Lecture Notes: Introduction to Cloud

Basics of Cloud Computing

1.1 Introduction to Cloud Computing

Traditionally, organisations used to have physical servers and data centres. All these resources had to be bought by the organisation and they would also need to hire employees to take care and maintain them. The entire ownership and management of these resources is the organisation’s responsibility. In case of any outages, it is again the organisation’s responsibility to fix them. Outages also used to cause huge losses to the organisations. The organisation would also need to employ a lot of people to take care of these resources, which would result in increased expenditure. Also, there was an upfront cost that the organisation had to pay when purchasing these resources. Scaling up is another major issue that organisations face. These were some of the challenges associated with the traditional approach.

Cloud computing offers some distinct advantages over traditional data centres and has the capability to address the challenges associated with traditional data centres.

Cloud computing is an on-demand delivery of IT resources over the internet. These resources can be accessed on a need basis. Whenever there is a requirement for resources, you can get them over the internet. These resources consist of both the computing power and the storage space. When resources are used from the cloud, you follow a pay-as-you-go model. You only pay for the number of resources that you are using and the duration you are using them for. This also removes the upfront cost involved in buying the resources. The maintenance of these resources is also taken care of by the cloud service providers, thus saving organisations a significant amount of money. You can scale as and when required. Whenever there is an increase in requirement, you can easily get more resources. With cloud computing, organisations now spend less amount of monetary and human resources on IT infrastructure and can instead focus more on their business. This ultimately leads to low operation costs and more efficient operations.

Earlier, Netflix used to have servers and data centres on their premises. In 2008, Netflix suffered a major outage at one of its data centres, which lasted three days, resulting in poor customer experience. So, they decided to rent all the infrastructure required and direct their focus towards business development. Once they decided to move to cloud, their business expanded, and they were able to serve a lot of content to their customers.
Some of the advantages of cloud computing are as follows:

Cost: With cloud computing, you no longer have to buy all the infrastructure. You can just rent these resources, which leads to significant savings.
Performance: Cloud service providers are responsible for updating the hardware and software resources, thus helping you achieve greater efficiency.
Security: Cloud service providers ensure that policies are in place to make the infrastructure, application and data more secure.
Scale: Using cloud computing, you can easily scale up when needed. During peak times, more resources can be provisioned and in case of less traffic or less demand for the resources, it can be automatically scaled down.
Speed: Using the dashboard or management console of the cloud service providers, you can quickly spin up new resources.
Productivity: This enables organisations to be more productive as they can focus more on their business and do not need to worry about managing the IT resources.

1.2 N-Tier Architecture

Any application comprises the presentation layer, application layer and data layer. The presentation layer is the interface with which users interact. It could be any page of a website. The application layer consists of all the logic that is associated with the application or the system. All the information is stored in the data layer and the relevant information is fetched from this layer. These layers can be combined in different ways.

In the 1-tier system, the presentation, application and data layers are combined. Any change in one layer will need a new version of the entire application. These systems are also known as client-only applications. These are usually shipped as a single software package; for example, MS Office. The different components of the applications are tightly coupled and a change in any one layer will need a new version of the entire application.

2-tier systems, which are also known as client-server applications, have combined presentation and application layers. The data layer though is separate. The different components communicate with each other over the internet. It is easy to change and scale one layer independently of the other.

3-tier systems, which are also known as web applications, have separate presentation, application and data layers. The different layers are loosely coupled, and the components make use of the internet to communicate with each other. Since these layers are separated from each other, any changes and scaling up or down can be done in each layer independently.

N-tier systems, which are also known as distributed applications, are an upgraded version of 3-tier systems. They contain some additional layers over the presentation, application and data layers. These systems are highly scalable across each layer. They are usually available as web and mobile applications with distributed backends. Since all the layers are separated from each other, they are highly scalable across each layer. These systems make full use of cloud computing.

1.3 Service Models

Service models are defined based on who controls what resources.

Following are the different service models:

On-premise: Everything is self-managed in this model. The hardware, operating systems, runtime, applications and data are self-managed in this service model. You need to buy servers, virtual machines, storage, networking, software, etc. You need to estimate your usage before buying the infrastructure resources and dedicated teams will be required to manage these resources.

IaaS: In this model, the infrastructure is taken on rent and all the hardware is cloud-managed. You need to manage the runtime, OS, application and data. You can add and remove resources as and when required. The billing for this model depends on the number of resources used. Examples of this model are Amazon EC2 and Google Compute Engines.

