Cloud Platforms in Industry

Unit -4

1. What is AWS?

Amazon web services Amazon Web Services (AWS) is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, and data storage. The platform is accessible through SOAP or RESTful Web service interfaces and provides a Web-based console where users can handle administration and monitoring of the resources required, as well as their expenses computed on a pay-as-you-go basis.

2. Name any two types of services provided by AWS

- Amazon Elastic Compute (EC2) service
- Amazon Simple Storage Service (S3)
- Communication services
- Amazon CloudWatch and
- Amazon Flexible Payment Service (FPS).

3. What is Amazon Machine Image(AMI)?

Amazon Machine Images (AMIs) are templates from which it is possible to create a virtual machine. They are stored in Amazon S3 and identified by a unique identifier in the form of ami-xxxxxx and a manifest XML file.

4. Name the major categories of EC2 instances.

- Standard instances.
- Micro instances
- High-memory instances
- High-CPU instances.
- Cluster Compute instances.
- Cluster GPU instances

5. What is a bucket? What type of storage does it provide?

A bucket is a container of objects. It can be thought of as a virtual drive hosted on the S3 distributed storage, which provides users with a flat store to which they can add objects.

Buckets are top level elements of the S3 storage architecture and do not support nesting. That is, it is not possible to create "subbuckets" or other kinds of physical divisions.

6. What are buckets and objects?

A bucket is a container of objects. It can be thought of as a virtual drive hosted on the S3 distributed storage, which provides users with a flat store to which they can add objects.

Objects constitute the content elements stored in S3. Users either store files or push to the S3 text stream representing the object's content. An object is identified by a name that needs to be unique within the bucket in which the content is stored.

7. What is AppEngine?

Google AppEngine is a PaaS implementation that provides services for developing and hosting scalable Web applications.

8. Mention the core components of AppEngine.

AppEngine is a platform for developing scalable applications accessible through the Web. The platform is logically divided into four major components:

- **infrastructure,**
- **the runtime environment,**
- **the underlying storage, and**
- **the set of scalable services that can be used to develop applications.**

9. What is Windows Azure?

Microsoft Windows Azure is a cloud operating system built on top of Microsoft datacenters' infrastructure and provides developers with a collection of services for building applications with cloud technology. Services range from compute, storage, and networking to application connectivity, access control, and business intelligence.

10. What is AppFabric and list the services it provides?

Fabric is a comprehensive middleware for developing, deploying, and managing applications on the cloud or for integrating existing applications with cloud services.

Services : communication, authentication and authorization, and data access.

11. What is a role? Mention the types .

A role is a runtime environment that is customized for a specific compute task. Roles are managed by the Azure operating system and instantiated on demand in order to address surges in application demand.

Currently, there are three different roles: **Web role, Worker role, and Virtual Machine (VM) role**.

12. Expand AMI and EBS.

Amazon Machine Images (AMIs)

Elastic Block Store (EBS)

13. What are CRM and ERP applications in Cloud Computing?

CRM stands for Customer Relationship Management and is a software that is hosted in cloud so that the users can access the information using internet. CRM software provides high level of security and scalability to its users and can be easily used on mobile phones to access the data. Some of the major CRM vendors include Oracle Siebel, Mothernode CRM, Microsoft Dynamics CRM, Infor CRM, SAGE CRM, NetSuite CRM.

(Now a days, many business vendors and service providers are using these CRM software to manage the resources so that the user can access them via internet. Moving the business computation from desktop to the cloud is proving a beneficial step in both the IT and Non-IT fields. Some of the major CRM vendors include Oracle Siebel, Mothernode CRM, Microsoft Dynamics CRM, Infor CRM, SAGE CRM, NetSuite CRM.)

ERP is an abbreviation for Enterprise Resource Planning and is a software similar to CRM that is hosted on cloud servers which helps the enterprises to manage and manipulate their business data as per their needs and user requirements. ERP software follows pay per use methodologies of payment.

There are various ERP vendors available like Oracle, SAP, Epicor, SAGE, Microsoft Dynamics, Lawson Softwares.

14. What are DropBox and iCloud?

most popular solution for online document storage is Dropbox, an online application that allows users to synchronize any file across any platform and any device in a seamless manner. Dropbox provides users with a free amount of storage that is accessible through the abstraction of a folder. Users can either access their Dropbox folder through a browser or by downloading and installing a Dropbox client, which provides access to the online storage by means of a special folder.

15. Write any 4 scientific applications of cloud computing.

Healthcare: ECG analysis in the cloud

Biology: protein structure prediction

Biology: gene expression data analysis for cancer diagnosis

Geoscience: satellite image processing

Business and consumer applications

16. What are the core functionalities provided by Google Docs?

Google Docs is a SaaS application that delivers the basic office automation capabilities with support for collaborative editing over the Web.

Google Docs allows users to create and edit text documents, spreadsheets, presentations, forms, and drawings. It aims to replace desktop products such as Microsoft Office and OpenOffice and provide similar interface and functionality as a cloud service.

17. What do you mean by Cloud Desktops?

Cloud desktops allow businesses to store users' desktop environments (the icons, wallpaper, windows, folders, toolbars, widgets, etc. users see every day when they go to work and logon to their computers) on a server, rather than on a local PC. Users can access their desktop remotely via laptops, smartphones, or tablets over a network that makes it feel as if they are working off the desktop itself.

18. What is Animoto?

Animoto is a company that creates videos out of images, music, and video fragments submitted by users. The process involves a considerable amount of storage and backend processing required for producing the video, which is finally made available to the user.

19. What is SalesForce.com?

SalesForce.com is a cloud computing platform for developing social enterprise applications. The platform is the basis for SalesForce.com, a Software-as-a-Service solution for customer relationship management. Force.com allows developers to create applications by composing ready-to-use blocks; a complete set of components supporting all the activities of an enterprise are available.

20. Expand CRM and ERP

CRM stands for Customer Relationship Management

ERP is an abbreviation for Enterprise Resource Planning

2) What is EC2 Instance? Explain different categories of EC2.

Amazon Elastic Compute (EC2)

EC2 instances represent virtual machines. They are created using AMI(Amazon Machine Images) as templates, which are specialized by selecting the number of cores, their computing power, and the installed memory. The processing power is expressed in terms of virtual cores and EC2 Compute Units (ECUs).

