Kalinga University

Course-BCSAIMLCS/BCAAIML
Sem-IV
Subject- Cloud Computing and it's Security
Subject Code: BCSAIMLCS203/BCAAIML202

UNIT II

What are the Pros and Cons of Cloud Computing in Software Development?

Advantages of Cloud Computing in Development:

1. **Easier to set and use** : The cloud is all about using a server structure but not being bogged down with the details of its implementation. By letting a third party manage a cloud infrastructure for you, you and your team reap the benefits of using it without having to worry about how to setup or maintain it. This also means you and your team can get instant access to the software you need without having to worry about hardware specifications.

2. **Easily scalable based on your needs** :Services that provide cloud services will allow you to scale how many machines and processing power you can use at any time, making it an extremely flexible service. This has the side benefit of also providing you with virtually any amount of storage space required for your team's operational needs.

3. **Lower cost of adoption**: Being scalable brings a secondary advantage with it. Most of the cloud services on offer use a pay-as-you-go system, where the costs of using the cloud are reflected on the needs of the customer. This makes the initial adoption of a cloud service far simpler and budget-friendly, since there is no need to pay for physical hardware or any dedicated personnel to deal with the cloud infrastructure.

4. **Accessible anywhere** : At the end of the day the cloud is an internet-based service, which allows developers to use and take advantage of it no matter where they are. If they have a computer with minimum specs and access to the internet, any member of the team will be able to get to the cloud and be ready to work.

5. **Enhanced Collaboration amongst your team**: By letting your dev team have access to a central point for development, you enhance their ability to cooperate and work on projects together in real-time. This also gives the side benefit of anyone in your team having instant access to the most recent versions of anything the team is working on.

6. **Automatic Back-ups**: Companies that provide cloud services will usually perform routine back-ups of all the data stored in the cloud. This frees up your team from doing and managing it themselves and puts them at ease by creating more redundancy of their work.

But no service ever brings only advantages, and cloud computing is no exception. By choosing Cloud Computing for your software development cycle, be aware that it has its drawbacks.

Disadvantages of Cloud Computing Software Development

- **1. Subscription Costs:** Although the initial cost of setting up a cloud is far lower than setting up a server yourself, continued use of a cloud platform is not without its share of costs. Be sure you are prepared to deal with a subscription that can potentially rise as the needs of your team demand more from your cloud service of choice.

- **2. Security and Downtime**: Any cloud service has the potential risk of breaches and leaks. Also the services can physically fail, which translates to downtime on the customer's end. Although these failures are very infrequent, they still happen even to the most secure or stable systems.

- **There is an upside to using an established service in this case**: The companies offering these services also have experience with providing security and stability for these systems, and are more likely to have the resources, technology stacks and personnel to set up a stronger and more secure platform than anyone starting from scratch.

- **3. Loss of Control over the Infrastructure and Customization**: These services are provided off-premises, and thus you lose the ability to control the hardware and (depending on the service) the software you can use. If you have specific needs in your development cycle, take care when choosing an appropriate cloud-based platform that can fulfil them.

- **4. Internet Reliance**: Cloud services are extremely flexible and powerful, but all of that means nothing if your team has limited or inexistent access to the internet. Make sure your company and its employees have good internet access before committing to a cloud solution, or you'll risk not being able to use it properly. Also remember that by making local back-ups of your cloud's data, depending on the quantity of data being transferred, you can potentially throttle your internet connection. Be especially careful if your bandwidth is on the slower side.

- **5. Limited Migration**: By committing to a cloud solution, you are essentially transferring your data to a third party. This can be detrimental when you want to change your provider:

- **It can potentially expose your information**: By allowing your data to circulate on the internet, there are more points of failure where hackers and other bad-faith actors can access it. This risk is doubled if you want to create a backup from one service and then upload it to another.

- **Your cloud's storage size can grow more than you can locally store**: Cloud services are very good when it comes to abstracting your storage space, and make it easy to store your data without worrying about hardware limitations. However, huge data sizes make it harder to create local back-ups, since you'll need to replicate the hardware in terms of storage size to create a local back-up. Depending on the size of your data this can be more or less feasible to do, both in terms of physical space and costs.

