The background of the slide features a dynamic, abstract scene. A large, translucent, glowing ring in shades of purple and blue curves across the center. Numerous small, semi-transparent spheres in pink, yellow, and light blue are scattered throughout the space, some appearing to orbit the central ring. The overall effect is futuristic and suggests a theme of data, connectivity, or the digital cloud.

Emerging Trends in Cloud Computing

PIYUSH PANT



What is Edge Computing?

Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location where it is needed, to improve response times and save bandwidth.

Instead of sending all data to a centralized cloud for processing, edge computing enables local devices or edge servers to perform necessary computations.

With the growth of IoT, 5G, and real-time applications, processing data at the edge is becoming essential to avoid delays and dependence on constant internet connectivity.

Why Edge Computing?

- **Latency Reduction:** Real-time data processing is possible due to proximity to the data source.
- **Bandwidth Efficiency:** Not all data needs to be sent to the cloud; only important summaries or alerts are sent.
- **Enhanced Reliability:** Systems can function independently even during network outages.
- **Security and Privacy:** Sensitive data can be processed locally, reducing exposure to the internet.
- **Example Scenario:** A factory machine detecting overheating and shutting down immediately without waiting for a cloud server response.

Edge Computing Workflow

- **Data Generation:** Devices like sensors, IoT machines, and smart appliances generate real-time data.
- **Local Processing:** The data is sent to an edge device (e.g., microcontroller, embedded server, or smart router) for immediate analysis.
- **Action:** The edge device takes necessary actions like triggering an alert or adjusting machine operations.
- **Cloud Sync:** Only valuable insights, reports, or exception cases are uploaded to the cloud for long-term storage and analytics.
- **Visualization:** A smart camera detects motion, processes video locally, and only uploads the motion clip instead of streaming continuously.

Real-World Examples of Edge Computing

- 1. Smart Surveillance Cameras:** Real-time motion detection and alerting.
- 2. Smart Agriculture:** Soil and environmental sensors control irrigation.
- 3. Autonomous Vehicles:** Real-time obstacle detection and navigation.

Additional Examples:

- Health wearables detecting abnormalities and alerting users
- Retail stores monitoring inventory through local sensors





Smart Surveillance Camera Use Case: Enhance home or business security through real-time video analysis.

Workflow:

- Cameras detect motion using infrared sensors.
- AI within the camera distinguishes between human, animal, or object.
- If a threat is detected, the camera records and sends an alert.
- Only important footage is uploaded to the cloud.

Advantages:

- Instant alerts for faster action
- Reduces unnecessary cloud usage and costs
- Local storage for privacy and compliance

Challenges:

- Requires smart hardware
- Needs periodic updates to maintain AI accuracy
- Potential for physical tampering

Smart Agriculture (Soil Monitoring) Use Case: Automate farming practices using real-time soil data.

Workflow:

- Sensors continuously monitor soil moisture, pH, and temperature.
- Edge gateway analyzes if the moisture is below threshold.
- Automatically turns on water pumps and controls drip irrigation.
- Uploads daily or weekly summaries to cloud dashboards.

Advantages:

- Reduces water usage and energy cost
- Increases crop yield through precision farming
- Supports offline farming in rural areas

Challenges:

- Edge devices must withstand harsh environments
- Initial cost of equipment and installation
- Requires farmer training



Autonomous Vehicles Use Case: Perform safe and intelligent driving decisions in real time.

Workflow:

- Sensors like cameras, radar, and LIDAR scan the surroundings
- Edge processor onboard processes data within milliseconds
- Decision-making algorithms adjust speed, steer, or brake
- Logs or summaries of trips are synced with the cloud later for AI training

Advantages:

- Ultra-low latency for accident prevention
- Doesn't rely on network for core functions
- Enables real-time learning and control

Challenges:

- Expensive high-performance processors
- Software complexity and high testing requirements
- Power consumption management