EC2 instances can identify six major categories:

• **Standard instances.** This class offers a set of configurations that are suitable for most applications. EC2 provides three different categories of increasing computing power, storage, and memory.

• **Micro instances.** This class is suitable for those applications that consume a limited amount of computing power and memory and occasionally need bursts in CPU cycles to process surges(a sudden large increase) in the workload. Micro instances can be used for small Web applications with limited traffic.

• **High-memory instances.** This class targets applications that need to process huge workloads and require large amounts of memory. Three-tier Web applications characterized by high traffic are the target profile. Three categories of increasing memory and CPU are available, with memory proportionally larger than computing power.

• **High-CPU instances.** This class targets compute-intensive applications. Two configurations are available where computing power proportionally increases more than memory.

• **Cluster Compute instances.** This class is used to provide virtual cluster services. Instances in this category are characterized by high CPU compute power and large memory and an extremely high I/O and network performance, which makes it suitable for HPC applications.

• **Cluster GPU instances.** This class provides instances featuring graphic processing units (GPUs) and high compute power, large memory, and extremely high I/O and network performance. This class is particularly suited for cluster applications that perform heavy graphic computations, such as rendering clusters. Since GPU can be used for general-purpose computing, users of such instances can benefit from additional computing power, which makes this class suitable for HPC applications.

3) Explain any two services of Amazon web service.

1. Communication services Amazon provides facilities to structure and facilitate the communication among existing applications and services residing within the AWS infrastructure. These facilities can be organized into two major categories: **virtual networking and messaging**.

Virtual networking: Virtual networking comprises a collection of services that allow AWS users to control the connectivity to and between compute and storage services. Amazon Virtual Private Cloud (VPC) an Amazon Direct Connect provide connectivity solutions in terms of i**nfrastructure**; Route 53 facilitates connectivity in terms of **naming**.

Amazon Direct Connect allows AWS users to create dedicated networks between the user private network and Amazon Direct Connect locations, called ports. This connection can be further partitioned in multiple logical connections and give access to the public resources hosted on the Amazon infrastructure.

Amazon Route 53 implements dynamic DNS services that allow AWS resources to be reached through domain names different from the amazon.com domain

Messaging: Messaging services constitute the next step in connecting applications by leveraging AWS capabilities. The three different types of messaging services offered are

- Amazon Simple Queue Service (SQS),
- Amazon Simple Notification Service (SNS), and
- Amazon Simple Email Service (SES).

***Amazon SQS constitutes disconnected model for exchanging messages between applications by means of message queues, hosted within the AWS infrastructur*e**. Using the AWS console users can

create an unlimited number of message queues and configure them to control their access. Applications can send messages to any queue they have access to. These messages are securely and redundantly stored within the AWS infrastructure for a limited period of time, and they can be accessed by other (authorized) applications. While a message is being read, it is kept locked to avoid spurious processing from other applications. Such a lock will expire after a given period.

Amazon SNS provides a publish-subscribe method for connecting heterogeneous applications. Amazon SNS allows applications to be notified when new content of interest is available. This feature is accessible through a Web service whereby AWS users can create a topic, which other applications can subscribe to. At any time, applications can publish content on a given topic and subscribers can be automatically notified. The service provides subscribers with different notification models (HTTP/HTTPS, email/email JSON, and SQS).

Amazon SES provides ***AWS users with a scalable email service that leverages the AWS infrastructure***. Once users are signed up for the service, they have to provide an email that SES will use to send emails on their behalf. To activate the service, SES will send an email to verify the given address and provide the users with the necessary information for the activation. Upon verification, the user is given an SES sandbox to test the service, and he can request access to the production version. Using SES, it is possible to send either SMTP-compliant emails or raw emails by specifying email headers and Multipurpose Internet Mail Extension (MIME) type Emails are queued for delivery, and the users are notified of any failed delivery.

2. Storage Service

AWS provides a collection of services for data storage and information management. The core service in this area is represented by **Amazon Simple Storage Service (S3)**. This is a distributed object store that allows users to store information in different formats. The core components of S3 are two: buckets and objects. Buckets represent virtual containers in which to store objects; objects represent the content that is actually stored.

S3 key concepts

As the name suggests, S3 has been designed to provide a simple storage service that's accessible through a **Representational State Transfer (REST)** interface, which is quite similar to a distributed file system but which presents some important differences that allow the infrastructure to be highly efficient:

• **The storage is organized in a two-level hierarchy.** S3 organizes its storage space into buckets that cannot be further partitioned. This means that it is not possible to create directories or other kinds of physical groupings for objects stored in a bucket. Despite this fact, there are

few limitations in naming objects, and this allows users to simulate directories and create logical groupings.

• **Stored objects cannot be manipulated like standard files**. S3 has been designed to essentially provide storage for objects that will not change over time. Therefore, it does not allow renaming, modifying, or relocating an object. Once an object has been added to a bucket, its content and position is immutable, and the only way to change it is to remove the object from the store and add it again.

• **Content is not immediately available to users.** The main design goal of S3 is to provide an eventually consistent data store. As a result, because it is a large distributed storage facility, changes are not immediately reflected. For instance, S3 uses replication to provide redundancy and efficiently serve objects across the globe; this practice introduces latencies when adding objects to the store—especially large ones—which are not available instantly across the entire globe.

• **Requests will occasionally fail.** Due to the large distributed infrastructure being managed, requests for object may occasionally fail. Under certain conditions, S3 can decide to drop a request by returning an internal server error. Therefore, it is expected to have a small failure rate during day-to-day operations, which is generally not identified as a persistent failure.

3. Compute Service

Compute services constitute the fundamental element of cloud computing systems. The fundamental service in this space is Amazon EC2, which delivers an IaaS solution. Amazon EC2 allows deploying servers in the form of virtual machines created as instances of a specific image. Images come with a preinstalled operating system and a software stack, and instances can be configured for memory, number of processors, and storage. Users are provided with credentials to remotely access the instance and further configure or install software if needed.

Amazon machine images: Amazon Machine Images (AMIs) are templates from which it is possible to create a virtual machine. They are stored in Amazon S3 and identified by a unique identifier in the form of ami-xxxxxx and a manifest XML file.

Amazon Elastic Compute (EC2)

EC2 instances represent virtual machines. They are created using AMI(Amazon Machine Images) as templates, which are specialized by selecting the number of cores, their computing power, and the installed memory. The processing power is expressed in terms of virtual cores and EC2 Compute Units (ECUs).

