storage, retrieval, and search.

Features in Preview

In a later iteration of the app engine, these functions will undoubtedly be made broadly accessible. However, because they are in the preview, their implementation may change in ways that are backward-incompatible. Sockets, MapReduce, and the Google Cloud Storage Client Library are a few of them.

Experimental Features

These might or might not be made broadly accessible in the next app engine updates. They might be changed in ways that are irreconcilable with the past. The “trusted tester” features, however, are only accessible to a limited user base and require registration in order to utilize them. The experimental features include Prospective Search, Page Speed, OpenID, Restore/Backup/Datastore Admin, Task Queue Tagging, MapReduce, and Task Queue REST API. App metrics analytics, datastore admin/backup/restore, task queue tagging, MapReduce, task queue REST API, OAuth, prospective search, OpenID, and Page Speed are some of the experimental features.

Third-Party Services

As Google provides documentation and helper libraries to expand the capabilities of the app engine platform, your app can perform tasks that are not built into the core product you are familiar with as app engine. To do this, Google collaborates with other organizations. Along with the helper libraries, the partners frequently provide exclusive deals to app engine users.

Advantages of Google App Engine

The Google App Engine has a lot of benefits that can help you advance your app ideas. This comprises:

1. **Infrastructure for Security:** The Internet infrastructure that Google uses is arguably the safest in the entire world. Since the application data and code are hosted on extremely secure servers, there has rarely been any kind of illegal access to date.

2. **Faster Time to Market:** For every organization, getting a product or service to market quickly is crucial. When it comes to quickly releasing the product, encouraging the development and maintenance of an app is essential. A firm can grow swiftly with Google Cloud App Engine's assistance.
3. **Quick to Start:** You don't need to spend a lot of time prototyping or deploying the app to users because there is no hardware or product to buy and maintain.
4. **Easy to Use:** The tools that you need to create, test, launch, and update the applications are included in Google App Engine (GAE).
5. **Rich set of APIs & Services:** A number of built-in APIs and services in Google App Engine enable developers to create strong, feature-rich apps.
6. **Scalability:** This is one of the deciding variables for the success of any software. When using the Google app engine to construct apps, you may access technologies like GFS, Big Table, and others that Google uses to build its own apps.
7. **Performance and Reliability:** Among international brands, Google ranks among the top ones. Therefore, you must bear that in mind while talking about performance and reliability.
8. **Cost Savings:** To administer your servers, you don't need to employ engineers or even do it yourself. The money you save might be put toward developing other areas of your company.
9. **Platform Independence:** Since the app engine platform only has a few dependencies, you can easily relocate all of your data to another environment.

Comparisons between Azure Vs AWS

Azure Cloud Service: Azure Cloud Service cloud service is provided by Microsoft in which the developers store the applications. It uses SQL databases, MySQL, Cosmos DB, etc for databases. In Azure, some tasks are provided which are: maintenance of servers, database software, operating system, etc. Azure is a good platform that is offered to developers to transfer or store their applications to the cloud.

Amazon Web Services (AWS): It is a type of cloud service. In AWS, For computation, Elastic Compute Cloud is used. AWS uses Simple Storage Service to store. In AWS for database, RDS and DynamoDB are used. The main difference between Azure and AWS is that Azure cloud service is provided by Microsoft. On the other hand, AWS cloud service is provided by Amazon.

Let's see the difference between Azure and AWS:

Azure	AWS
Azure was launched in 2010	AWS was launched in 2006
In azure, For computation, virtual machines are used.	In AWS, For computation, Elastic Compute Cloud is used.
Azure uses blocks to store.	While it uses Simple Storage Service to store.
Azure is a virtual network.	While AWS is a virtual private cloud.
Azure cloud spans 140 availability zones. (as of Feb 2023).	AWS cloud spans 61 availability zones. (as of Feb 2023).
SQL databases, MySQL, Cosmos DB, etc., are used in azure for databases.	In AWS for database, RDS and DynamoDB are used.
The pricing model offered by Microsoft is less flexible.	The pricing model offered by AWS is more flexible.
There are four levels of certification in Azure.	AWS has six levels of specialty certifications.
Microsoft Azure has a 22% market share.	Amazon Web Services has a 33% market share
Some famous clients of Azure are: Nike, Dell, Starbucks, etc	Some famous clients of AWS area as follws : Netflix, Adobe, Spotify, etc