General Advantages of Edge Computing

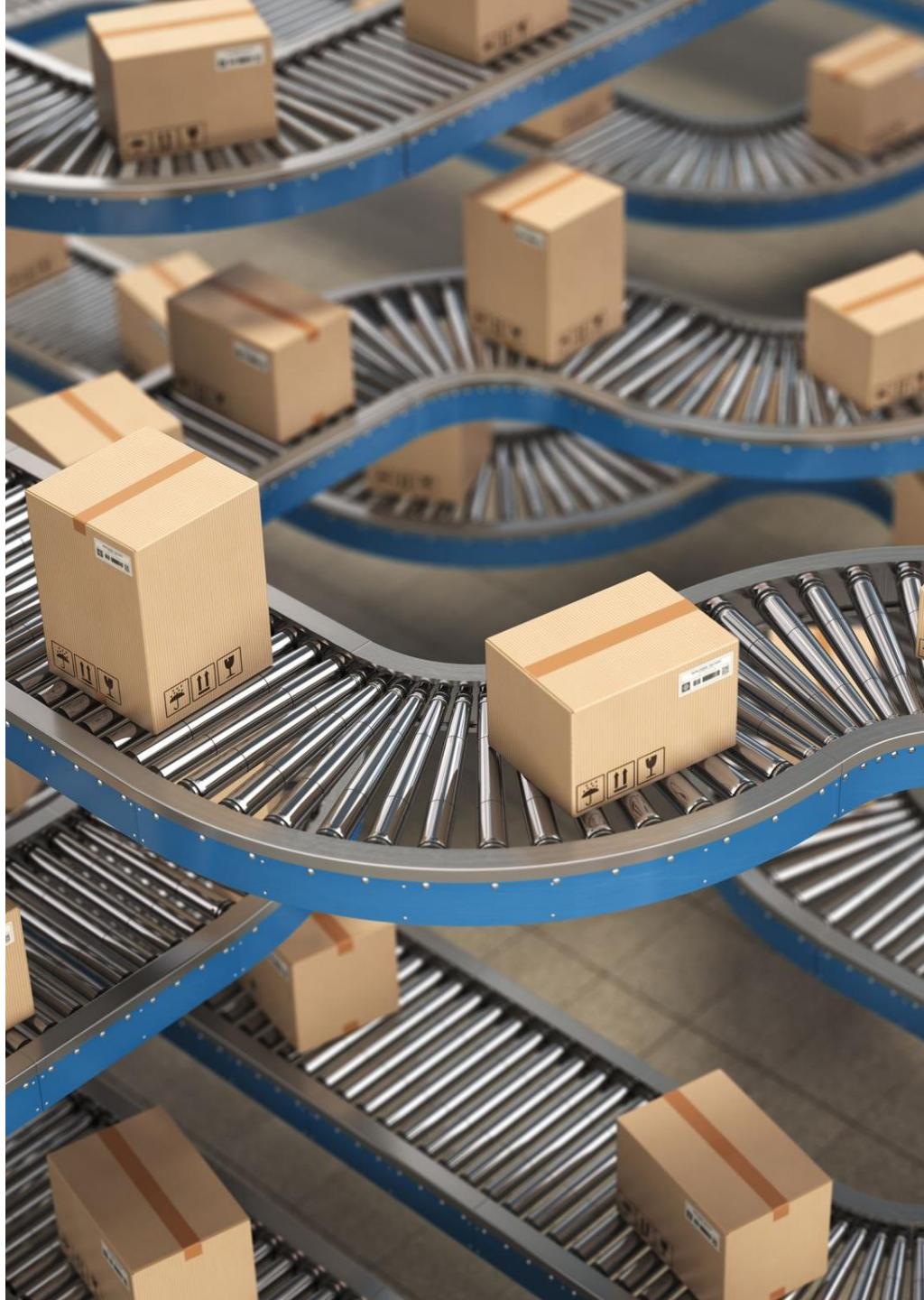
- **Speed:** Processes data faster, ideal for real-time applications
- **Reduced Costs:** Less cloud processing and data storage
- **Scalability:** Edge devices can be added without overloading central servers
- **Privacy Compliance:** Keeps data local in sensitive applications like healthcare
- **Reliable Operations:** Keeps systems running even during internet disruptions

Disadvantages of Edge Computing

- **Limited Resources:** Smaller devices can't handle complex tasks like cloud servers
- **Management Overhead:** Managing and updating thousands of edge nodes can be difficult
- **Security Risks:** Physical access can compromise devices
- **Cost of Deployment:** Setting up edge devices at scale can be expensive initially
- **Data Fragmentation:** Data spread across many devices can make analytics harder

When to Use Edge Computing?

- **Time-sensitive applications:** Industrial robots, self-driving vehicles
- **Bandwidth-constrained environments:** Remote areas, rural deployments
- **Intermittent connectivity:** Ships, field research stations, remote mining sites
- **Sensitive Data Processing:** Health monitors, financial applications
- **Decision Tip:** Use edge computing when decisions must be fast, local, or secure.



Future of Edge Computing

- **5G Integration:** Faster, more reliable networks will enhance edge capabilities
- **AI at the Edge:** Running machine learning models locally to improve predictions
- **Edge in Healthcare:** Real-time patient monitoring and diagnostics
- **Smart Cities:** Traffic control, waste management, and surveillance
- **Hybrid Cloud Models:** Mixing cloud and edge for optimal results





EPS10

YOUR TEXT HERE

Place for your text here. Place for
your text here. Place for your text
here. Place for your text here.

Summary

- Edge computing complements cloud by enabling localized data processing
- Reduces latency and cloud dependence
- Essential for IoT, real-time automation, and mission-critical systems
- Requires investment and planning, but offers high value in many sectors

What is Serverless Architecture?

Definition:

Serverless architecture is a cloud-computing model where cloud service providers automatically handle the infrastructure, allowing developers to focus only on writing code without managing servers.