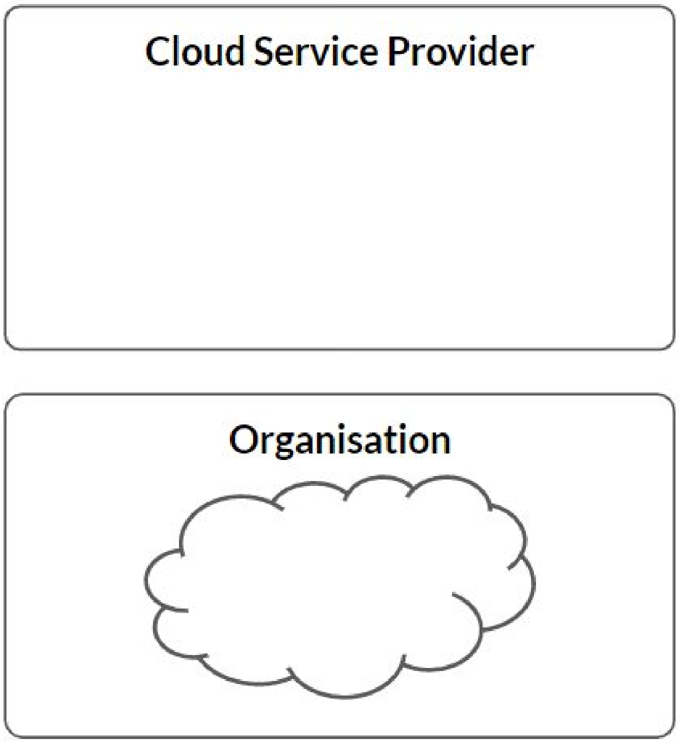
PaaS: This model provides you with an on-demand environment with hardware, software and development tools. In this model, the entire hardware, runtime and OS are cloud-managed, and you only need to manage the data and application. You can develop, run and manage applications without worrying about infrastructure and other setups. You are billed for the number of features you use and the number of hours you use them for. Examples of this model are AWS Elastic Beanstalk and Heroku.

SaaS: In this model, the entire resources are cloud-managed. These can be accessed using web browsers or mobile applications and the billing model here is subscription-based. Cloud service providers take full responsibility for hardware, software, storage and security. Examples of this model include Netflix, Facebook and Google apps.

1.4 Deployment Models

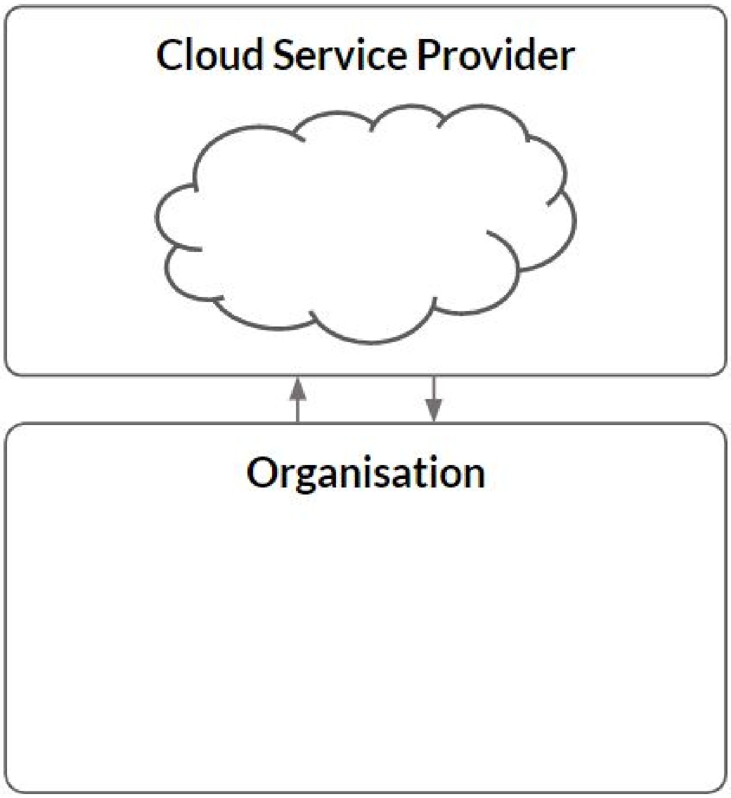
Deployment models are defined based on the location of the infrastructure.

The different deployment models are private, public, hybrid and multi-cloud.