Cloud Based Services

Cloud Computing can be defined as the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. Companies offering such kinds of cloud computing services are called cloud providers and typically charge for cloud computing services based on usage. Grids and clusters are the foundations for cloud computing.

Types of Cloud Computing

Most cloud computing services fall into five broad categories:

1. Software as a service (SaaS)
2. Platform as a service (PaaS)
3. Infrastructure as a service (IaaS)
4. Anything/Everything as a service (XaaS)
5. Function as a Service (FaaS)

Software as a Service(SaaS)

Software-as-a-Service (SaaS) is a way of delivering services and applications over the Internet. Instead of installing and maintaining software, we simply access it via the Internet, freeing ourselves from the complex software and hardware management. It removes the need to install and run applications on our own computers or in the data centers eliminating the expenses of hardware as well as software maintenance. SaaS provides a complete software solution that you purchase on a **pay-as-you-go** basis from a cloud service provider. Most SaaS applications can be run directly from a web browser without any downloads or installations required. The SaaS applications are sometimes called **Web-based software, on-demand software, or hosted software.**

Advantages of SaaS

1. **Cost-Effective:** Pay only for what you use.
2. **Reduced time:** Users can run most SaaS apps directly from their web browser without needing to download and install any software. This reduces the time spent in installation and configuration and can reduce the issues that can get in the way of the software deployment.
3. **Accessibility:** We can Access app data from anywhere.
4. **Automatic updates:** Rather than purchasing new software, customers rely on a SaaS provider to automatically perform the updates.
5. **Scalability:** It allows the users to access the services and features on-demand.

The various companies providing *Software as a service* are Cloud9 Analytics, Salesforce.com, Cloud Switch, Microsoft Office 365, Big Commerce, Eloqua, dropBox, and Cloud Tran.

Disadvantages of Saas :

1. **Limited customization**: SaaS solutions are typically not as customizable as on-premises software, meaning that users may have to work within the constraints of the SaaS provider's platform and may not be able to tailor the software to their specific needs.

2. **Dependence on internet connectivity**: SaaS solutions are typically cloud-based, which means that they require a stable internet connection to function properly. This can be problematic for users in areas with poor connectivity or for those who need to access the software in offline environments.

3. **Security concerns:** SaaS providers are responsible for maintaining the security of the data stored on their servers, but there is still a risk of data breaches or other security incidents.

4. **Limited control over data:** SaaS providers may have access to a user's data, which can be a concern for organizations that need to maintain strict control over their data for regulatory or other reasons.

Platform as a Service

PaaS is a category of cloud computing that provides a platform and environment to allow developers to build applications and services over the internet. PaaS services are hosted in the cloud and accessed by users simply via their web browser.
A PaaS provider hosts the hardware and software on its own infrastructure. As a result, PaaS frees users from having to install in-house hardware and software to develop or run a new application. Thus, the development and deployment of the application take place **independent of the hardware**.
The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. To make it simple, take the example of an annual day function, you will have two options either to create a venue or to rent a venue but the function is the same.

Advantages of PaaS:

1. **Simple and convenient for users:** It provides much of the infrastructure and other IT services, which users can access anywhere via a web browser.

2. **Cost-Effective:** It charges for the services provided on a per-use basis thus eliminating the expenses one may have for on-premises hardware and software.

3. **Efficiently managing the lifecycle:** It is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.

4. **Efficiency:** It allows for higher-level programming with reduced complexity thus, the overall development of the application can be more effective.

The various companies providing *Platform as a service* are Amazon Web services Elastic Beanstalk, Salesforce, Windows Azure, Google App Engine, cloud Bees and IBM smart cloud.

Disadvantages of Paas:

1. **Limited control over infrastructure:** PaaS providers typically manage the underlying infrastructure and take care of maintenance and updates, but this can also mean that users have less control over the environment and may not be able to make certain customizations.

2. **Dependence on the provider**: Users are dependent on the PaaS provider for the availability, scalability, and reliability of the platform, which can be a risk if the provider experiences outages or other issues.

3. **Limited flexibility:** PaaS solutions may not be able to accommodate certain types of workloads or applications, which can limit the value of the solution for certain organizations.

What are Web Services?