- **Key Concept:** Serverless doesn't mean there are no servers. It means the responsibility for server management (scaling, provisioning, etc.) is shifted to the cloud provider.
- **Context:** Serverless allows developers to write functions and deploy them without worrying about the underlying infrastructure.



Key Components of Serverless Architecture



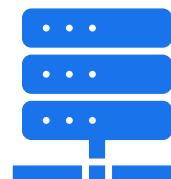
Functions as a Service (FaaS):
Small units of code that are executed
in response to events.



Backend as a Service (BaaS):
Managed backend services, like
databases, authentication, and file
storage.



Event-driven:
Triggered by events, such as HTTP
requests, file uploads, or database
changes.



Managed Services:
Infrastructure and scaling are handled
by the cloud provider (e.g., AWS
Lambda, Google Cloud Functions,
Azure Functions).

How Does Serverless Work?

Event Triggers:
Events like HTTP requests, file uploads, or database changes occur.

Function Invocation:
The serverless provider runs the function in response to the event.

Execution:
The function runs for a short time, completing a task like processing data or sending an email.

Scaling:
Serverless architectures automatically scale based on traffic, with no manual intervention.

Pay-per-Use:
You only pay for the execution time, reducing costs.

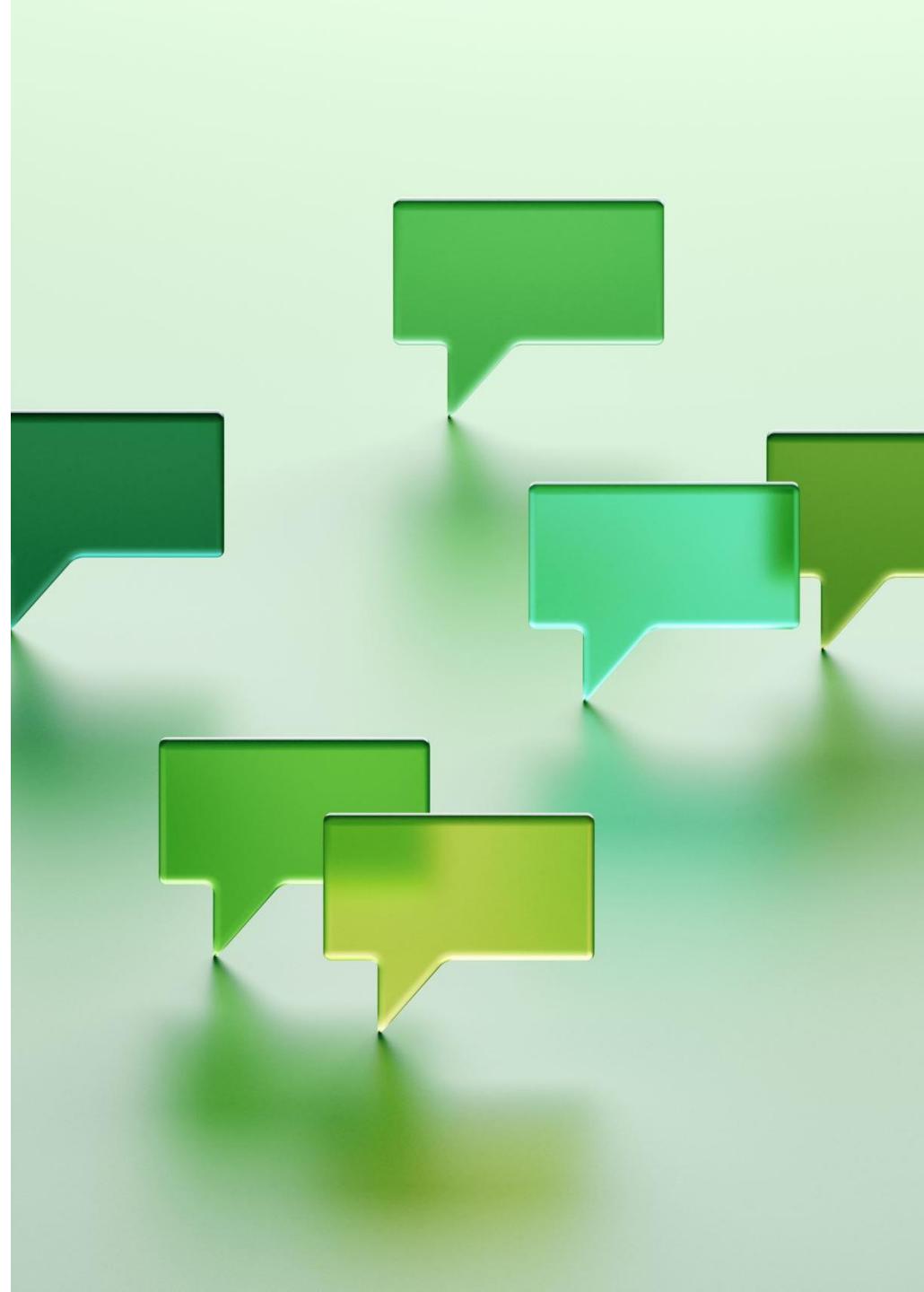


Real-World Examples of Serverless Architecture

- **Web Applications:**
Using cloud functions to handle user authentication, authorization, and request handling.
- **Data Processing:**
Processing large amounts of data with serverless functions triggered by data events.
- **Chatbots:**
Deploying chatbots on platforms like Slack or Facebook Messenger without managing the servers.