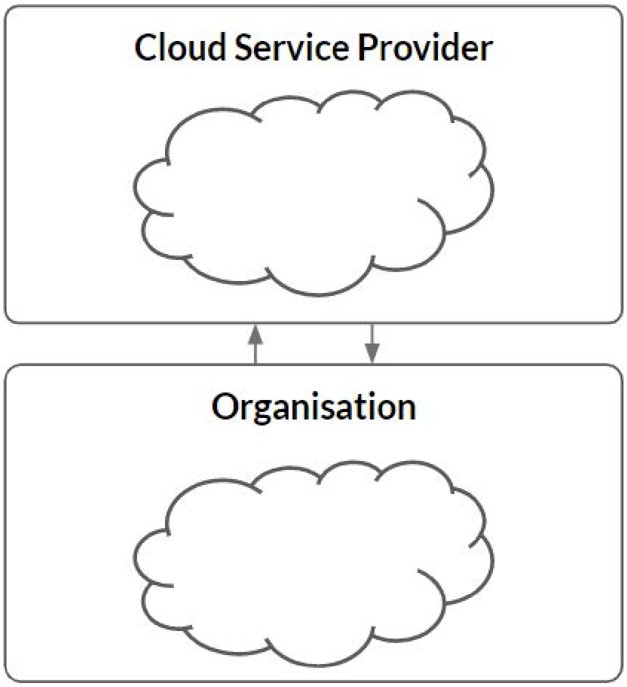
Private Cloud

In private cloud, a single organisation completely owns the entire cloud service. This is similar to an on-premise service model. They cater mostly to a single customer only. The entire cloud is located in the customer’s on-premise data centre and offers high control over data security and confidentiality.



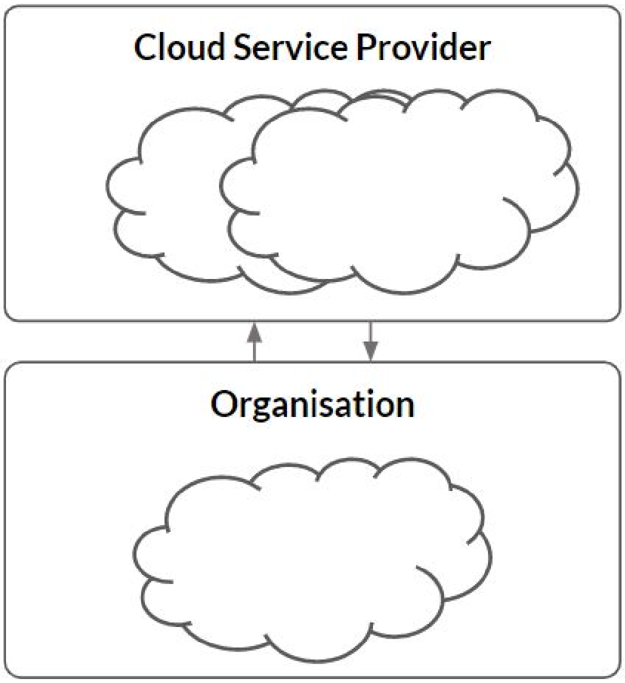
Public Cloud

In **public cloud,** the cloud service provider owns the entire cloud service. The hardware and infrastructure are also located in different parts of the world. Multiple customers share the cloud resources, and you can access these resources using a web console provided by the cloud service provider over the public internet.



Hybrid Cloud

A **hybrid cloud** is a combination of the private cloud and the public cloud. The organisation manages some part of the infrastructure and the cloud service provider manages the rest. This enables easy movement between the clouds as per the requirements. In most cases, the organisation takes care of the storage and the cloud service provider manages the server and the application. It also allows us to move between different clouds depending on the business requirements.



Multi Cloud

In the **multi-cloud** model**,** organisations turn to multiple cloud service providers. It offers the flexibility in choosing the right cloud service provider for the right use case and also allows organisations to choose the cloud service provider that offers them the lowest price for the services.