Emerging Technologies Enabled by the Cloud

echnology has advanced considerably. Our lives have been radically changed by technology. Technology has given us far more than we can utilize with the daily addition of new features and upgrades. Cloud computing is a significant shift brought about by technology in our lives. It has fundamentally altered how business is conducted.

To use cloud computing effectively, there have been several technological advancements made. In the process, these technologies are revolutionizing not only the cloud computing environment but also the whole computer industry.

It was the ground-breaking new technology that would revolutionize computing as we knew it when cloud computing first made headlines, and boy did it create waves! The cloud has changed from the leading edge of new technology to the new technology itself. Without cloud computing as the enabler, emerging technologies like artificial intelligence (AI) and the internet of things (IoT) would not be feasible. Cloud computing makes these new technologies viable because they need massive amounts of data and lightning-fast processing speeds on demand.

Here we will learn about the new emerging technologies enabled by the Cloud –

Home Automation

Home automation technology constantly expands its versatility by introducing contemporary features to meet people's rising needs. The creation and deployment of a new home automation system that takes advantage of cloud computing as a service. The cloud server handles and controls user data, information, and appliance status. The hardware interface module interfaces with the sensors and actuators to deliver physical service. The Home Server is in charge of configuring the devices and providing the user interface. The online

services are deployed via the cloud, which is mostly required for data security and availability.

Serverless Computing

Serverless computing addressed these problems by managing important maintenance and scaling demands for businesses, allowing them to focus on other critical operations in their cloud service providers' systems.

The trend for pay-as-you-go and pay-for-use computing models has taken up serverless computing, addressing the bulk of software load. This function-as-a-service paradigm allowed cloud systems to run more quickly and effectively.

The "serverless" cloud computing application execution paradigm offloads to cloud service providers and tools all responsibility for ordinary infrastructure management operations such as scaling, scheduling, patching, provisioning, and so on. Furthermore, it allows developers to focus their time and attention on the specific business logic for their apps or processes. They don't require any infrastructure to administer or operate.

Blockchain

Blockchain is distributed database that stores data electronically in blocks and then uses cryptography to establish a chain of these data blocks. Blockchain blocks are essentially information containers that hold encrypted and stored data. Blockchain applications in cloud computing are connected to the Cloud of Things (CoT), a hybrid cloud computing and the Internet of Things (IoT).

Blockchain may be used for secure network management in blockchain-based cloud computing by hosting blockchain network as Blockchain as a Service (BaaS) in a cloud environment. BaaS enables IoT applications by providing blockchain-enabled services such as smart contract services, user transaction verification, and cloud blockchain storage.

Augmented Reality Cloud

Augmented reality digitally augments the world around the user when seen through a phone or other digital viewing device. AR filters are available on social media sites such as Snapchat and Instagram. An AR Cloud is a digital 3D representation of the actual world. An AR cloud's objective is to allow for the ongoing sharing of AR experiences across various devices. It will give data and services directly relevant to the user's actual surroundings. Consider the following scenario: a user walks into a spa. In such instances, consumers may use their AR glasses or smartphone to check out its reviews and a list of services offered, therapists available, and other information about the firm.

An AR Cloud is a scalable and shared point cloud that recognizes the geometry and forms of the actual world. It is an instant, universal localizer that works from any angle on numerous devices simultaneously. It gives multi-user interaction in real-time, even with faraway users.