Additional Examples:

- Real-time image processing
- IoT data processing
- Serverless REST APIs



Example 1 – Web Applications with Serverless

Use Case:

Running dynamic web applications without managing the server infrastructure.

Workflow:

- User requests a resource (e.g., a webpage).
- Serverless function handles the request, fetching data or interacting with a database.
- Response is sent back to the user.
- Serverless architecture auto-scales based on traffic demand.

Advantages:

- No server management or provisioning
- Scales automatically with usage
- Cost-effective for unpredictable workloads

Challenges:

- Cold start latency for infrequent requests
- Limited execution time for functions



Flow of a Serverless Web Application Handling a User Request

- **User Interaction** – The user submits a request (e.g., clicking a button, filling out a form, or sending a message).
- **API Gateway Receives the Request** – The request is sent to an **API Gateway**, which acts as an entry point and routes it to the appropriate backend service.
- **Triggering a Serverless Function** – The API Gateway invokes a **serverless function** (e.g., AWS Lambda, Google Cloud Functions, Azure Functions) to process the request.
- **Processing & Business Logic Execution** – The function executes the necessary logic, such as validating input, retrieving data, or performing calculations.
- **Database Query (If Needed)** – If the request requires data retrieval or storage, the function interacts with a **serverless database** (e.g., Firebase Firestore, DynamoDB).
- **Response Generation** – The function prepares a response based on the processed data.
- **Returning the Response to the User** – The API Gateway sends the response back to the frontend, displaying the requested information.
- **Logging & Monitoring** – The system logs the request and response for analytics and debugging.

Data Processing with Serverless

Use Case:

Real-time data processing for large datasets or streaming data.

Workflow:

- Data is uploaded or changed in storage (e.g., S3 bucket).
- Serverless function is triggered to process the data (e.g., resize images or filter logs).
- Processed data is stored or further processed.

Advantages:

- Efficient for batch and real-time processing
- Scales automatically with data volume
- Easy to integrate with cloud storage

Challenges:

- Limits on execution time and memory usage
- Complex workflows may require multiple functions

Chatbots in Serverless

Use Case:

Deploying intelligent chatbots to respond to customer queries without managing servers.

Workflow:

- User sends a message to the chatbot platform.
- Serverless function processes the request and fetches relevant information.
- Chatbot sends a reply to the user.

Advantages:

- Easily scalable as user volume grows
- Reduces infrastructure management overhead
- Fast to deploy and iterate

Challenges:

- Potential for cold start delays
- Requires integration with other cloud services for full functionality

Flow of a Serverless Chatbot

- **User Sends a Message** – A user types a query (e.g., "*What is my order status?*") in a chat interface like **Slack, Facebook Messenger, or a website chatbot**.
- **API Gateway Receives the Request** – The message is sent to an **API Gateway**, which routes it to the chatbot's backend.
- **Triggering a Serverless Function** – A **serverless function** (e.g., AWS Lambda, Google Cloud Functions, Azure Functions) is activated to process the request.
- **Natural Language Processing (NLP)** – The function sends the message to an **AI-powered NLP service** (e.g., Dialogflow, Microsoft Bot Framework) to understand the user's intent.
- **Database Query (If Needed)** – If the chatbot needs additional information (e.g., order details), it queries a **serverless database** like Firebase or DynamoDB.
- **Response Generation** – The chatbot formulates a response based on the processed data.
- **Message Sent Back to User** – The chatbot replies with the relevant information (e.g., "*Your order is out for delivery and will arrive by 5 PM.*").
- **Logging & Monitoring** – The interaction is logged for future improvements, and analytics tools track chatbot performance.

Differences Between Serverless and PaaS

Scalability	Pricing Model	Startup Time	Deployment & Management	Use Cases
<ul style="list-style-type: none">• Serverless: Automatically scales instantly based on demand. Functions run only when triggered.• PaaS: Requires manual scaling or predefined configurations to handle increased traffic.	<ul style="list-style-type: none">• Serverless: Pay only for execution time—no charges when idle.• PaaS: Pay for reserved resources, even if they are not fully utilized.	<ul style="list-style-type: none">• Serverless: Functions start on demand, which may cause cold start latency.• PaaS: Applications run continuously, avoiding cold starts.	<ul style="list-style-type: none">• Serverless: No need to manage servers—just deploy functions.• PaaS: Developers still manage runtime environments and configurations.	<ul style="list-style-type: none">• Serverless: Best for event-driven applications, chatbots, APIs, and microservices.• PaaS: Ideal for web applications, databases, and enterprise software.

