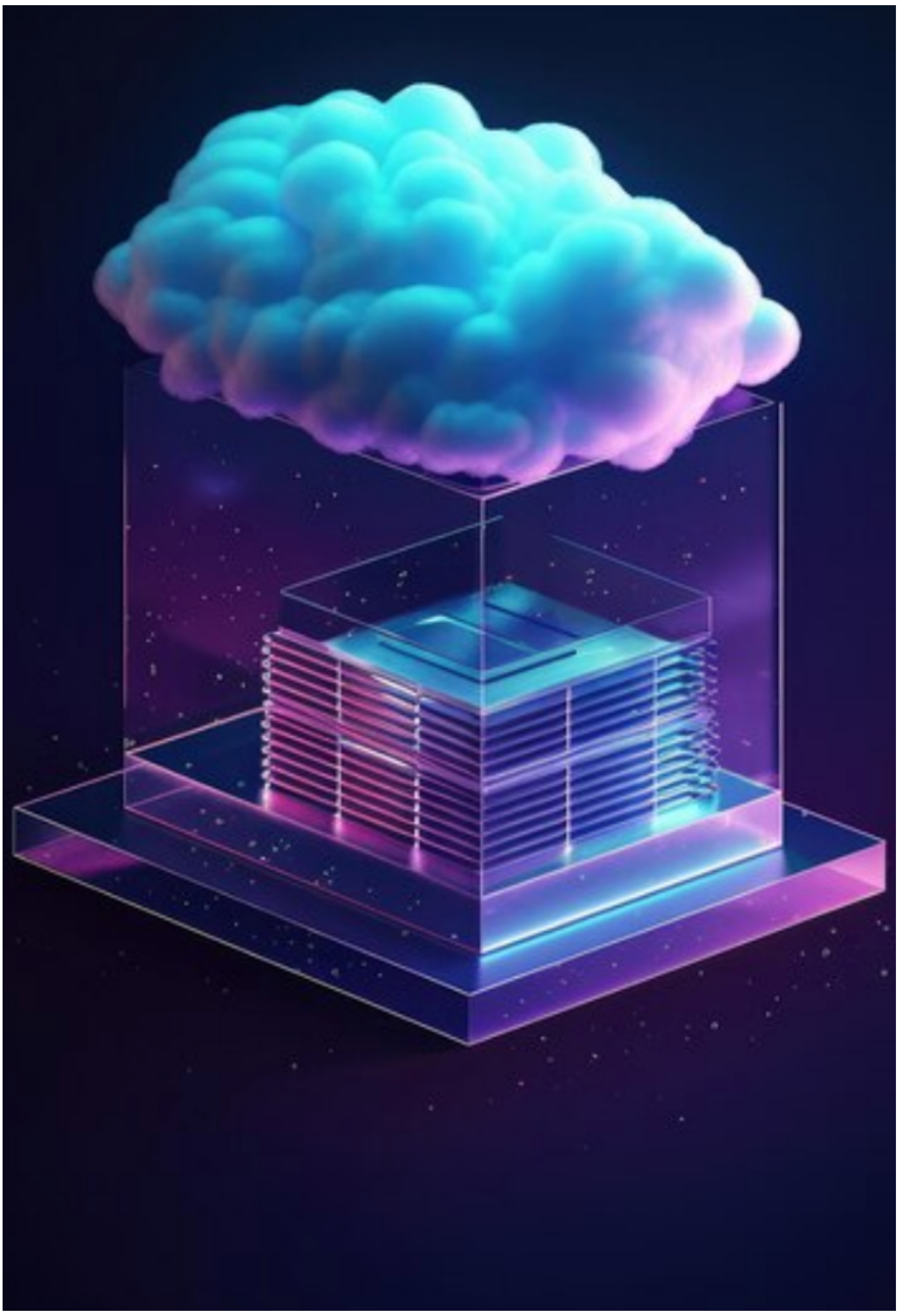
# Lesson 2

## What is cloud-native Pattern?



# Before the reading, explain what the skimming strategy is.

Skimming is a reading strategy where the reader quickly looks over a text to get a sense of its main ideas, structure, and content without reading every word. The purpose of skimming is to quickly identify the key points, main headings, and overall structure of a text. Skimming is often used to preview a text before reading it in-depth or to quickly review a large amount of material to determine its relevance.

## Key elements of skimming include:

- Reading Headings and Subheadings: Focus on the titles and subtitles of sections to understand the organization of the text.
- Reading the First and Last Sentence of Paragraphs: This can provide insights into the main idea of each paragraph.
- Looking at Bold or Italicized Text: These may indicate important terms or concepts.
- Examining Graphics and Visuals: Pay attention to charts, graphs, images, and captions as they often convey essential information.