EC2 instances can identify six major categories:

• **Standard instances.** This class offers a set of configurations that are suitable for most applications. EC2 provides three different categories of increasing computing power, storage, and memory.

• **Micro instances.** This class is suitable for those applications that consume a limited amount of computing power and memory and occasionally need bursts in CPU cycles to process surges(a sudden large increase) in the workload. Micro instances can be used for small Web applications with limited traffic.

• **High-memory instances.** This class targets applications that need to process huge workloads and require large amounts of memory. Three-tier Web applications characterized by high traffic are the target profile. Three categories of increasing memory and CPU are available, with memory proportionally larger than computing power.

• **High-CPU instances.** This class targets compute-intensive applications. Two configurations are available where computing power proportionally increases more than memory.

• **Cluster Compute instances.** This class is used to provide virtual cluster services. Instances in this category are characterized by high CPU compute power and large memory and an extremely high I/O and network performance, which makes it suitable for HPC applications.

• **Cluster GPU instances.** This class provides instances featuring graphic processing units (GPUs) and high compute power, large memory, and extremely high I/O and network performance. This class is particularly suited for cluster applications that perform heavy graphic computations, such as rendering clusters. Since GPU can be used for general-purpose computing, users of such instances can benefit from additional computing power, which makes this class suitable for HPC applications.

4) What is Storage Services? Explain S3 Key concepts in Storage service

AWS provides a collection of services for data storage and information management. The core service in this area is represented by **Amazon Simple Storage Service (S3)**. This is a distributed object store that allows users to store information in different formats. The core

components of S3 are two: buckets and objects. Buckets represent virtual containers in which to store objects; objects represent the content that is actually stored.

S3 key concepts

As the name suggests, S3 has been designed to provide a simple storage service that's accessible through a **Representational State Transfer (REST)** interface, which is quite similar to a distributed file system but which presents some important differences that allow the infrastructure to be highly efficient:

• **The storage is organized in a two-level hierarchy.** S3 organizes its storage space into buckets that cannot be further partitioned. This means that it is not possible to create directories or other kinds of physical groupings for objects stored in a bucket. Despite this fact, there are few limitations in naming objects, and this allows users to simulate directories and create logical groupings.

• **Stored objects cannot be manipulated like standard files**. S3 has been designed to essentially provide storage for objects that will not change over time. Therefore, it does not allow renaming, modifying, or relocating an object. Once an object has been added to a bucket, its content and position is immutable, and the only way to change it is to remove the object from the store and add it again.

• **Content is not immediately available to users.** The main design goal of S3 is to provide an eventually consistent data store. As a result, because it is a large distributed storage facility, changes are not immediately reflected. For instance, S3 uses replication to provide redundancy and efficiently serve objects across the globe; this practice introduces latencies when adding objects to the store—especially large ones—which are not available instantly across the entire globe.

• **Requests will occasionally fail.** Due to the large distributed infrastructure being managed, requests for object may occasionally fail. Under certain conditions, S3 can decide to drop a request by returning an internal server error. Therefore, it is expected to have a small failure rate during day-to-day operations, which is generally not identified as a persistent failure.

5) Explain any two services of Google AppEngine.

1. Storage service (refer Q. No. 6)

2. Application service(refer Q. No. 7)

3. Compute service : Web applications are mostly designed to interface applications with users by means very common channel, that is, the Web. Most of the interaction is performed synchronously: Users navigate the Web pages and get instantaneous feedback in response to their actions. This feedback is often the result of some computation happening on the Web application, which implements the intended logic to serve the user request. Sometimes this approach is not applicable—for example, in long computations or when some operations need to be triggered at a given point in time. A good design for these scenarios provides the user with immediate feedback and a notification once the required operation is completed. AppEngine offers a**dditional services such as Task Queues and Cron Jobs** that simplify the execution of computations and that cannot be performed within the timeframe of the Web request.

Task queues: **Task Queues allow applications to submit a task for a later execution**. This service is particularly useful for long computations that cannot be completed within the maximum response time of a request handler. The service allows users to have up to 10 queues that can execute tasks at a configurable rate. (In fact, a task is defined by a Web request to a given URL, and the queue invokes the request handler by passing Web request to the handler. It is the responsibility of the request handler to perform the "task execution," which is seen from the queue as a simple Web request. The queue is designed to reexecute the task in case of failure in order to avoid transient failures preventing the task from a successful completion.)

Cron jobs Sometimes the length of computation might not be the primary reason that an operation is not performed within the scope of the Web request. It might be possible that the required operation needs to be performed at a specific time of the day, which does not coincide with the time of the Web request. In this case, it is possible to schedule the required operation at the desired time by using the Cron Jobs service. This service operates similarly to Task Queues but invokes the request handler specified in the task at a given time and does not reexecute the task in case of failure. This behavior can be useful to implement maintenance operations or send periodic notifications.

6) Explain about storage service in Google AppEngine.

AppEngine provides various types of storage, which operate differently depending on the volatility of the data. There are three different levels of storage:

- in memory-cache,
- storage for semi-structured data, and
- long-term storage for static data.

Static file servers : Web applications are composed of dynamic and static data. Dynamic data are a result of the logic of the application and the interaction with the user. Static data often are mostly

constituted of the components that define the graphical layout of the application (CSS files, plain HTML files, JavaScript files, images, icons, and sound files) or data files. These files can be hosted on static file servers, since they are not frequently modified. Such servers are optimized for serving static content, and users can specify how dynamic content should be served when uploading their applications to AppEngine.

Data Store: Data Store is a service that allows developers to store semi structured data. The service is designed to scale and optimized to quickly access data. Data Store can be considered as a large object database in which to store objects that can be retrieved by a specified key. Both the type of the key and the structure of the object can vary.

With respect to the traditional Web applications backed by a relational database, DataStore imposes less constraint on the regularity of the data. These design decisions originated from a careful analysis of data usage patterns for Web applications and were taken in order to obtain a more scalable and efficient data store. The underlying infrastructure of Data Store is based on Bigtable, a redundant, distributed, and semi-structured data store that organizes data in the form of tables

Data Store provides high-level abstractions that simplify interaction with Bigtable. Developers define their data in terms of entity and properties, and these are persisted and maintained by the service into tables in Bigtable.