General Advantages of Serverless Architecture

Scalability: Automatically scales based on demand.

Cost-Effective: Pay only for the actual execution time.

Faster Development: Focus on writing business logic, not infrastructure.

Reduced Overhead: Cloud provider handles scaling, monitoring, and maintenance.

Disadvantages of Serverless Architecture

Cold Starts: Initial delay when a function is invoked after a period of inactivity.

Limited Execution Time: Functions often have a time limit (e.g., AWS Lambda has a 15-minute timeout).

Vendor Lock-in: Tied to specific cloud platforms and services.

Debugging & Monitoring: Can be harder to debug and monitor without dedicated tools.

When to Use Serverless Architecture?

- **Event-driven applications:** Applications triggered by events like file uploads, HTTP requests, etc.
- **Microservices architecture:** When breaking down applications into smaller, independently scalable units.
- **Cost-sensitive, variable workloads:** Applications with unpredictable traffic or periodic workloads.
- **Decision Tip:**
Ideal for applications with variable usage patterns and limited infrastructure management



Future of Serverless Architecture

More Cloud Platforms: More providers adopting serverless offerings.

Serverless Databases: Serverless architecture for data storage and queries.

Serverless AI/ML: Deploying machine learning models with serverless architectures.

Edge Computing Integration: Serverless functions running at the edge, enabling real-time responses in IoT applications.

Summary

Serverless architectures provide a flexible, scalable approach to application development without managing servers.

They are cost-effective, scalable, and easy to implement for event-driven use cases.

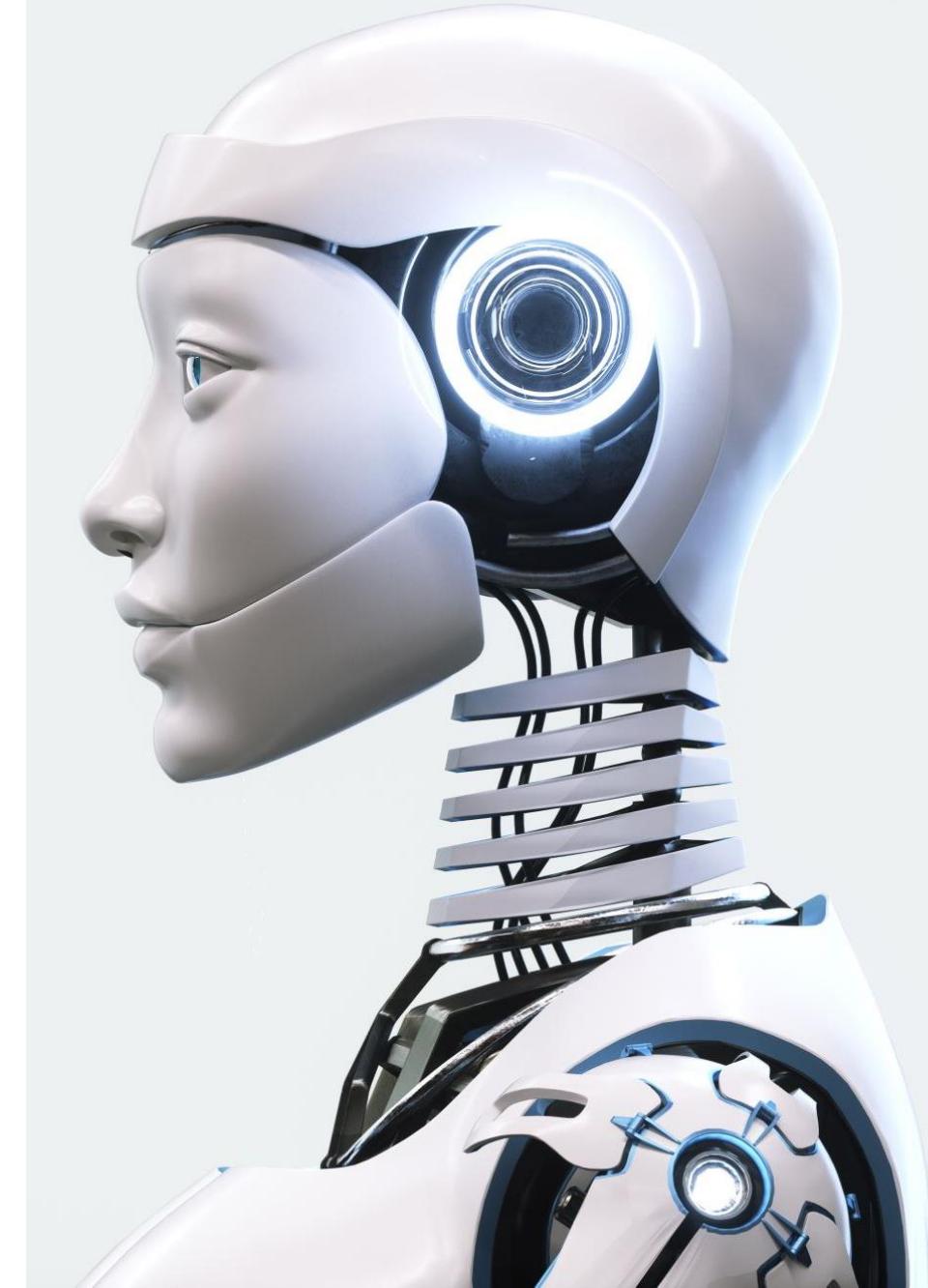
Ideal for real-time applications, data processing, and microservices.