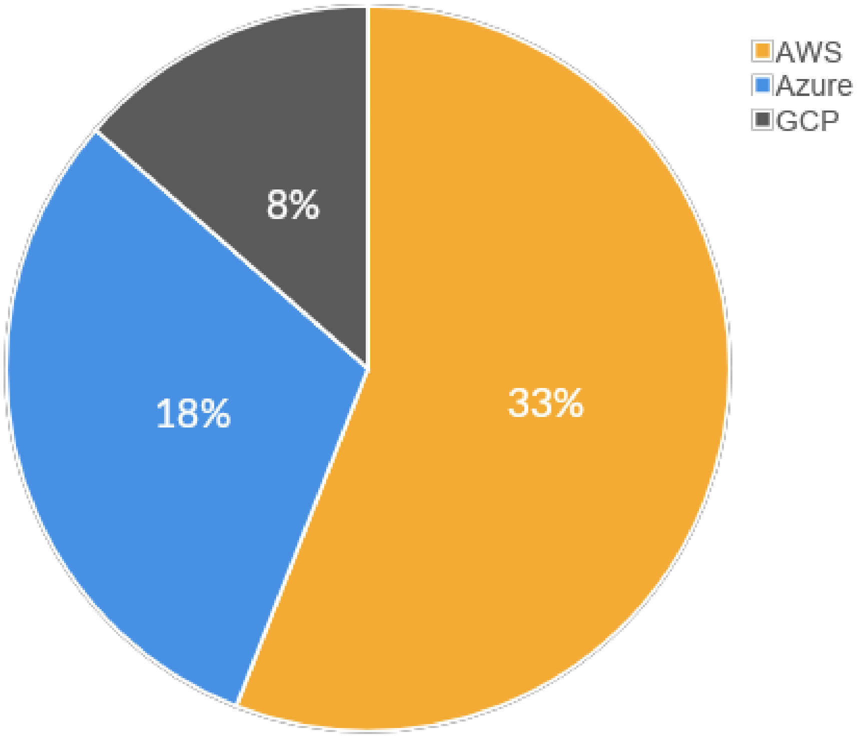
1.5 Characteristics of Cloud Computing

Economies of scale: Cloud is used by many customers across the world. This helps cloud service providers achieve economies of scale.
Measured service: Resource utilisation is tracked for each application and each user who is availing resources from cloud service providers. It will provide both the user and the resource provider a correct account of service utilisation. You are only charged for the resources that you are using.
Total cost of ownership: Total expense and actual cost that goes into buying IT resources are significantly reduced because of the pay-as-you-go model. You end up paying only for the resources that you have used. This significantly reduces the costs.
Rapid elasticity: Cloud computing provides rapid elasticity. When there is an increase in demand for the resources, it scales up, and as the demand decreases, it scales down. Resources are allocated to the user when there is a requirement and some of the resources are removed once the requirement decreases.

Load balancing: Traffic to an application is automatically distributed between different servers. This improves the application's performance.
Resource pooling: IT resources are shared across multiple applications and occupants. Multiple users are serviced from the same physical resource.
Loose coupling: Different application components are loosely coupled allowing each component to scale independently of the other.
Fault tolerance: Multiple replicas of each service are run so that if one of them fails, the system falls over to another replica to ensure that the application is not affected.
Security: Cloud service providers enforce strong preventive, detective and corrective measures to ensure that customer data is secure from unauthorised access and malicious attacks.

1.6 Cloud Providers

Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platforms (GCP) are some of the major cloud service providers. In terms of market share, AWS leads the charts with a 33% share, followed by Microsoft Azure at 18% and GCP at 8%.



Market Share for Major Cloud Service Providers

Virtualisation

2.1 Virtualisation

Virtualisation refers to the creation of a virtual version of physical resources such as hardware components and network resources. It is the use of software to simulate hardware. Software is used to create an abstraction layer over hardware that allows it to divide the hardware of a single machine into multiple virtual machines. It enables a more efficient utilisation of physical hardware. It helps cloud service providers to serve multiple users with their existing physical computer hardware. Cloud users only need to purchase the computing resources that they need.

A virtual machine (VM) is a virtual representation of a physical computer. Multiple VMs, each with their own operating systems and applications, can be hosted on a single computer. It leads to better resource utilisation since multiple VMs can run on the same physical hardware.

A hypervisors is a lightweight software layer that helps create and run VMs. It coordinates between a VM and the underlying physical hardware and ensures that VMs do not interfere with each other. Moreover, it allows one host to run multiple VMs on them and ensures that the VMs do not interfere with each other. Type 1 hypervisors, also called bare metal hypervisors, run directly on the physical hardware. Type 2 hypervisors run as an application within a host operating system. Reduced expenses, increased efficiency and faster provisioning of resources are some benefits of virtualisation.

2.2 Containerisation

Containerisation is packaging up the entire software code and various dependencies into a single package. This ensures that the code can run uniformly on any infrastructure. All the related configuration files, libraries and dependencies required for a code are bundled together into a single package. It allows applications to follow the 'write once and run anywhere' principle. Portability, speed, agility, efficiency and security are some of the benefits of containerisation. Some of the differences between containerisation and virtualisation are listed in the following table.