Data Store also provides facilities for creating indexes on data and to update data within the context of a transaction. Indexes are used to support and speed up queries. A query can return zero or more objects of the same kind or simply the corresponding keys.

7) Explain different application services in Google AppEngine.

Applications hosted on AppEngine take the most from the services made available through the runtime environment. These services simplify most of the common operations that are performed in Web applications:

- access to data,
- account management,
- integration of external resources,
- messaging and communication,
- image manipulation, and asynchronous computation.

UrlFetch:

Web 2.0 has introduced the concept of composite Web applications. Different resources are put together and organized as meshes within a single Web page. Meshes are fragments of HTML generated in different ways. They can be directly obtained from a remote server or rendered from an XML document retrieved from a Web service, or they can be rendered by the browser as the result of an embedded and remote component.

The sandbox environment does not allow applications to open arbitrary connections through sockets, but it does provide developers with the capability of retrieving a remote resource through

HTTP/HTTPS by means of the UrlFetch service. Applications can make synchronous and asynchronous Web requests and integrate the resources obtained. One of the interesting features of UrlFetch is the ability to set deadlines for requests so that they can be completed (or aborted) within a given time. Moreover, the ability to perform such requests asynchronously allows the applications to continue with their logic while the resource is retrieved in the background.

MemCache:

AppEngine provides developers with access to fast and reliable storage, which is DataStore. Despite this, the main objective of the service is to serve as a scalable and long-term storage, where data are persisted to disk redundantly in order to ensure reliability and availability of data against failures. This design poses a limit on how much faster the store can be compared to other solutions, especially for objects that are frequently accessed—for example, at each Web request. AppEngine provides caching services by means of MemCache. This is a distributed in-memory cache that is optimized for fast access and provides developers with a volatile store for the objects that are frequently accessed. The caching algorithm implemented by MemCache will automatically remove the objects that are rarely accessed. The use of MemCache can significantly reduce the access time to data; developers can structure their applications so that each object is first looked up into MemCache and if there is a miss, it will be retrieved from DataStore and put into the cache for future lookups

Mail and instant messaging:

Communication is another important aspect of Web applications. It is common to use email for following up with users about operations performed by the application. To facilitate the implementation of such tasks, AppEngine provides developers with the ability to send and receive mails through Mail. The service allows sending email on behalf of the application to specific user accounts. It is also possible to include several types of attachments and to target multiple recipients. Mail operates asynchronously, and in case of failed delivery the sending address is notified through an email detailing the error.

AppEngine provides also another way to communicate with the external world: the Extensible Messaging and Presence Protocol (XMPP). Any chat service that supports XMPP, such as Google Talk, can send and receive chat messages to and from the Web application, which is identified by its own address. Even though the chat is a communication medium mostly used for human interactions, XMPP can be conveniently used to connect the Web application with chat bots.

Account management:

Web applications often keep various data that customize their interaction with users. These data normally go under the user profile and are attached to an account. AppEngine simplifies account management by allowing developers to leverage Google account management by means of Google

Accounts. The integration with the service also allows Web applications to offload the implementation of authentication capabilities to Google's authentication system.

Using Google Accounts, Web applications can conveniently store profile settings in the form of key-value pairs, attach them to a given Google account, and quickly retrieve them once the user authenticates.

Image manipulation:

Web applications render pages with graphics. Often simple operations, such as adding watermarks or applying simple filters, are required. *AppEngine allows applications to perform image resizing, rotation, mirroring, and enhancement by means of Image Manipulation*, a service that is also used in other Google products.

(

- **Synchronous request** — (Default) Where the client blocks and waits for the result of the remote request before continuing execution
- **Asynchronous request** — Where the client continues execution after initiating the request and processes the result whenever the AppServer makes it available
- A **sandbox** is an isolated testing environment that enables users to run programs or open files without affecting the application, system or platform on which they run.

)

8) Write a note on Application life cycle in Google AppEngine.

AppEngine provides support for almost all the phases characterizing the life cycle of an application: **testing and development, deployment, and monitoring**. The SDKs released by Google provide developers with most of the functionalities required by these tasks. Currently ***there are two SDKs*** **available for development**: **Java SDK and Python SDK.**

Application development and testing : Developers can start building their Web applications on a local development server. This is a self-contained environment that helps developers tune applications without uploading them to AppEngine.

Java SDK :The Java SDK provides developers with the facility for building applications with the Java 5 and Java 6 runtime environments. Alternatively, it is possible to develop applications within the Eclipse development environment by using the Google AppEngine plug-in, which integrates the features of the SDK within the powerful Eclipse environment. Using the Eclipse software installer, it is possible to download and install Java SDK, Google Web Toolkit, and Google AppEngine plug-ins into Eclipse. These three components allow developers to program powerful and rich Java applications for AppEngine.

The SDK supports the development of applications by using the servlet abstraction, which is a common development model. Together with servlets, many other features are available to build applications. Moreover, developers can easily create Web applications by using the Eclipse Web Platform, which provides a set of tools and components.

The plug-in allows developing, testing, and deploying applications on AppEngine. Other tasks, such as retrieving the log of applications, are available by means of command-line tools that are part of the SDK.

Python SDK : The Python SDK allows developing Web applications for AppEngine with Python 2.5. It provides a standalone tool, called **GoogleAppEngineLauncher**, for managing Web applications locally and deploying them to AppEngine. The tool provides a convenient user interface that lists all the available Web applications, controls their execution, and integrates them with the default code editor for editing application files. In addition, the launcher provides access to some important services for application monitoring and analysis, such as the logs, the SDK console, and the dashboard. The log console captures all the information that is logged by the application while it is running. The console SDK provides developers with a Web interface via which they can see the application profile in terms of utilized resource. This feature is particularly useful because it allows developers to preview the behaviour of the applications once they are deployed on AppEngine, and it can be used to tune applications made available through the runtime.

The Python implementation of the **SDK also comes with an integrated Web application framework called webapp t**hat includes a set of models, components, and tools that simplify the development of Web applications. This is not the only Web framework that can be used to develop Web applications. **Another Web framework that is known to work well is Django.**

Application deployment and management Once the application has been developed and tested, it can be deployed on AppEngine with a simple click or command-line tool. Before performing such task, it is necessary to create an application identifier, which will be used to locate the application from the Web browser by typing the address http://<application-id>.appspot.com

An application identifier is mandatory because it allows unique identification of the application while it's interacting with AppEngine. Developers use an **app identifier to upload and update** applications. It is possible to register an application identifier by logging into AppEngine and selecting the "Create application" option. It is also possible to provide an application title that is descriptive of the application; the title can be changed over time.