There are challenges like cold starts and vendor lock-in.



What is Artificial Intelligence (AI)?

- **Definition:**
Artificial Intelligence refers to the simulation of human intelligence in machines designed to think, learn, and make decisions.
- **Types of AI:**
 - **Narrow AI:** Specialized in one task (e.g., image recognition, voice assistants).
 - **General AI:** Hypothetical AI capable of understanding any task that a human can.
- **Goal:**
AI aims to create systems that can perform tasks that typically require human intelligence.



What is Big Data?

- **Definition:**

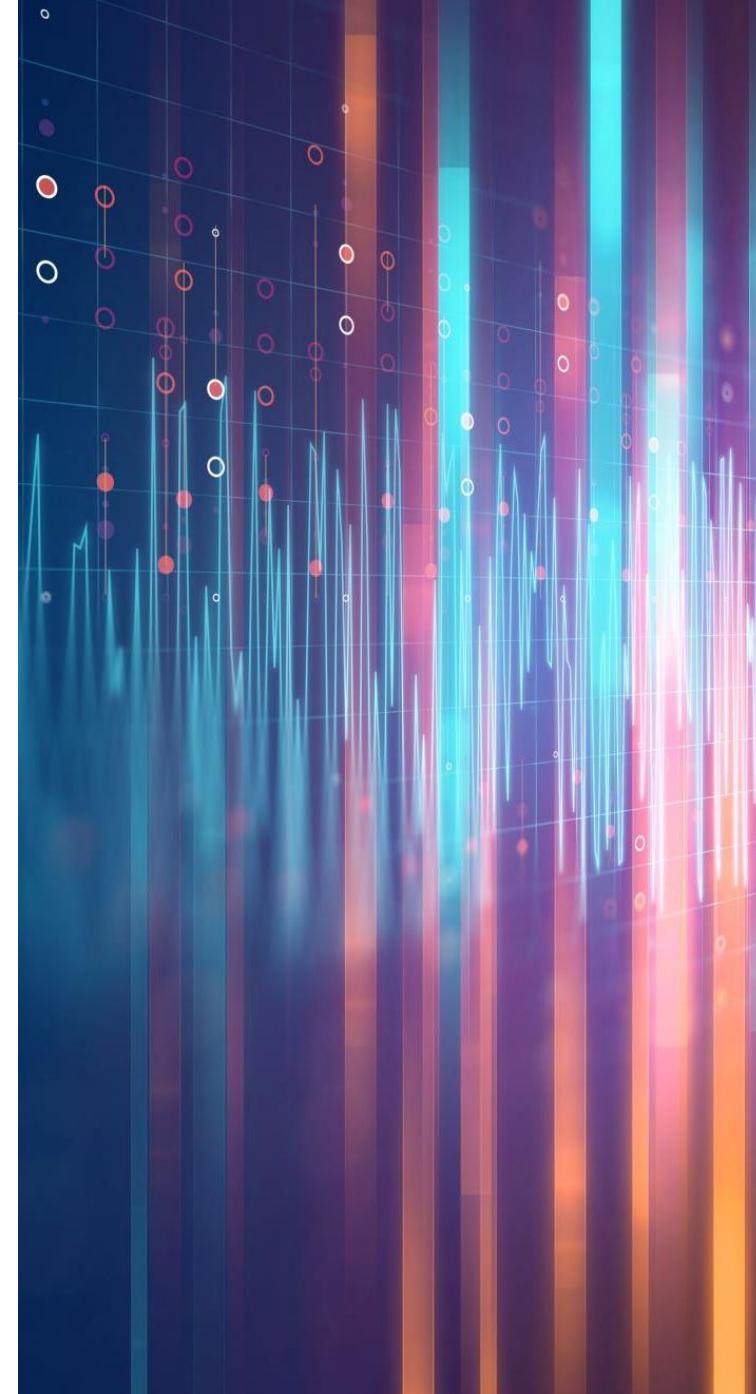
Big Data refers to extremely large datasets that can be analyzed computationally to reveal patterns, trends, and associations.

- **Characteristics of Big Data (The 3 Vs):**

- **Volume:** Large amounts of data.
- **Velocity:** Speed at which data is generated and processed.
- **Variety:** Different forms of data (structured, unstructured, semi-structured).

- **Big Data in Action:**

Big Data is used in various industries for predictive analytics, customer insights, and more.