Feature	Virtual Machine	Container
Operating System	Runs a complete operating system	Runs only the user mode and

		applications required
Storage	Creates and uses a virtual hard disk per VM	Shares the storage of the host across multiple containers
Isolation	Complete isolation from the host and other VMs	Lesser isolation than VMs due to shared host OS
Patching and Updates	Updates need to be downloaded and installed in each VM	Dockerfile needs to be updated and rebuilt
Guest Compatibility	Can run any OS in the VM	Needs to run on the same OS as the host
Fault Tolerance	VMs can fall over to other VMs in cluster when they fail	Containers are just recreated immediately after failing

2.3 Docker

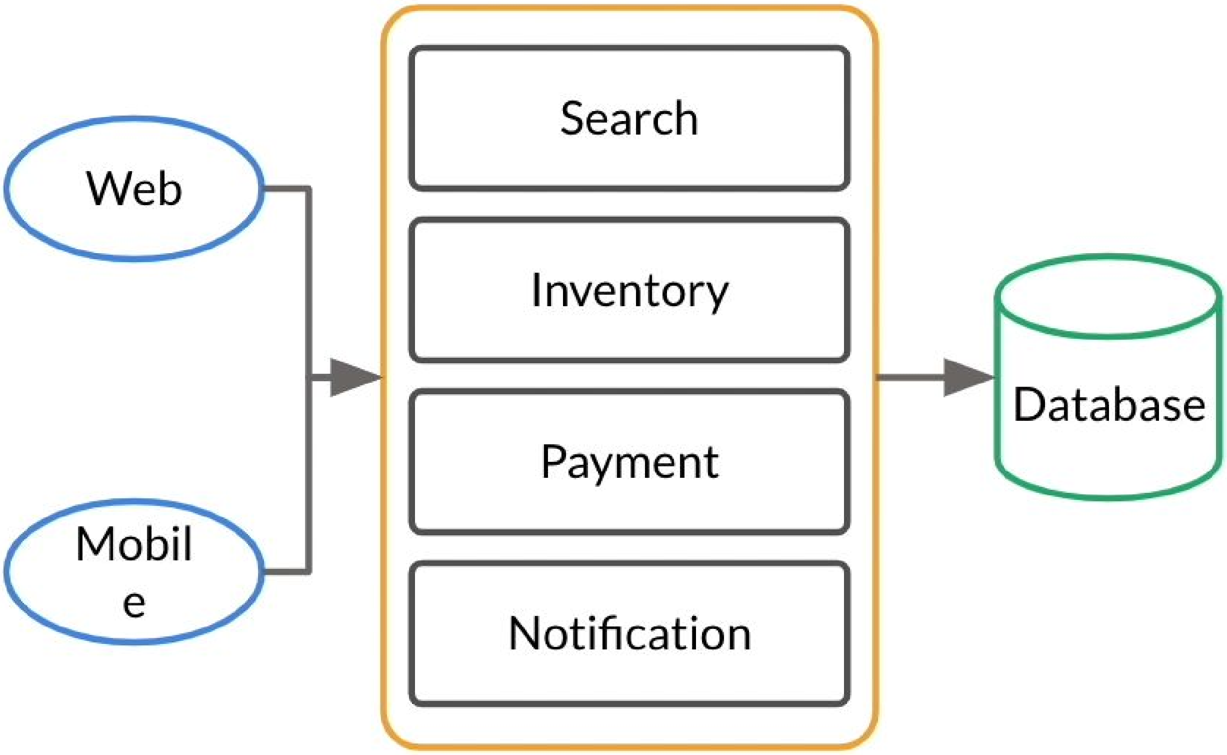
Docker is an open-source containerisation platform for building, deploying and managing containerised applications. It is a toolkit that enables developers to build, deploy, run, update and stop containers using simple commands and work-saving automation.

Some of the key terminologies related to Docker are as follows:

- **Dockerfile**: A list of commands that Docker Engine will run in order to assemble the image
- **Docker image**: Contains executable application source code as well as all the tools, libraries and dependencies that the application code needs to run as a container
- **Docker containers**: The live, running instances of Docker images
- **Docker Hub**: The public repository of Docker images that calls itself the ‘world’s largest library and community for container images’