Once an application identifier has been created, it is possible to deploy an application on AppEngine. This task can be done using either the respective development environment (GoogleAppEngineLauncher and Google AppEngine plug-in) or the command-line tools. Once the application is uploaded, nothing else needs to be done to make it available. AppEngine will take care of everything. Developers can then manage the application by using the administrative console. This is the primary tool used for application monitoring and provides users with insight into resource usage (CPU, bandwidth) and services and other useful counters.

9) Explain different roles in Compute service of Microsoft Azure.

Compute services are the core components of Microsoft Windows Azure, and they are delivered by means of the abstraction of roles. A role is a runtime environment that is customized for a specific compute task. Roles are managed by the Azure operating system and instantiated on demand in order to address surges in application demand. Currently, there are three different roles: **Web role, Worker role, and Virtual Machine (VM) role**

1. **Web role** The Web role is designed to implement scalable Web applications. Web roles represent the units of deployment of Web applications within the Azure infrastructure. They are hosted on the IIS 7 Web Server, which is a component of the infrastructure that supports Azure. When Azure detects peak loads in the request made to a given application, it instantiates multiple Web roles for that application and distributes the load among them by means of a load balancer.

   Since version 3.5, the .NET technology natively supports Web roles; developers can directly develop their applications in Visual Studio, test them locally, and upload to Azure. It is possible to develop ASP.NET. Since IIS 7 also supports the PHP runtime environment by means of the FastCGI module, Web roles can be used to run and scale PHP Web applications on Azure (CGI Web Role).

2. **Worker role** Worker roles are designed to host general compute services on Azure. They can be used to quickly provide compute power or to host services that do not communicate with the external world through HTTP. A common practice for Worker roles is to use them to provide background processing for Web applications developed with Web roles.

   Developing a worker role is like a developing a service. Compared to a Web role whose computation is triggered by the interaction with an HTTP client (i.e., a browser), a Worker role runs continuously from the creation of its instance until it is shut down. The Azure SDK provides developers with convenient APIs and libraries that allow connecting the role with the service provided by the runtime and easily controlling its startup as well as being notified of changes in the hosting environment. As with Web roles, the .NET technology provides complete support for Worker roles.

3. **Virtual machine role**: The Virtual Machine role allows developers to fully control the computing stack of their compute service by defining a custom image of the Windows Server 2008 R2 operating system and all the service stack required by their applications. The Virtual Machine role is based on the Windows Hyper-V virtualization technology, which is natively integrated in the Windows server technology at the base of Azure. Developers can image a Windows server installation complete with all the required applications and components, save it into a Virtual Hard Disk (VHD) file, and upload it to Windows Azure to create compute instances on demand. Different types of instances are available, and Table 9.7 provides an overview of the options offered during 2011-2012.

Table 9.7 Windows Azure Compute Instances Characteristics, 2011–2012

Compute Instance Type	CPU	Memory	Instance Storage	I/O Performance	Hourly Cost (USD)
Extra Small	1.0 GHz	768 MB	20 GB	Low	$0.04
Small	1.6 GHz	1.75 GB	225 GB	Moderate	$0.12
Medium	2 × 1.6 GHz	3.5 GB	490 GB	High	$0.24
Large	4 × 1.6 GHz	7 GB	1,000 GB	High	$0.48
Extra Large	8 × 1.6 GHz	14 GB	2,040 GB	High	$0.96

Compared to the Worker and Web roles, the VM role provides finer control of the compute service and resource that are deployed on the Azure Cloud. An additional administrative effort is required for configuration, installation, and management of services.

10) Explain different storage services of Microsoft Azure.

Compute resources are equipped with local storage in the form of a directory on the local file system that can be used to temporarily store information that is useful for the current execution cycle of a role. If the role is restarted and activated on a different physical machine, this information is lost. Windows Azure provides different types of storage solutions that complement compute services with a more durable and redundant option compared to local storage. Compared to local storage, these services can be accessed by multiple clients at the same time and from everywhere, thus becoming a general solution for storage.

1. **Blobs**: Azure allows storing large amount of data in the form of **binary large objects (BLOBs)** by means of the blobs service. This service is optimal to store large text or binary files. Two types of blobs are available:

• **Block blobs.** Block blobs are **composed of blocks** and **are optimized for sequential access**; therefore they are appropriate for media streaming. Currently, blocks are of 4 MB, and a single block blob can reach 200 GB in dimension.

• **Page blobs**. Page blobs are **made of pages** that are identified by an offset from the beginning of the blob. A page blob can be split into multiple pages or constituted of a single page. This type of blob is **optimized for random access** and can be used to host data different from streaming. Currently, the maximum dimension of a page blob can be 1 TB

2. **Azure drive** : Page blobs can be used to store an entire file system in the form of a single Virtual Hard Drive (VHD) file. This can then be mounted as a part of the NTFS file system by Azure compute resources, thus providing persistent and durable storage. **A page blob mounted as part of an NTFS tree is called an Azure Drive.**
3. **Tables** : Tables constitute a semi-structured storage solution, allowing users to store information in the form of entities with a collection of properties. Entities are stored as rows in the table and are identified by a key, which also constitutes the unique index built for the table. Users can insert, update, delete, and select a subset of the rows stored in the table. Unlike SQL tables, there are no schema enforcing constraints on the properties of entities and there is no facility for representing relationships among entities. For this reason, tables are more similar to spreadsheets rather than SQL tables.
4. **Queues:** Queue storage allows applications to communicate by exchanging messages through durable queues, thus avoiding lost or unprocessed messages. Applications enter messages into a queue, and other applications can read them in a first-in, first-out (FIFO) style.
   To ensure that messages get processed, when an application reads a message it is marked as invisible; hence it will not be available to other clients. Once the application has completed processing the message, it needs to explicitly delete the message from the queue. This two-phase process ensures that messages get processed before they are removed from the queue.

11) Write a note on SQL Azure.