Machine Learning

Machine Learning assists users in making predictions and developing algorithms that can learn independently from past data. However, various machine learning algorithms, such as SVM, Linear Regression, Logistic Regression, Decision Tree, Nave Bayes, K-Means, Random Forest, Gradient Boosting algorithms, and so on, require massive amounts of storage, making it difficult for data scientists and machine learning professionals to work with. In such cases, cloud computing becomes a game changer for implementing machine learning models. Cloud computing aids in the enhancement and expansion of machine learning applications. Intelligent Cloud refers to the integration of machine learning with cloud computing.

Cloud operates on the premise of "pay for what you need." The Cloud's pay-per-use approach is ideal for businesses who want to use ML capabilities without incurring high costs. It lets you deal with machine learning functions without requiring strong data science knowledge. It makes it easier for us to experiment with different ML technologies and scales up when projects go into production and demand grows.

Final Thoughts

Since industrialization, and even before that, technology has advanced and influenced our lives. The aforementioned technologies are required to develop and use robots, chatbots, and a variety of other activities, including the realization of Human Intelligence. Investing in these top developing technologies now will improve your firm, provide security, and provide a roadmap for the future of cloud computing.

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.

What is Google Cloud Platform (GCP)?

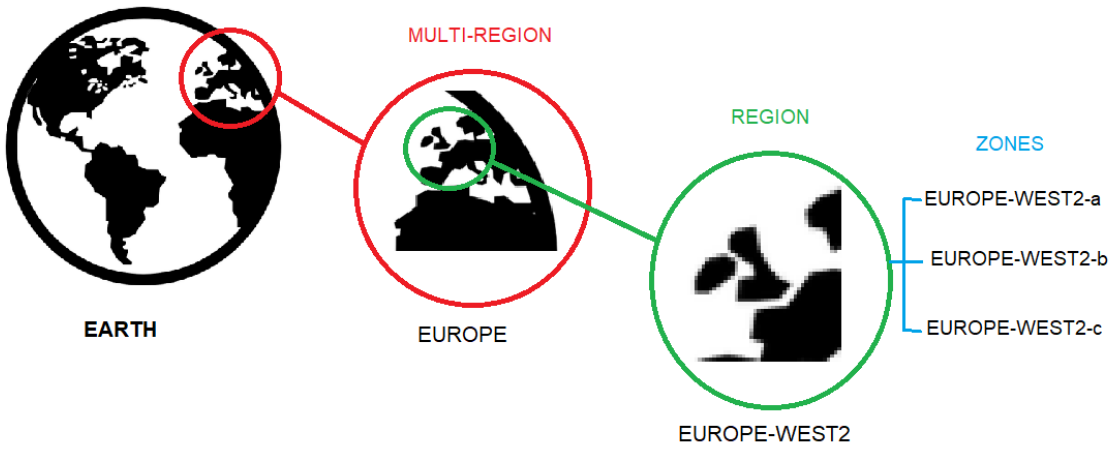
Before we begin learning about the Google Cloud Platform, we will talk about what is Cloud Computing. Basically, it is using someone else's computer over the internet. Examples- GCP, AWS, IBM Cloud, etc. Some interesting features of cloud computing are as follows:

- You get computing resources on-demand and self-service. The customer has to use a simple User Interface and they get the computing power, storage requirements, and network you need, without human intervention.
- You can access these cloud resources over the internet from anywhere on the globe.
- The provider of these resources has a huge collection of these resources and allocates them to customers out of that collection.
- The resources are elastic. If you need more resources you can get more, rapidly. If you need less, you can scale down back.

- The customers pay only for what they use or reserve. If they stop using resources, they stop paying.
-

What is Google Cloud Platform(GCP)?

Starting from 1998 with the launch of Google Search. google has developed one of the largest and most powerful IT Infrastructures in the world. Today, this infrastructure is used by billions of users to use services such as Gmail, YouTube, Google Photos, and Maps. In 2008, Google decided to open its network and IT infrastructure to business customers, taking an infrastructure that was initially developed for consumers' applications to public service and launching the Google Cloud platform.
All the services listed above are provided by Google hence the name Google Cloud Platform (GCP). Apart from these, there are many other services provided by GCP and also many concepts related to it that we are going to discuss in this article.
Regions and zones
Let's start at the finest grain level (i.e. the smallest or first step in the hierarchy), the **Zone**. A zone is an area where Google Cloud Platform Resources like virtual machines or storage are deployed.