# Socialize vocabulary about the reading: "What is cloud-native Pattern? ”

- Cloud-Native: Meaning: Refers to the software model for building, deploying, and managing modern applications in cloud computing environments.

- Microservices: Meaning: Small, independent software components that collectively form complete cloud-native software, each focusing on a specific problem.

- Immutable Infrastructure: Meaning: Servers hosting cloud-native applications remain unchanged after deployment; if more resources are needed, the old server is discarded, and the application is moved to a new server.

- API (Application Programming Interface): Meaning: A method for two or more software programs to exchange information; in cloud-native systems, APIs tie together loosely coupled microservices.

- Service Mesh: Meaning: A software layer managing communication between multiple microservices in cloud infrastructure, introducing additional functionality without writing new code.

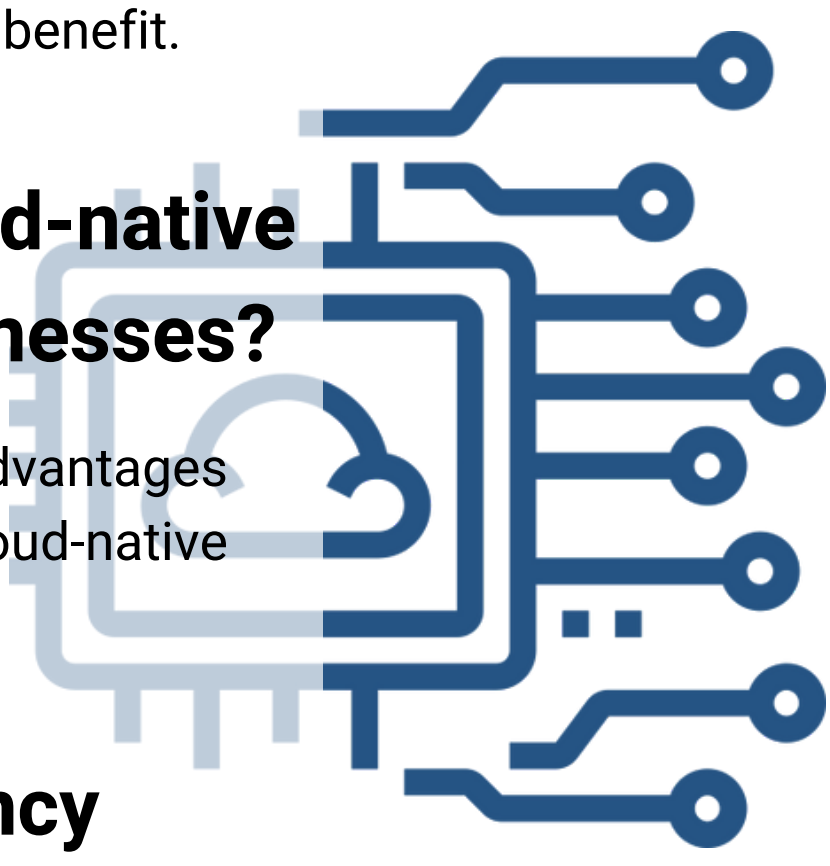
- Containers: Meaning: The smallest computing unit of a cloud-native application, containing microservice code and necessary files; allows applications to run independently of the underlying system.

- CNCF (Cloud Native Computing Foundation): Meaning: An open-source foundation supporting organizations in their cloud-native journey; develops critical cloud-native components, including Kubernetes.

# Reading: What is cloud-native pattern?

Cloud-native refers to the software model for building, deploying, and managing modern applications in cloud computing environments. Modern businesses want to create highly scalable, flexible, and resilient applications that can be quickly updated to meet customer demands. They do this by using modern tools and techniques that inherently support application development on cloud infrastructure. These cloud-native technologies support rapid and frequent application changes without affecting service delivery, providing users with a competitive and innovative benefit.

## How does a cloud-native approach benefit businesses?



Organizations gain competitive advantages in several ways when building cloud-native software applications.

## Higher level of efficiency

Cloud-native development brings with it agile practices like DevOps and continuous delivery (CD). Developers use automated tools, cloud services, and a modern design culture to create scalable applications quickly.