How AI and Big Data Work Together

AI and Big Data: Big Data is the fuel that drives AI applications. AI models need large volumes of data for training, and Big Data analytics processes vast datasets.

AI in Big Data Processing: AI models analyze, classify, and predict patterns in Big Data.

Big Data in AI: Big Data platforms store, manage, and organize large datasets that AI models rely on for predictions.

Example:

AI models predict consumer behavior based on Big Data from social media, transactions, and browsing history.



Benefits of AI and Big Data in the Cloud

Scalability: Cloud computing offers unlimited resources, allowing AI and Big Data applications to scale efficiently.

Cost Efficiency: Pay-as-you-go pricing model for cloud services reduces the cost of AI and Big Data operations.

Accessibility: Data and AI models are accessible from anywhere, improving collaboration.

High-performance: Cloud platforms offer the computing power required for processing large datasets and running complex AI algorithms.

AI and Big Data Use Case 1 – Healthcare

- **Use Case:**
AI and Big Data are revolutionizing healthcare by providing insights into patient data for diagnostics and treatment.
- **Workflow:**
 - **Data Collection:** Health records, medical imaging, and sensor data.
 - **Big Data Processing:** Process large volumes of data to identify trends.
 - **AI Model Training:** Machine learning models are trained to identify diseases, predict patient outcomes, and personalize treatments.
 - **Cloud Deployment:** AI and Big Data are processed on cloud platforms, enabling easy access for healthcare professionals.
- **Benefits:**
 - Faster diagnostics
 - Personalized treatments
 - Improved patient outcomes



AI and Big Data Use Case 2 – Financial Services

- **Use Case:**

Financial institutions use AI and Big Data to predict market trends, detect fraud, and enhance customer experiences.

- **Workflow:**

- Data Collection: Transaction records, customer behavior data, market trends.
- Big Data Processing: Process large volumes of financial data to detect anomalies and trends.
- AI Model Training: Machine learning algorithms are used for fraud detection, risk analysis, and investment predictions.
- Cloud Deployment: AI models are deployed on cloud platforms for scalability and real-time analysis.

- **Benefits:**

- Enhanced fraud detection
- Optimized financial predictions
- Improved customer service



AI and Big Data Use Case 3 – Retail

Use Case:

Retailers use AI and Big Data to analyze consumer behavior and optimize inventory management.

Workflow:

- **Data Collection:** Customer browsing history, purchase data, and social media interactions.
- **Big Data Processing:** Process large volumes of customer and sales data to identify buying patterns.
- **AI Model Training:** Machine learning models predict consumer behavior and optimize inventory levels.
- **Cloud Deployment:** AI models are deployed in the cloud to scale and handle high volumes of data.

Benefits:

- Improved customer targeting
- Reduced inventory costs
- Personalized shopping experiences



Challenges of AI and Big Data in the Cloud

- **Data Privacy:** Handling sensitive data in compliance with privacy regulations (e.g., GDPR).
- **Data Quality:** Big Data is often unstructured, which makes it difficult to process and analyze.
- **Security Risks:** Storing large amounts of data on cloud platforms increases the risk of cyber-attacks.
- **Skill Gap:** The need for skilled professionals to manage and analyze Big Data and AI applications in the cloud.

Key Cloud Providers for AI and Big Data

- **Amazon Web Services (AWS):** Offers machine learning services (SageMaker), big data processing (EMR), and storage (S3).
- **Microsoft Azure:** Provides AI tools (Azure Machine Learning) and big data services (HDInsight, Azure Synapse).
- **Google Cloud Platform (GCP):** Offers AI tools (AI Platform), big data processing (BigQuery), and storage (Cloud Storage).
- **Comparative Advantage:**
Different providers offer varying capabilities in terms of scalability, pricing, and specific AI and Big Data services.



Use Case

Imagine a company that sells electronics online. Every day, it receives:

- Thousands of **product reviews**
- Comments and mentions on **Twitter or Facebook**
- **Customer support emails**

The company wants to:

- Detect common issues
- Understand customer satisfaction trends
- Adjust products or services based on real feedback

-
- **Type of Data:**
 - **Unstructured Text:** Free-form text that doesn't follow a fixed format.
 - E.g., "Battery drains too fast!" or "Amazing product. Very happy with the camera quality."

AI Techniques Used:

- **Sentiment Analysis:** Identifies whether the tone of the text is **positive**, **negative**, or **neutral**.
- **Topic Modeling:** Groups similar keywords or phrases to identify main topics people talk about (e.g., "battery", "camera", "price").

Work flow

- **Ingest Data**

User reviews and social media posts are stored in Amazon S3.

- **Clean and Prepare**

AWS Glue removes unwanted characters, parses date formats, and extracts review texts.