The Internet is the worldwide connectivity of hundreds of thousands of computers of various types that belong to multiple networks. On the World Wide Web, a web service is a standardized method for propagating messages between client and server applications. A web service is a software module that is intended to carry out a specific set of functions. Web services in cloud computing can be found and invoked over the network. The web service would be able to deliver functionality to the client that invoked the web service.

A web service is a set of open protocols and standards that allow data to be exchanged between different applications or systems. Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks such as the Internet in a similar way to inter-process communication on a single computer.

Any software, application, or cloud technology that uses standardized web protocols (HTTP or HTTPS) to connect, interoperate, and exchange data messages – commonly XML (Extensible Markup Language) – across the internet is considered a web service.
Web services have the advantage of allowing programs developed in different languages to connect with one another by exchanging data over a web service between clients and servers. A client invokes a web service by submitting an XML request, which the service responds with an XML response**.**

Functions of Web Services

- It's possible to access it via the internet or intranet networks.
- XML messaging protocol that is standardized.
- Operating system or programming language independent.
- Using the XML standard, it is self-describing.
- A simple location approach can be used to locate it.

Components of Web Service

XML and HTTP is the most fundamental web services platform. The following components are used by all typical web services:

SOAP (Simple Object Access Protocol)

SOAP stands for "Simple Object Access Protocol." It is a transport-independent messaging protocol. SOAP is built on sending XML data in the form of SOAP Messages. A document known as an XML document is attached to each message. Only the structure of the XML document, not the content, follows a pattern. The best thing about Web services and SOAP is that everything is sent through HTTP, the standard web protocol.

A root element known as the element is required in every SOAP document. In an XML document, the root element is the first element. The "envelope" is separated into two halves. The header comes first, followed by the body. The

routing data, or information that directs the XML document to which client it should be sent to, is contained in the header. The real message will be in the body.

UDDI (Universal Description, Discovery, and Integration)

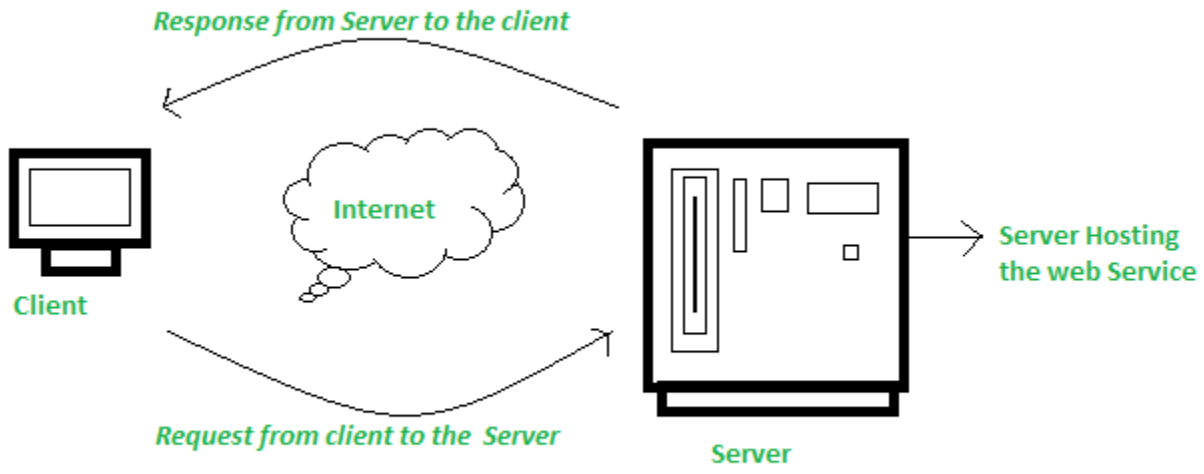
UDDI is a standard for specifying, publishing and discovering a service provider's online services. It provides a specification that aids in the hosting of data via web services. UDDI provides a repository where WSDL files can be hosted so that a client application can discover a WSDL file to learn about the various actions that a web service offers. As a result, the client application will have full access to the UDDI, which serves as a database for all WSDL files.
The UDDI registry will hold the required information for the online service, just like a telephone directory has the name, address, and phone number of a certain individual. So that a client application may figure out where it is.

WSDL (Web Services Description Language)

If a web service can't be found, it can't be used. The client invoking the web service should be aware of the location of the web service. Second, the client application must understand what the web service does in order to invoke the correct web service. The WSDL, or Web services description language, is used to accomplish this. The WSDL file is another XML-based file that explains what the web service does to the client application. The client application will be able to understand where the web service is located and how to use it by using the WSDL document.