## Reduce cost

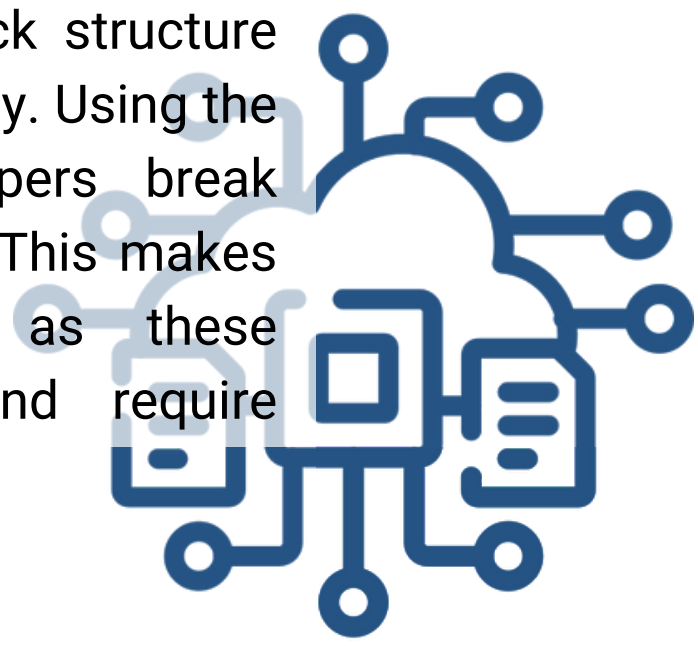
By adopting the cloud-native approach, companies do not have to invest in acquiring and maintaining expensive physical infrastructure. This translates into long-term savings in operating expenses. The cost savings from building cloud-native solutions can also benefit your customers.

## Ensure availability

Cloud-native technology enables businesses to build highly available, resilient applications. Feature updates cause no downtime, and businesses can scale up application resources during peak seasons to deliver a positive customer experience.

## What are cloud native applications?

Cloud-native applications are software programs that consist of several small, interdependent services called microservices. Traditionally, developers created monolithic applications with a single block structure that contained all the necessary functionality. Using the cloud-native approach, software developers break functionalities into smaller microservices. This makes cloud-native applications more agile as these microservices operate independently and require minimal computing resources to run.



## Cloud native applications compared to traditional enterprise applications

Traditional enterprise applications were built using less flexible software development methods. Typically, developers worked on a large batch of software functionality before its test release. As such, traditional enterprise applications took longer to deploy and were not scalable.

On the other hand, cloud-native applications use a collaborative approach and are highly scalable across different platforms. Developers use software tools to greatly automate the build, test, and deployment procedures in cloud-native applications. You can configure, deploy, or replicate microservices in an instant, something not possible with traditional applications.

## What is the CNCF?

The Cloud Native Computing Foundation (CNCF) is an open source foundation that helps organizations start their cloud native journey . Established in 2015, CNCF supports the open source community in the development of critical cloud-native components, including Kubernetes. Amazon is a member of the CNCF.

# What is cloud native application architecture?

Cloud-native architecture combines software components that development teams use to build and run scalable cloud-native applications. The CNCF lists immutable infrastructure, microservices, declarative APIs, containers, and service meshes as the technology building blocks of cloud-native architecture.

# Immutable infrastructure

An immutable infrastructure means that the servers for hosting cloud-native applications remain unchanged after deployment. If the application requires more computing resources, the old server is discarded and the application is moved to a new high-performance server. By avoiding manual updates, immutable infrastructure makes cloud-native deployment a predictable process.

# Microservices

Microservices are small, independent software components that work collectively as complete cloud-native software. Each microservice focuses on a small, specific problem. Microservices are loosely coupled, meaning they are independent software components that communicate with each other. Developers make changes to the application by working on individual microservices. This way, the application continues to work even if a microservice fails.

## API

Application programming interface (API) is a method that two or more software programs use to exchange information. Cloud-native systems use APIs to tie together loosely coupled microservices. The API tells you what data the microservice wants and what results it can provide, rather than specifying the steps to achieve the result.

## Service mesh

The service mesh is a software layer in cloud infrastructure that manages communication between multiple microservices. Developers use the service mesh to introduce additional functionality without having to write new code in the application.

## Containers

Containers are the smallest computing unit of a cloud-native application. They are software components that include the microservice code and other files required in cloud-native systems. By containerizing microservices, cloud-native applications run independently of the underlying operating system and hardware. This means that software developers can deploy cloud-native applications on-premises, in cloud infrastructure, or in hybrid clouds. Developers use containers to package microservices with their respective dependencies, such as resource files, libraries, and scripts that the main application needs to run.

**Benefits of containers**

- Some of the advantages of containers include:
- Uses less computing resources than implementing conventional applications
- You can implement them almost instantly
- You can scale the cloud computing resources your application requires more efficiently

Adapted from: https://aws.amazon.com/es/what-is/cloud-native/

# True/False questions about the previous reading.

Para consolidar las respuestas ingrese al cuestionario online.

- Cloud-native applications are typically monolithic, consisting of a single block structure with all necessary functionality.

- Immutable infrastructure ensures that servers hosting cloud-native applications undergo manual updates after deployment to maintain flexibility.

- Microservices in cloud-native applications are tightly coupled, meaning they are dependent on each other for functionality.

- The Cloud Native Computing Foundation (CNCF) was established in 2015 to support closed-source initiatives in cloud-native development.

- Containers in cloud-native applications allow them to run independently of the underlying operating system and hardware, facilitating deployment in various environments.