SQL Azure is a relational database service hosted on Windows Azure and built on the SQL Server technologies. The service extends the capabilities of SQL Server to the cloud and provides developers with a scalable, highly available, and fault-tolerant relational database. SQL Azure is accessible from either the Windows Azure Cloud or any other location that has access to the Azure Cloud. It is fully compatible with the interface exposed by SQL Server, so applications built for SQL Server can transparently migrate to SQL Azure. Moreover, the service is fully manageable using REST APIs,

allowing developers to control databases deployed in the Azure Cloud as well as the firewall rules set up for their accessibility.

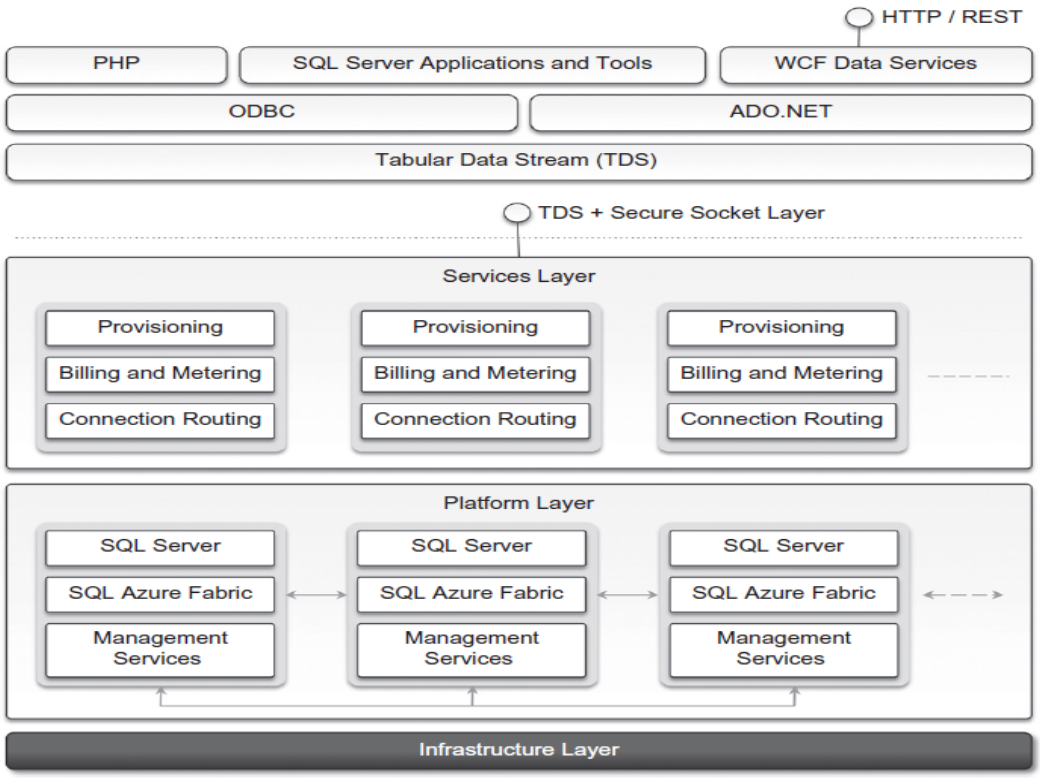


FIGURE 9.4
SQL Azure architecture.

Figure 9.4 shows the architecture of SQL Azure. Access to SQL Azure is based on the Tabular Data Stream (TDS) protocol, which is the communication protocol underlying all the different interfaces used by applications to connect to a SQL Server-based installation such as ODBC and ADO.NET. On the SQL Azure side, access to data is mediated by the service layer, which provides provisioning, billing, and connection-routing services. These services are logically part of server instances, which are managed by SQL Azure Fabric. This is the distributed database middleware that constitutes the infrastructure of SQL Azure and that is deployed on Microsoft datacenters.

Developers have to sign up for a Windows Azure account in order to use SQL Azure. Once the account is activated, they can either use the Windows Azure Management Portal or the REST APIs to create servers and logins and to configure access to servers.

Currently, the SQL Azure service is billed according to space usage and the type of edition. Currently, two different editions are available: Web Edition and Business Edition.

12) What is AppFabric? Explain different services of AppFabric.

AppFabric is a comprehensive middleware for developing, deploying, and managing applications on the cloud or for integrating existing applications with cloud services.

communication, authentication and authorization, and data access

Access control AppFabric provides the capability of encoding access control to resources in Web applications and services into a set of rules that are expressed outside the application code base. These rules give a great degree of flexibility in terms of the ability to secure components of the application and define access control policies for users and groups.

Access control services also integrate several authentication providers into a single coherent identity management framework. Applications can leverage Active Directory, Windows Live, Google, Facebook, and other services to authenticate users. This feature also allows easy building of hybrid systems, with some parts existing in the private premises and others deployed in the public cloud.

Service bus: Service Bus constitutes the **messaging and connectivity infrastructure** provided with AppFabric for building distributed and disconnected applications in the Azure Cloud and between the private premises and the Azure Cloud. Service Bus allows applications to interact with different protocols and patterns over a reliable communication channel that guarantees delivery.

Service Bus allows services to be available by simple URLs, which are untied from their deployment location. It is possible to support publish-subscribe models, full-duplex communications point to point as well as in a peer-to-peer environment, unicast and multicast message delivery in one-way communications, and asynchronous messaging to decouple application components.