Cloud Architecture

3.1 Monoliths



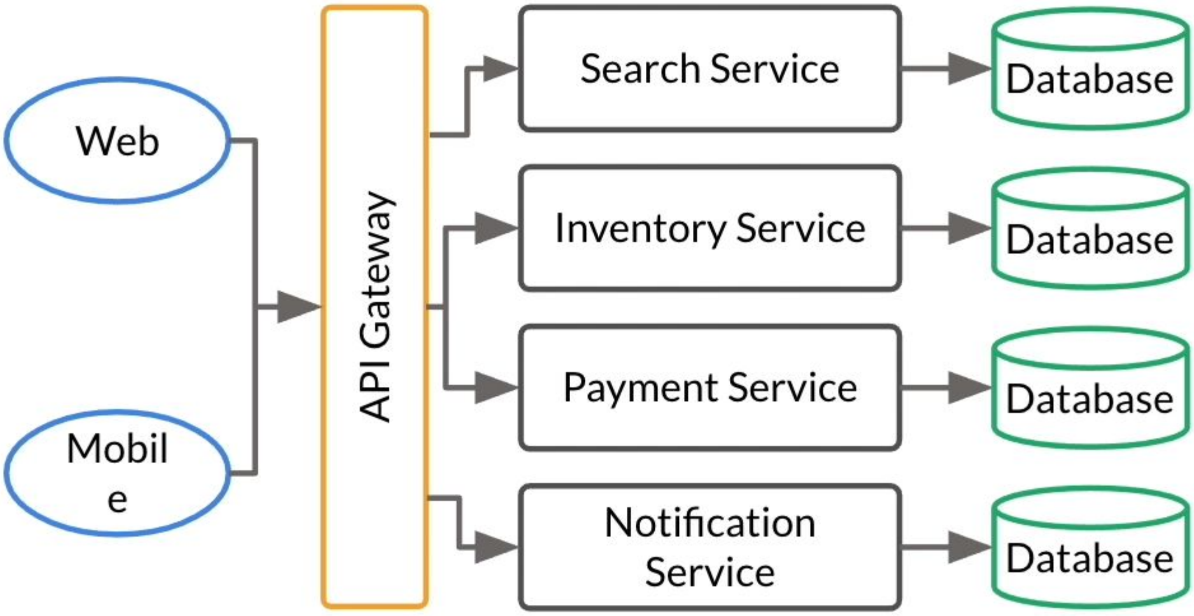
Monolith Architecture

Monolith is an approach in which the entire application is packaged as a single unit. The entire code for different components of the application is written together, and all the components are deployed together. You have a single database for the entire application and the same application is deployed for all the platforms. Communication between different components of the application is a simple code invocation. This approach comes with its set of advantages and disadvantages, which are provided in the following table.

Advantages	Disadvantages
The entire application is part of a single code base, thus making it easier to import and develop in any Integrated Development	iAs the code size increases, it becomes difficult to manage, thus decreasing the development speed.

Environment (IDE).	
Testing is easier as the entire functionality is available for testing.	The increased code size overloads the IDE and the server, which lowers productivity when making changes to the application.
It is easy to replace in the runtime environment by bundling it together as a single unit.	A change in any part of the application will need the entire application to be tested and deployed again.
Running copies of the application is made easier by just copying the entire application to different instances.	Any issue in one of the components will bring the entire application down.

3.2 Microservices



Microservices Architecture

In the microservices architecture, an application consists of many loosely coupled services. These services can be deployed independently. Each service has its own stack, including a database, and can be considered to be a small application in itself. Based on the business requirements, a bigger application is broken down into many small components.

This approach also has several advantages and disadvantages, which have been listed in the following table.

Advantages	**Disadvantages**
The code is easy to understand and modify.	Complexity arises due to the distributed nature of applications.
It is faster to test and deploy the application.	Strong cooperation is required between multiple teams if any service needs to communicate with the other services.
Multiple teams can work on different components independently of the others.	There is an additional overhead for the organisation to manage a system composed of multiple services.
If there is any bug in any one service, the other services are not affected.	More resources are required for developing and maintaining the application.

3.3 Event-Driven Architecture

The different components of an application can communicate with each other in one of the two ways – synchronous or asynchronous. In synchronous communication the exchange of information takes place in real time. The client sends a request to the server and waits for the response from the server. During this process, the client is completely blocked. HTTP/HTTPS are synchronous communication protocols. In asynchronous communication the exchange of information takes place in non-real time. This means that after the client has sent a request to the server, it can continue to do other tasks and not wait for a response from the server. This type of communication is achieved through message queues such as Apache Kafka or Amazon Web Services (AWS) Simple Queue Service.

Event-driven architecture makes use of events to communicate between different services. An event signifies a change in the state of the system. Events can carry the entire state or an identifier for the state change. The producer publishes the event that is put in the queue or bus and is subsequently listened to by the consumer. The consumer then takes action based on the event. A queue is a temporary storage that holds messages until they are processed or consumed.

Message queues are temporary storages that hold messages until they are processed. It allows different components and services to communicate and process tasks asynchronously. Multiple producers and consumers can use the same message queue at the same time.