- **Analyze with NLP**

Cleaned data is sent to Amazon Comprehend, which labels each review:

- Sentiment: Positive / Negative / Neutral
- Extracts topics like “battery” or “customer service”

- **Store and Query Results**

The enriched data is stored back in S3 or a database, and Athena is used to query it:

- "Show all reviews about battery that are negative"

- **Visualize Trends**

QuickSight displays results in charts and graphs for business teams to act on insights.

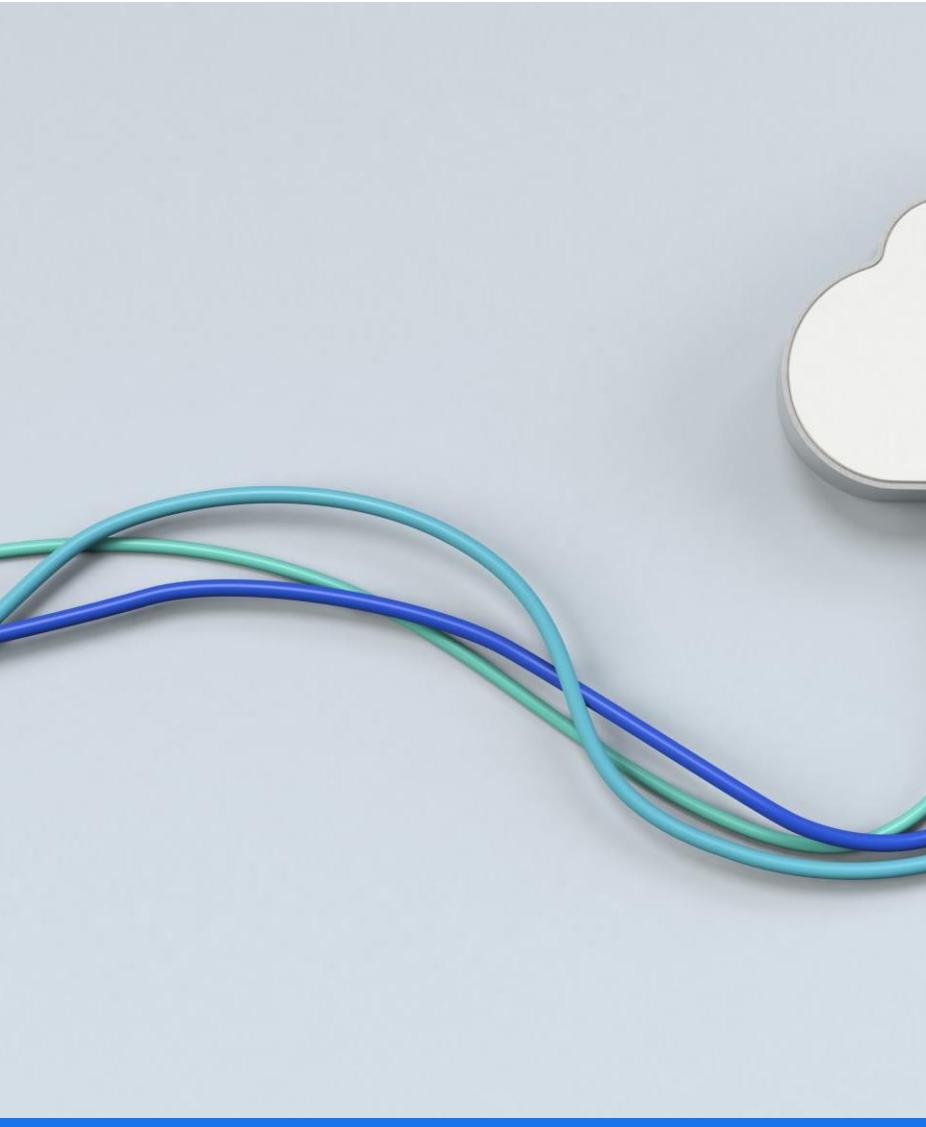


Advantages of Using Cloud for AI and Big Data

- **Cost Efficiency:** No need to maintain expensive hardware and infrastructure.
- **Scalability:** Easily scale to handle increasing amounts of data and computational demands.
- **Flexibility:** Cloud platforms offer a wide range of tools for AI and Big Data applications.
- **Speed and Performance:** Cloud computing provides high-performance computing to run complex AI algorithms and process large datasets.

Future of AI and Big Data in the Cloud

- **AI Democratization:** Cloud platforms make AI accessible to small and medium businesses.
- **Automated Data Processing:** Integration of AI with Big Data will lead to automated data preprocessing, reducing the manual workload.
- **Edge Computing:** Combining AI, Big Data, and Edge Computing will bring real-time data analytics closer to the source.
- **Quantum Computing:** Future integration of quantum computing with cloud-based AI and Big Data solutions will drive new capabilities.



Summary

- AI and Big Data in the cloud enable scalable, efficient, and cost-effective solutions for industries like healthcare, finance, and retail.
- Cloud platforms offer the resources to process massive datasets and run AI models at scale.
- Challenges include data privacy, security, and the need for skilled professionals.