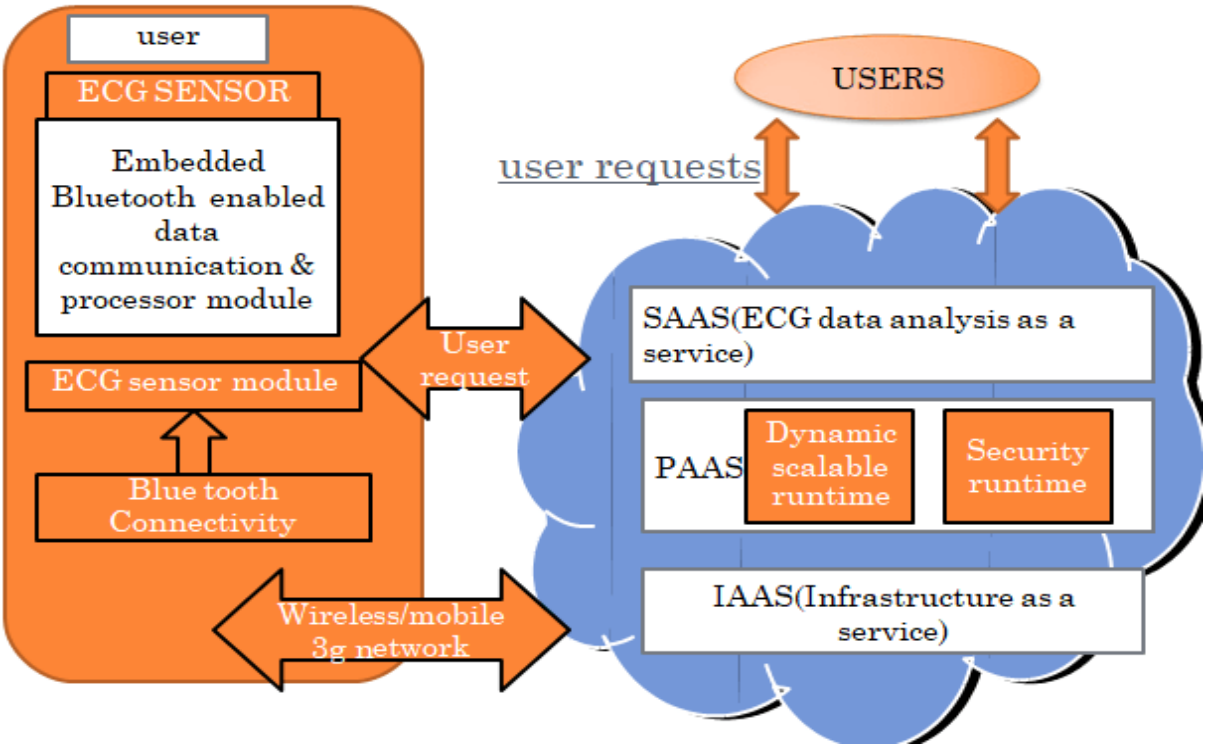
Azure cache : Windows Azure provides a set of durable storage solutions that allow applications to persist their data. These solutions are based on disk storage, which might constitute a bottleneck for the applications that need to gracefully scale along the clients' requests and dataset size dimensions.

Azure Cache is a service that allows developers to quickly access data persisted on Windows Azure storage or in SQL Azure. The service implements a distributed in-memory cache of which the size can be dynamically adjusted by applications according to their needs.

13) Describe how cloud computing technology can be applied to support remote ECG monitoring.

ECG (Electrocardiogram) analysis in the cloud refers to the utilization of cloud computing resources and services to process and analyze ECG data.

- Cloud computing technologies allows the remote monitoring of a patient's heart beat data.
- Through this way the patient at risk can be constantly monitored without going to the hospital for ECG analysis.
- At the same time the Doctor's can instantly be notified with cases that need's their attention.
- Here in this fig there are different types of computing devices equipped with ECG sensors to constantly monitor the patient's heart beat.
- The respective information is transmitted to the patient's mobile device that will immediately forwarded to the cloud- hosted web services for analysis.
- The entire web services from the front end of a platform that is completely hosted in the cloud that consist of three layers:Saas,Paas,Iaas.



Here's an overview of how ECG analysis can be performed in the cloud:

1. Data Collection and Storage:

- ECG data can be collected using wearable devices, monitoring systems, or medical equipment.
- The collected data is securely transmitted to the cloud for storage and further analysis.
- Cloud storage services provide a scalable and reliable platform to store large volumes of ECG data.

2. Data Preprocessing:

- ECG data often requires preprocessing before analysis to remove noise, artifacts, and baseline wander.
- Cloud-based preprocessing techniques can be applied to the raw ECG data using algorithms for filtering, signal enhancement, and normalization.
- Preprocessed ECG data is stored or transmitted to subsequent analysis modules.

3. Signal Processing and Analysis:

- Cloud-based signal processing algorithms can be applied to analyze ECG data for various purposes, such as arrhythmia detection, heart rate variability analysis, and ischemia detection.
- Machine learning and data mining techniques can be employed in the cloud to train models and perform automated analysis on ECG data.

4. Real-time Monitoring and Alerting:

- Cloud platforms enable real-time monitoring of ECG data streamed from wearable devices or monitoring systems.
- Cloud-based algorithms can continuously analyze the incoming ECG data to detect abnormalities or critical events.
- In case of any anomalies or predefined thresholds being crossed, the cloud system can generate alerts or notifications to healthcare providers or patients.

5. Collaboration and Integration:

- Cloud-based ECG analysis allows for seamless collaboration among healthcare professionals, researchers, and data scientists.
- Multiple users can access and analyze the same ECG data simultaneously, enabling collaborative diagnosis and research.

6. Security and Privacy:

- Cloud providers implement robust security measures to protect sensitive ECG data, including encryption, access controls, and compliance with healthcare data protection regulations.

- Compliance with standards such as HIPAA (Health Insurance Portability and Accountability Act) ensures the privacy and security of patient health information.

https://easyexamnotes.com/ecg-analysis-in-cloud/

14) Describe how protein structure prediction can be done using cloud technology.

Applications in biology often require high computing capabilities and often operate on large data sets that cause extensive I/O operations. Because of these requirements, biology applications have often made extensive use of supercomputing and cluster computing infrastructures.

Protein structure prediction is a computationally intensive task that is fundamental to different types of research in the life sciences. Among these is the design of new drugs for the treatment of diseases. Protein structure prediction is the result of complex computations aimed at identifying the structure that minimizes the required energy. This task requires the investigation of a space with a massive number of states, consequently creating a large number of computations for each of these states. The computational power required for protein structure prediction can now be acquired on demand, without owning a cluster or navigating the administration to get access to parallel and distributed computing facilities. Cloud computing grants access to such capacity on a pay-per-use basis.

- **Jeeva is a computational platform which simplifies the development of new prediction algorithms** and improves the efficiency at the same time.
- Jeeva web portal system consists of an interactive web interface and a Grid middleware.
- With the interactive web interface, users can submit prediction requests for protein secondary structures, collect results, and manage the history of prediction data.
- By means of the Grid middleware, researchers can not only deploy their prediction applications in a distributed environment easily, but also monitor and manage the execution in the distributed environment.
- The prediction task uses machine learning techniques (support vector machines) for determining the secondary structure of proteins.
- These techniques translate the problem into one of pattern recognition, where a sequence has to be classified into one of three possible classes (E, H, and C).