Compute, Storage, Database and Networking

4.1 Compute Engines

Compute engine is a set of virtualised hardware resources available to a VM instance. These virtual resources include the following:

- CPU: Defines the number and type of processors
- Memory: Defines the size and speed of memory (RAM)
- Storage: Defines the disk space present in the system to store data
- Networking: Defines the dedicated network bandwidth or speed

Compute engines are classified into different families depending on the workload. The three different families are as follows:

- **General purpose:** These instances offer a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. Such instances are ideal for applications that use these resources in equal proportions such as web servers. If you are unsure about the kind of workload your application will handle, or you are just starting with a new application, then general purpose machines can be used.
- **Compute-optimized**: These offer extremely high CPU performance for compute-intensive applications that benefit from high-performance processors. These machines are ideal for batch processing workloads, media transcoding, gaming servers, machine learning, etc. Compute-optimized instances can be very useful in this case.
- **Memory-optimized:** These offer higher memory and have more memory per core than others. These machines are ideal for memory intensive applications like in-memory databases. For e.g., if you are running a distributed cache like Redis, you should use memory-optimized machines.

Compute families are further broken down into instance types. So, each family of machines internally has multiple instance types (small, large, extra-large, etc.) depending on the quantity of resources used.

4.2 Storage

Local storage involves storing data on some kind of a physical storage device such as hard drives, SSDs, pen drives, etc. Data is stored offline, so you do not need the internet to access data. Lower storage volumes are pretty inexpensive. You remain the custodian of the and have complete control over data security. One of the largest disadvantages of local storage, however, is that it is vulnerable to hardware failure, data loss and theft. Local storage also means limited availability. Data can only be accessed at the same physical location where it is stored.

Cloud storage is storing data in an online space such as Google Drive, Dropbox or Amazon S3. Once you have stored your data in any online space, it gives you the ability to access that data from any place or device. This also enables multiple people to work and collaborate on the data simultaneously. Since the data is stored in a shared online space, you are not the sole custodian of that data anymore. The security of your data is partially controlled by the third-party cloud service provider like AWS or Azure. As you are storing data online, you will always need the internet to access the stored data. As is the case with other cloud resources, here also you need to pay only for the amount of the storage that you are using. The storage space is virtually unlimited and you can store any amount of data you need. Although rare, cloud storage is still vulnerable to leaks and attacks, that can lead to loss of data, or data being made available to unauthorised people.

The different types of cloud storage are as follows:

- **File storage:** Here the data is stored in a hierarchical system divided into directories and subdirectories (files and folders). The same file system can be accessed by multiple devices at the same time. Due to the complex storage mechanism, file storage has higher latency than other storage types, and also suffers from limited scalability. An example for file storage is Amazon Elastic File System.
- **Block storage:** Here the data is broken into multiple blocks and stored across distributed systems. Each block represents a separate hard drive. This type of storage is very useful for use cases like databases where we need low latency and high availability. However, block storage has the highest cost per gigabyte (GB) of storage space and becomes very expensive for large data sets. An example of block storage is Amazon Elastic Block Store.
- **Object storage:** Here, the data is stored as an object which consists of the data, some metadata, and a unique identifier. Data can be retrieved by knowing the unique identifier, or by searching for metadata. This type of storage is useful for storing very large data sets of unstructured data. They offer high throughput and very high scalability. An example of object storage is Amazon Simple Storage Service (S3).

4.3 Databases

Cloud databases can be thought of as database services available in the cloud. Two common deployment models that are available in the cloud for using database services listed are as follows:

- **User-managed:** Users provision the compute capacity from cloud service providers and manage their own database installation. So, you can get the required virtual machine,

for e.g., an EC2 server from AWS, and then deploy your database on it. You will have to manage the installation as well as maintenance and upgrades of the database yourself.
- **Cloud-managed:** Users can get a fully managed database service provided by the cloud service provider. This is also known as Database-as-a-Service (DBaaS). You can get a fully managed database such as Amazon DynamoDB, and AWS will take care of managing the database, applying upgrades, and handling the underlying infrastructure.

Cloud databases support both relational databases such as MySQL and PostgreSQL and non-relational databases such as MongoDB and DynamoDB.

4.4 Networking

Cloud networking can be understood as networking between a set of virtual resources that are hosted on the cloud. These are network capabilities and resources that are provided by cloud service providers to enable communication between VMs and other cloud resources that are hosted in a public or private cloud platform. In a traditional scenario, these networking capabilities are provided by physical appliances such as switches, routers and local area network (LAN) cables. Network resources for the cloud can be virtual routers, virtual firewalls, network adaptors, local area networks, switches as well as any other network management software that is exposed as an application programming interface (API) or command line interface (CLI).