How Does Web Service Work?

The diagram depicts a very simplified version of how a web service would function. The client would use requests to send a sequence of web service calls to a server that would host the actual web service.



Remote procedure calls are what are used to make these requests. Calls to methods hosted by the relevant web service are known as Remote Procedure Calls (RPC). Example: Flipkart offers a web service that displays prices for items offered on Flipkart.com. The front end or presentation layer can be written in .Net or Java, but the web service can be communicated using either programming language.
The data that is exchanged between the client and the server, which is XML, is the most important part of a web service design. XML (Extensible markup language) is a simple intermediate language that is understood by various

programming languages. It is a counterpart to HTML. As a result, when programs communicate with one another, they do so using XML. This creates a common platform for applications written in different programming languages to communicate with one another.
For transmitting XML data between applications, web services employ SOAP (Simple Object Access Protocol). The data is sent using standard HTTP. A SOAP message is data that is sent from the web service to the application. An XML document is all that is contained in a SOAP message. The client application that calls the web service can be created in any programming language because the content is written in XML.

Features/Characteristics Of Web Service

Web services have the following features:

(a) XML Based: The information representation and record transportation layers of a web service employ XML. There is no need for networking, operating system, or platform binding when using XML. At the middle level, web offering-based applications are highly interoperable.

(b) Loosely Coupled: A customer of an internet service provider isn't necessarily directly linked to that service provider. The user interface for a web service provider can change over time without impacting the user's ability to interact with the service provider. A strongly coupled system means that the patron's and server's decisions are inextricably linked, indicating that if one interface changes, the other should be updated as well.
A loosely connected architecture makes software systems more manageable and allows for easier integration between different structures.

(c) Capability to be Synchronous or Asynchronous: Synchronicity refers to the client's connection to the function's execution. The client is blocked and the client has to wait for the service to complete its operation, before continuing in synchronous invocations. Asynchronous operations allow a client to invoke a task and then continue with other tasks.
Asynchronous clients get their results later, but synchronous clients get their effect immediately when the service is completed. The ability to enable loosely linked systems requires asynchronous capabilities.

What is On Demand Computing?

What does On Demand Computing mean?

On demand computing (or utility computing) means accessing computing resources as you need them. Businesses pay only for what they use without any long-term commitments, so they don't need to invest heavily in tech infrastructure. This business computing model empowers companies to be more adaptive and responsive to changing conditions.

To better answer the question of 'What is on demand cloud computing?', let's explore exactly how on demand computing works:

Cloud Platforms Provide the Foundation

On demand computing is made possible through cloud platforms run by managed service providers such as Beeks. These companies maintain large networks of secure data centres that businesses can leverage instead of building their own.

This shared infrastructure allows for greater business agility; companies can scale their usage up or down based on the enterprise's demand.

Tracking Usage

A cloud service provider will track and monitor a client's resource usage closely through meters. This allows a cloud provider to charge a user's enterprise accurately based only on what they consume i.e. server time, number of users, or storage capacity.

For example, cloud usage is typically measured in gigabytes stored per month. The amount of data uploaded, downloaded, and retained in the cloud storage system is carefully metered. So, a company will only pay for what they use — whether that is 50 gigabytes or 500 terabytes of cloud data storage space.

Self-Service Access

On demand computing enables self-service access to computing resources without needing internal IT teams to handle all the setup and management. Cloud providers offer user-friendly web dashboards and automation tools for businesses to manage their own usage.

For example, in on demand self-service cloud computing, if your business rapidly gains new customers and suddenly needs additional computing resources, you can log in and scale up within minutes. On the other hand, if your enterprise needs minimal computing resources, you can scale down as required.

Integrating with Other Apps & Systems

A major benefit of utility computing is that it can integrate easily with a company's existing software tools and platforms without complex coding. The cloud providers enable these connections through simple interfaces called APIs (application programming interfaces).

For example, if a company website gets sudden traffic spikes, extra computing power from the cloud automatically helps meet peak requirements.