- A popular implementation based on support vector machines divides the pattern recognition problem into three phases: **initialization, classification, and a final phase**.

Even though these three phases have to be executed in sequence, it is possible to take advantage of parallel execution in the classification phase, where multiple classifiers are executed concurrently. This creates the opportunity to sensibly reduce the computational time of the prediction. The prediction algorithm is then translated into a task graph that is submitted to Aneka. Once the task is completed, the middleware makes the results available for visualization through the portal.

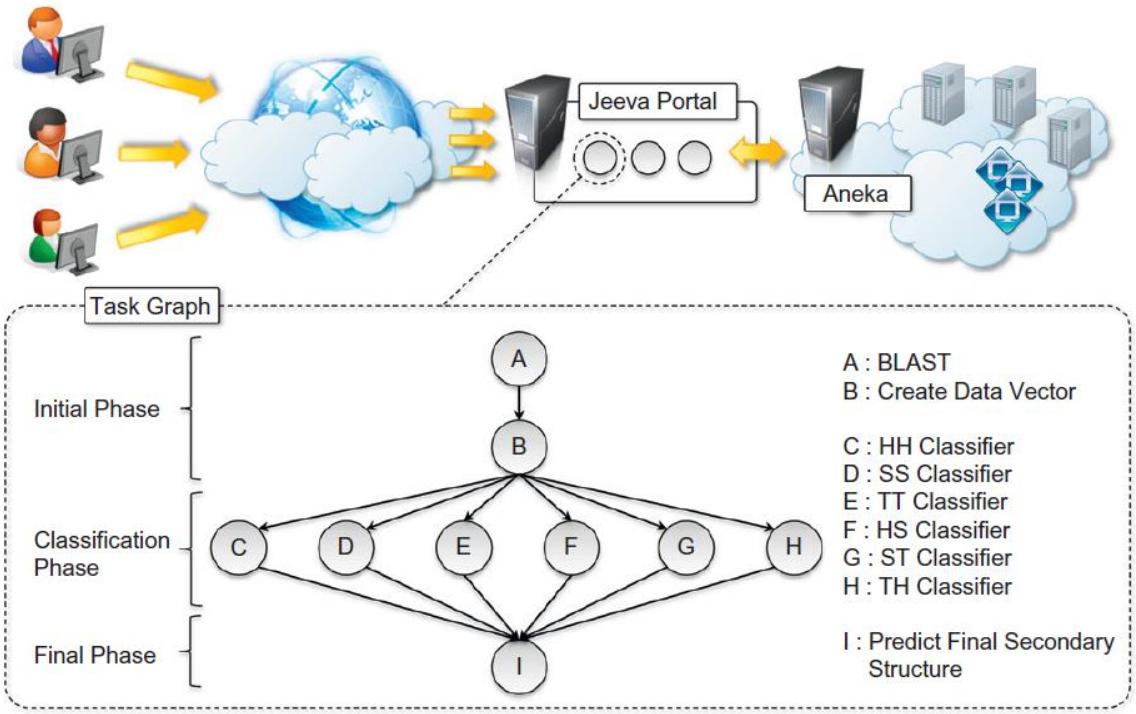


FIGURE 10.2

Architecture and overview of the Jeeva Portal.

15) Explain how gene expression data analysis for cancer diagnosed using cloud technology.

Gene expression profiling is the measurement of the expression levels of thousands of genes at once. It is used to understand the biological processes that are triggered by medical treatment at a cellular level. Together with protein structure prediction, this activity is a fundamental component of drug design, since it allows scientists to identify the effects of a specific treatment.

Another important **application of gene expression profiling is cancer diagnosis and treatment.** Cancer is a disease characterized by uncontrolled cell growth and proliferation.

This behavior occurs because genes regulating the cell growth mutate. This means that all the cancerous cells contain mutated genes. In this context, gene expression profiling is utilized to provide a more accurate classification of tumors. The classification of gene expression data samples into distinct classes is a challenging task. The dimensionality of typical gene expression datasets ranges from several thousands to over tens of thousands of genes. However, only small sample sizes are typically available for analysis.

Some of the tools for gene expression analysis are

1. AltAnalyze
2. Dchip
3. Babelomics suit
4. Myrna

- The **eXtended Classifier System (XCS)** has been successfully utilized for classifying large datasets in the bioinformatics and computer science domains. However, the effectiveness of XCS, when confronted with high dimensional datasets (such as microarray gene expression data sets), has not been explored in detail.
- **Cloud-CoXCS**, is a machine learning classification system for gene expression datasets on the Cloud infrastructure.
- It is composed of three components: **CoXCS, Aneka, and Cloud computing infrastructure.**
- CoXCS divides the entire search space into subdomains and employs the standard XCS algorithm in each of these subdomains.
- Such a process is computationally intensive but can be easily parallelized because the classifications problems on the subdomains can be solved concurrently.
- Cloud-CoXCS (see Figure 10.3) is a cloud-based implementation of CoXCS that leverages Aneka to solve the classification problems in parallel and compose their outcomes.

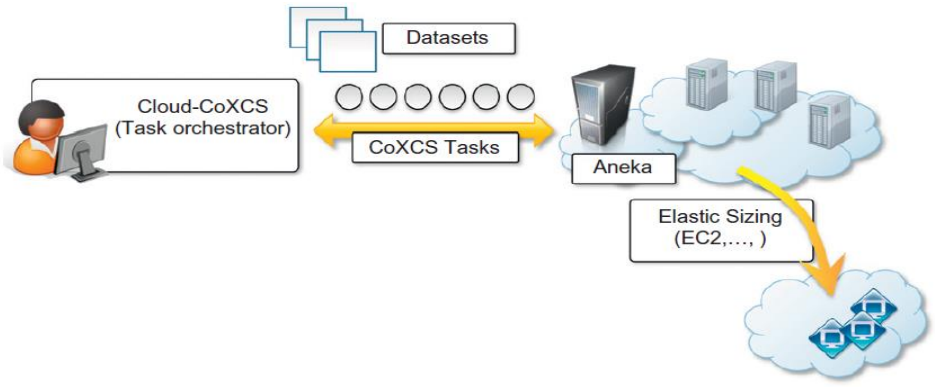


FIGURE 10.3
Cloud-CoXCS: An environment for microarray data processing on the cloud.