Two common models for using network resources are as follows:
- **Cloud-enabled:** In cloud-enabled networking, the actual physical network is present on-premises, but some or all resources used to manage it are in the cloud. Core network infrastructure remains in-house, and software for network management, monitoring, maintenance, and security services are present in the cloud. One example is using a SaaS-based firewall such as Cloudflare to protect an on-premises network.
- **Cloud-based networking:** In cloud-based networking, the entire network is in the cloud. This includes both network management software resources and physical hardware resources. Cloud-based networking is used to provide connectivity between applications and resources deployed in the cloud. Examples of cloud-based networking are virtual private clouds (VPCs).

4.5 Virtual Private Cloud (VPC)

A virtual private cloud (VPC) is a private cloud-like environment within a public cloud service provider’s infrastructure. It provides the ability to define isolated virtual networks and deploy cloud resources inside them. A VPC has its own logically isolated virtual network. So, all resources inside a VPC live in a secured private space that is isolated from other tenants and users of the cloud service provider. Inside a VPC, you can control the IP addresses and applications that can access particular resources. So, for e.g., you could be running an EC2 instance and a database in the same VPC. You can then allocate specific IP addresses or IP address ranges that can be used by these resources.

When required, you can dynamically increase or decrease the size of virtual networks. For e.g., you can start with a small range of IP addresses for your VPC, and when the number of resources increases, or you need more IP addresses, you can expand the range. The resources present inside a VPC are at a reduced risk because they are isolated from other customers and applications. Within a VPC, you can further create smaller portions or subdivisions called subnets. These subnets contain the actual machines or instances.

4.6 Security

Cloud security involves the practices, procedures, technologies and controls used to protect data, services, applications, and the related infrastructure in the cloud. It includes protecting these resources from both internal and external threats. There are three types of activities involved in cloud security.

- **Preventive:** Preventive measures try to minimise threats by enforcing strong security controls like user authentication, authorisation and encryption. You can define a list of authorised users that have access to your application, and make sure that these users are verified before they get access to the data. Similarly, you can encrypt your data so that in case the data gets leaked, the attacker will not be able to make any sense of the encrypted data.
- **Detective:** Detective measures continuously monitor your systems to identify and handle ongoing threats like detecting malicious activities. These are similar to intrusion detection systems and run a variety of machine learning algorithms to identify any suspicious activities in your systems. For e.g., if your system is available only to users from your own country, you could configure an alarm that goes off if your system is accessed from an IP address outside your country.
- **Corrective:** Corrective measures are taken as part of post-attack damage control and remediation. This may include activities like restoring database backups or changing the access controls such as resetting user passwords.

Some of the different security vulnerabilities present in cloud are as follows:

- **Data loss:** Malicious attacks or hardware failures can often lead to data corruption or deletion. Based on the configuration, you can either lose some part or the whole of data stored in the cloud.

- **Data breach and leak:** Unauthorised access to the data can lead to leak of confidential information to the public. Sometimes, this data is also sold in the black market or held for ransom.
- **Weak access management:** Poor access management techniques can lead to weak security controls around user authentication data. This allows attackers to compromise such information and gain unauthorised access to your data.
- **Insecure APIs:** Not having strict access restrictions for public interfaces might lead to malicious access of code in unwanted and unexpected ways.
- **Misconfiguration:** Leaving security settings at default or not auditing changes to settings can lead to misconfiguration and leave the resources open for attackers.
- **Denial of service:** Attackers generate very heavy traffic and keep the application busy, thus preventing legitimate users from accessing the application and its services.

Some common techniques and solutions available in the cloud for securing resources are as follows:

- **Security groups:** A security group is a virtual firewall for VMs. It contains a set of rules to filter the incoming and outgoing traffic for an instance or machine. Security groups act at an instance level. If there are multiple instances in the same subnet, different security groups can be assigned to each instance. Security groups evaluate all the rules before allowing traffic. For instance, if you have 10 rules for a group, and the first rule itself does not match, the rest nine rules will still be evaluated.
- **Access control lists:** Access control lists (ACLs) are network traffic filters that can control the incoming or outgoing traffic. This is an additional layer of protection over the security groups and contains a list of ‘who has what?’ access to the cloud resources. The ACLs are applicable at the subnet level, so all the resources in a subnet inherit the same rules. In security groups, all the rules are evaluated. In ACLs, rules are evaluated in the same order as they are declared. For example, if you have 10 rules for a group, and the first rule itself does not match, the traffic is blocked there itself and the other rules are not evaluated.
- **Web application firewall (WAF):** WAF is a firewall that protects the web applications by monitoring and filtering HTTP traffic from the internet. A WAF uses a set of rules or policies that aim to protect the applications from vulnerabilities by filtering out the malicious traffic. A WAF can be based either on a blocklist or a whitelist. A blocklist-based WAF provides protection against known attacks, whereas a whitelist-based WAF only admits the pre-approved traffic.