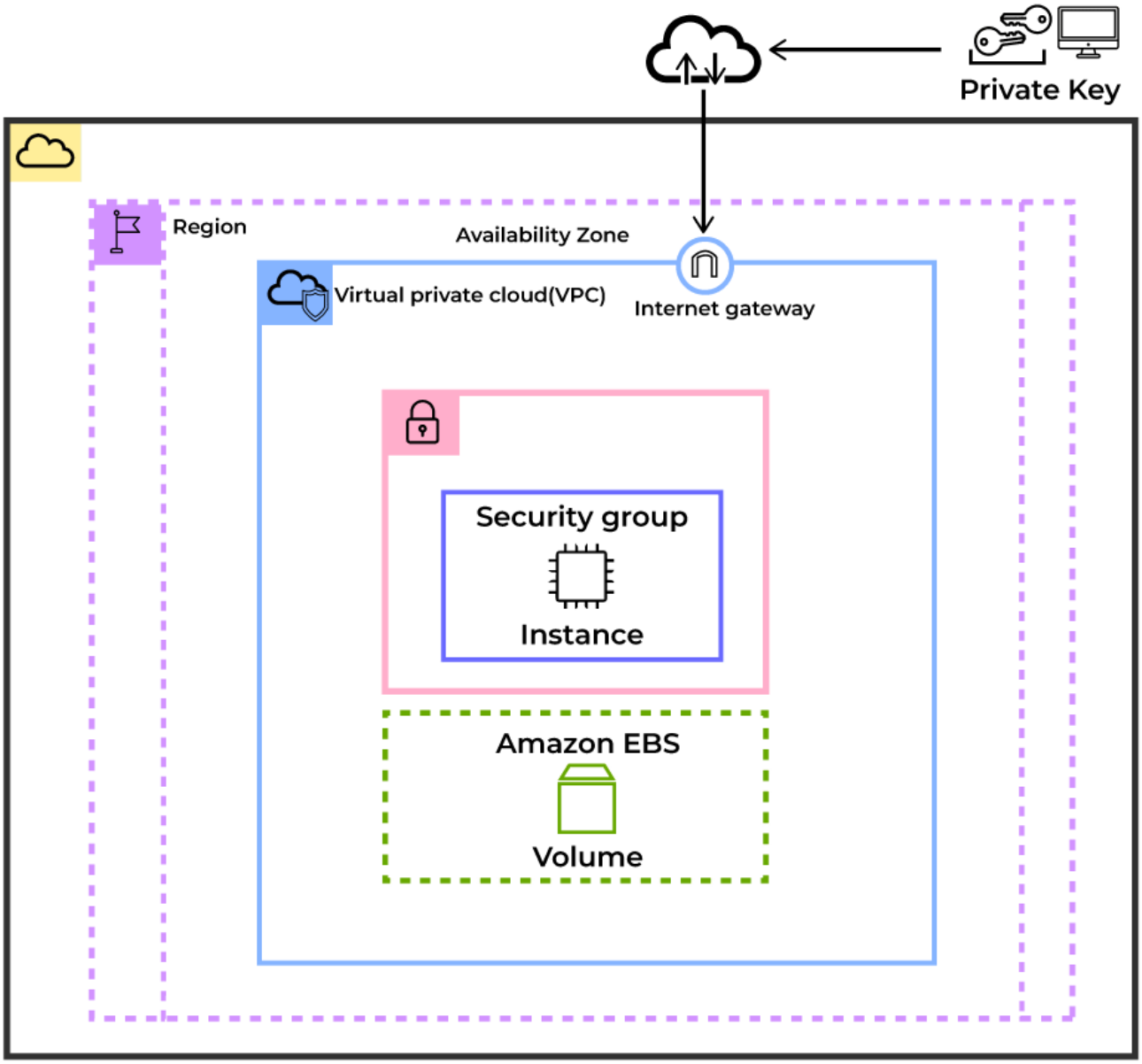
Cloud Services Development Services and Tools –

What is Amazon EC2 (Elastic Compute Cloud)?

Amazon Web service offers EC2 which is a short form of Elastic Compute Cloud (ECC) it is a cloud computing service offered by the Cloud Service Provider AWS. You can deploy your applications in EC2 servers without any worrying about the underlying infrastructure. You configure the EC2-Instance in a very secure manner by using the VPC, Subnets, and Security groups. You can scale the configuration of the EC2

instance you have configured based on the demand of the application by attaching the autoscaling group to the EC2 instance. You can scale up and scale down the instance based on the incoming traffic of the application.