For example, when you launch a virtual machine in GCP using Compute Engine, it runs in a zone you specify (suppose Europe-west2-a). Although people consider a zone as being sort of a GCP Data Center, that's not strictly accurate because a zone doesn't always correspond to one physical building. You can still visualize the zone that way, though.
Zones are grouped into **regions** which are independent geographic areas and much larger than zones (for example- all zones shown above are grouped into a single region Europe-west2) and you can choose what regions you want your GCP resources to be placed in. All the zones within a neighborhood have fast network connectivity among

them. Locations within regions usually have trip network latencies of under five milliseconds.

As a part of developing a fault-tolerant application, you'll need to spread your resources across multiple zones in a region. That helps protect against unexpected failures. You can run resources in different regions too. Lots of GCP customers do this, both to bring their applications closer to users around the world, and also to guard against the loss of a whole region, say, due to a natural disaster.

A few GCP Services supports deploying resources in what we call a **Multi-Region**. For example, Google Cloud Storage, lets you place data within the Europe Multi-Region. What that means is that it is stored redundantly in a minimum of two different geographic locations, separated by at least 160 kilometers within Europe. Previously, GCP had 15 regions. Visit cloud.google.com to ascertain what the entire is up to today.

What Are The Benefits Of Using Google Cloud Platform?

Following are some use cases of google cloud platform.

- **Scalability and flexibility:** You can scale the servers based up on the load if there is high incoming traffic then you can scale up and if the traffic is less then you can scale down which make it flexible and mostly suited for the business use cases.
- **Cost-effectiveness:** Google Cloud Platform follows the pay-as-you-go model which it will charge only for the how much time you use service and at level you are going to use the services.
- **High performance:** You can highly reliable on the infrastructure of the google cloud because they are highly available and spread across the multiple regions around the world.
- **Security:** Google Cloud Platform is more secure you can trust the service offered by the gcp for the security options such as encryption, access control, and data loss prevention.

What are the different types of Google Cloud Platform services?

Compute

- **Compute Engine:** It is used to provision the virtual machine machines to deploy the application with the need of your required ram,rom and security groups.
- **Google Kubernetes Engine(GKE):** Google cloud provide kubernetes(GKE) as an service where you can deploy the application and restof the things like autoscaling and load balancing will be taken care by the google cloud.
- **App Engine:** A scalable runtime environment, Google App Engine is mostly used to run Web applications. These dynamic scales as demand change over time because of Google's vast computing infrastructure. Because it offers a secure execution

environment in addition to a number of services, App Engine makes it easier to develop scalable and high-performance Web apps.

Storage

- **Cloud Storage:** You can store the data which is required to be highly available and which is in large amount.
- **Persistent Disk:** Presistent disk is an storage disk which can be attached to the virtual machine and can be reused for the another virtual machine.
- **Cloud SQL:** Cloud SQL is an fully managed service by the google cloud platform and it offers services like MySQL, PostgreSQL and SQL Server.

Networking

- **Virtual Private Cloud (VPC):** You can deploy your application in the private network which can be achieved by the google cloud.
- **Cloud Load Balancing:** This is most important service in the google cloud which is used to distribute the cloud across the multiple replicas of the applications.
- **Cloud CDN:** This is the service which will cache the content and delivers to the end users with the help of edge locations.

