Cloud deployment models

The various types of cloud-computing deployment models include *public cloud, private cloud*, and *hybrid cloud*.

>Public cloud

Public clouds are owned and operated by cloud providers that offer rapid access over a public network to affordable computing resources.

Here are key aspects of a public cloud:

1. Enables flexible and scalable IaaS for storage and compute services at a moment's notice.
2. Enables powerful PaaS for cloud-based application development and deployment environments.
3. Gives access to innovative SaaS business apps for applications ranging from customer resource management (CRM) to transaction management and data analytics.

A private cloud is an infrastructure that is operated solely for a single organization. Private clouds can take advantage of cloud's efficiencies while providing more control of resources and allowing clients to steer clear of multitenancy.

Here are key aspects of a private cloud:

- Provides self-service interface controls services, which enable IT staff to provision, allocate, and deliver quickly on-demand IT resources.
- Facilitates highly automated management of resource pools for everything from compute capability to storage, analytics, and middleware.
- Provides sophisticated security and governance for a company's specific requirements.

>Hybrid cloud

A hybrid cloud uses a private cloud foundation that is combined with the strategic integration and use of public cloud services. Most companies with private clouds evolve to manage workloads across data centers, private clouds, and public clouds, which creates hybrid clouds.

Here are key aspects of a hybrid cloud:

Enables companies to keep critical applications and sensitive data within a traditional data center environment or private cloud.

Enables taking advantage of public cloud resources like SaaS for the latest applications and IaaS for elastic virtual resources.
Facilitates portability of data, apps, and services and more choices for deployment models.

>Cloud-native applications

- Applications are broken into separate services called microservices.
- Microservices can be developed in different programming languages (polyglot development).
- Microservices communicate with each other by using an agreed upon protocol (such as REST or gRPC).
- Microservices work together as a whole to make up an application, yet each can be independently scaled, continuously improved, and quickly iterated through automation and orchestration processes.

Some advantages of cloud-native apps:

- Compared to traditional monolithic apps, cloud-native applications can be easier to manage as iterative improvements occur through agile and DevOps processes.
- Composed of individual microservices, cloud-native applications can be improved incrementally and automatically to add continuously new and improved application features.
- Improvements can be made non-intrusively, causing no downtime or disruption of the user experience.
- Scaling up or down proves easier with the elastic infrastructure that underpins cloud-native apps.
- The cloud-native development process more closely matches the speed and innovation that is demanded by today's business environment.

Even with the advantages that are provided by cloud-native applications, there are also some disadvantages to consider:

- Although microservices enable an iterative approach to application improvement, they also create the necessity of managing more elements. Rather than one large application, it becomes necessary to manage far more small and discrete services.
- Cloud-native apps demand more toolsets to manage the DevOps pipeline, replace traditional monitoring structures, and control microservices architecture.

- Cloud-native applications allow for rapid development and deployment, but they also demand a business culture that can cope with the pace of that innovation.