The following figure shows the EC2-Instance which is deployed in VPC (Virtual Private Cloud).



Use Cases of Amazon EC2 (Elastic Compute Cloud)

The following are the use cases of Amazon EC2:

1. **Deploying Application:** In the AWS EC2 instance, you can deploy your application like **.jar,.war,** or **.ear** application without maintaining the underlying infrastructure.

2. **Scaling Application:** Once you deployed your web application in the EC2 instance know you can scale your application based upon the demand you are having by scaling the AWS EC2-Instance.
3. **Deploying The ML Models:** You can train and deploy your ML models in the EC2-instance because it offers up to 400 Gbps), and storage services purpose-built to optimize the price performance for ML projects.
4. **Hybrid Cloud Environment:** You can deploy your web application in EC2-Instance and you can connect to the database which is deployed in the on-premises servers.
5. **Cost-Effective:** Amazon EC2-instance is cost-effective so you can deploy your gaming application in the Amazon EC2-Instances

What is Google App Engine (GAE)?

Pre-requisite:- **Google Cloud Platform**

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.** For an in-depth understanding of how GAE fits into DevOps workflows, the **DevOps Engineering – Planning to Production** course offers step-by-step instructions on using GAE in production environments

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.

What is the IBM Cloud platform?

The IBM Cloud® platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud® spins up fast and performs reliably in a tested and supported environment you can trust!

IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads. Available in data centers worldwide, with multizone regions in North and South America, Europe, Asia, and Australia, you are enabled to deploy locally with global scalability.

IBM Cloud offers the most open and secure public cloud for business with a next-generation hybrid cloud platform, advanced data and AI capabilities, and deep enterprise expertise across 20 industries. Solutions are available depending on your needs for working in the public cloud, on-premises, or a combination:

- With public cloud, the resources are made available to you over the public internet. It is a multi-tenant environment, and resources like hardware and infrastructure are managed by IBM®.
- A hybrid cloud solution is a combination of public and private giving you the flexibility to move workloads between the two based on your business and technological needs. IBM uses Red Hat OpenShift on IBM Cloud, the market-leading hybrid cloud container platform for hybrid solutions that enables you to build once and deploy anywhere. With IBM Cloud Satellite, you can create a hybrid environment that brings the scalability and on-demand flexibility of public cloud services to the applications and data that runs in your secure private cloud.
- Support for multicloud and hybrid multicloud solutions is also available, which makes it easy for you to work with different vendors. IBM Cloud Paks are software products for hybrid clouds that enable you to develop apps once and deploy them anywhere.
- Virtual Private Cloud (VPC) is available as a public cloud service that lets you establish your own private cloud-like computing environment on shared public cloud infrastructure. With VPC, enterprises can define and control a virtual network that is logically isolated from all other public cloud tenants, creating a private, secure place on the public cloud.

With our open source technologies, such as Kubernetes, Red Hat OpenShift, and a full range of compute options, including virtual machines, containers, bare metal, and serverless, you have the control and flexibility that's required to support workloads in your hybrid environment. You can deploy cloud-native apps while also ensuring workload portability.

Whether you need to migrate apps to the cloud, modernize your existing apps by using cloud services, ensure data resiliency against regional failure, or use new paradigms and deployment topologies to innovate and build your cloud-native apps, the platform's open architecture is built to accommodate your use case.

What's built into the platform?

As the following diagram illustrates, the IBM Cloud platform is composed of multiple components that work together to provide a consistent and dependable cloud experience.

- A robust console that serves as the front end for creating, viewing, managing your cloud resources
- An identity and access management component that securely authenticates users for both platform services and controls access to resources consistently across IBM Cloud
- A catalog that consists of hundreds of supported products
- A search and tagging mechanism for filtering and identifying your resources
- An account and billing management system that provides exact usage for pricing plans and secure credit card fraud protection

Figure 1. Components of the IBM Cloud platform

Whether you have existing code that you want to modernize and bring to the cloud or you're developing a brand new application, your developers can tap into the rapidly growing ecosystem of available services and runtime frameworks in IBM Cloud.