Data analytics

- **Bigquery:** All organizations look for unlocking business insights from their data. But it can be hard to scalably ingest, store, and analyze that data as it rapidly grows. Google's enterprise data warehouse called BigQuery, was designed to make large-scale data analysis accessible to everyone.
- **Dataflow:** It is the analysis of flow of data in control flow graph, i.e., the analysis that determines the information regarding the definition and use of data in program. With the help of this analysis, optimization can be done.
- **Pub/Sub:** Pub/sub Consider a scenario of synchronous message passing. You have two components in your system that communicate with each other. LeLet'ss call the sender and receiver. The receiver asks for a service from the sender and the sender serves the request and waits for an acknowledgment from the receiver. There is another receiver that requests a service from the sender. The sender is blocked since it hasn't yet received any acknowledgment from the first receiver. The sender isn't able to serve the second receiver which can create problems. To solve this drawback, the Pub-Sub model was introduced.

Machine learning

- **Vertex AI Platform:** As we know Artificial Intelligence (AI) has undergone advancements throughout the years leading to a transformation of industries and reshaping how businesses operate. The emergence of cloud-based AI platforms has further accelerated this revolution enabling organizations to leverage the potential of AI.
- **AI Platform Training:** You can train the AI model with the help of AI Platform Training in google cloud.

- **AI Platform Prediction:** You can make the predictions by using the your machine learning models.

Productivity and collaboration

- **Google Workspace:** Most of us are familiar with various Google Workspace products(also called G Suite) like Calendar, Drive, and Gmail, etc.
- **Cloud Identity and Access Management (IAM):** Identity Access Management is used by the root user (administrator) of the organization. The users represent one person within the organization, and the users can be grouped in that all the users will have the same privileges to the services.

How to get started with Google Cloud Platform?

Follow the steps mentioned below to get started with the Google Cloud Platform.

Step 1: Create an account in google cloud platform depending on your business need select the plan if you want to learn the service the create free account gcp will provide the $300 free credit to get you started.

Step 2: After creating free account it will validate till 90n days know you can start using by creating the project which means all the resources can be tracked under this project.

Step 3: Know you need to start the billing then only you can start use the service and also the billing will be on pay-as-you-go model you will be charged for the amount of resources you are going to use.

Step 4: You need to be careful while choosing the services. You must select the service according to need of you application.

Step 5: Once you have choosen the service know you can deploy you application into the cloud.

Step 6: After deploying the application know you can monitor you resource usage by using the services available in the gcp.

Google Cloud Platform use cases

Google Cloud Platform is well suited for the build and deploy and manage the applications.

- **E-commerce:** You deploy and manage the e-commerce websites by autoscaling and load balancing you can manage the millions of users and transactions.
- **Media and entertainment:** You can store the static and dynamic data can deliver it to the across the world with out any latency to the end users.
- **Financial services:** Google Cloud Platform is well suited for the sinical application because of the level of security it is offering.
- **Healthcare:** You can store the data of patient and take care the outcomes of the health of patient.

Google Cloud Platform security

Google Cloud Platform offers following security options.

- **Encryption:** Google cloud platform offers security like encryption at rest and in transit for all of your data.

- **Access control:** You can control the access to the individual users like which services they can access and which service they can't depending on the use cases.
- **Network Security:** You can create the VPC where you can secure the application by deploying the application in the private network and also you can configure the firewalls andsecurity groups etc.
- **Identity-Aware Proxy (IAP):** With IAP, users may manage application access according to their context and identity. It aids in preventing unwanted access.

Google Cloud Platform Future

Google Cloud Platform (GCP) is evolving constantly by expanding its resources and increasing its regions and availability zone across the world which make it mote available for the users to use reduces the latency. GCP is upgrading itself according to the market trends gcp play an major role in the upcoming years it will play major role it will helps for the businesses to thrive in the increasingly data-driven and interconnected world.

- Artificial intelligence (AI) and machine learning (ML).
- Edge computing.
- Data analytics and data management.
- Cybersecurity.
- Sustainability.

Google Cloud Platform pricing(GCP)

Google was the primary major Cloud provider to **bill by the second** instead of rounding up to greater units of your time for its virtual machines as a service offering. This may not sound like a big deal, but charges for rounding up can really add up for customers who are creating and running lots of virtual machines. Per second billing is obtainable for a virtual machine use through Compute Engine and for several other services too.

Compute Engine provides automatically applied use discounts which are **discounts** that you simply get for running a virtual machine for a big portion of the billing month. When you run an instance for at least 25% of a month, Compute Engine automatically gives you a reduction for each incremental minute you employ it. Here's one more way Compute Engine saves you money.

Normally, you choose a virtual machine type from a typical set of those values, but Compute Engine also offers custom virtual machine types, in order that you'll fine-tune the sizes of the virtual machines you use. That way, you'll **tailor your pricing** for your workloads.

Google Cloud Platform Open API's(GCP)

Some people are afraid to bring their workloads to the cloud because they're afraid they'll get locked into a specific vendor. But in many ways, Google gives customers the power to run their applications elsewhere, if Google becomes not the simplest provider for his or her needs. Here are some samples of how Google helps its customers **avoid feeling locked** in. GCP services are **compatible with open source**

products. For example, take Cloud Bigtable, a database that uses the interface of the open-source database Apache HBase, which provides customers the advantage of code portability. Another example, Cloud Dataproc provides the open-source big data environment Hadoop, as a managed service, etc.

Google Cloud Platform Certification(GCP)

Google Cloud Platform offers wide range of certifications to validate you skills some of the certifications as mentioned follows.

- **Foundational:** It is an basic certification to test your basics on google cloud platform like features,benefits and use cases of google cloud.
- **Associate Cloud Enginee:** Associate Cloud Enginee this certification will test your fundamentals on google cloud platform which are like deploying and maintaining the projects.
- **Professional Cloud Architect:** Professional Cloud Architect will test you depth knowledge on the complete overview of the services implementation of and managing the services of google cloud.
- **Professional Cloud DevOps Engineer:** Professional Cloud DevOps Engineer will test you knowledge on the services like deployment automation and scaling the application at the sudden loads.
- **Professional Cloud Network Engineer:** Professional Cloud Network Engineer will validates the you ability on the desging of the networks for the business use in cloud environments.

Why choose GCP?

- GCP allows you to choose between computing, storage, big data, machine learning, and application services for your web, mobile, analytics, and, back-end solutions.
- It’s global and it is cost-effective.
- It’s open-source friendly.
- It’s designed for security.

Advantages of Google Cloud Platform (GCP)

1. **Good documentation:** We are talking about many pages in total, including a reasonably detailed API Reference guide.
2. **Different storage classes for every necessity:** Regional (frequent use), Nearline (infrequent use), and Coldline (long-term storage).
3. **High durability:** This suggests that data survives even within the event of the simultaneous loss of two disks.
4. **Many regions available to store your data:** North Ameria, South America, Europe, Asia, and Australia.
5. The **“Console” tab** within the documentation allows you to **try for free of charge** different SDKs. It’s incredibly useful for developers

6. One of the simplest free layers within the industry. **$300 free credit** to start with any GCP product during the primary year. Afterward, 5 GB of Storage to use forever without any charges.

Disadvantages of Google Cloud Platform (GCP)

1. **The support fee is sort of hefty:** Around 150 USD per month for the foremost basic service (Silver class).
2. **Downloading data** from Google Cloud Storage is **expensive**. 0, 12 USD per GB.
3. Google Cloud Platform **web interface is somewhat confusing**. Sometimes I am lost while browsing around the menus.
4. Prices in both Microsoft Azure (around 0.018 USD per GB/month) or Backblaze B2 (about 0.005 USD per GB/month) are less than Google Cloud Storage.
5. It has a **high pricing schema**, almost like AWS S3, so it's easy to **urge unexpected costs** (e.g. number of requests, transfers, etc.).

Three Categories of Cloud Services

- **Infrastructure as a Service (IaaS):** It provides you with all the hardware components you require such as computing power, storage, network, etc.
- **Platform as a Service (PaaS):** It provides you with a platform that you can use to develop applications, software, and other projects.
- **Software as a Service (SaaS):** It provides you with complete software to use like Gmail, Google Drive, etc.