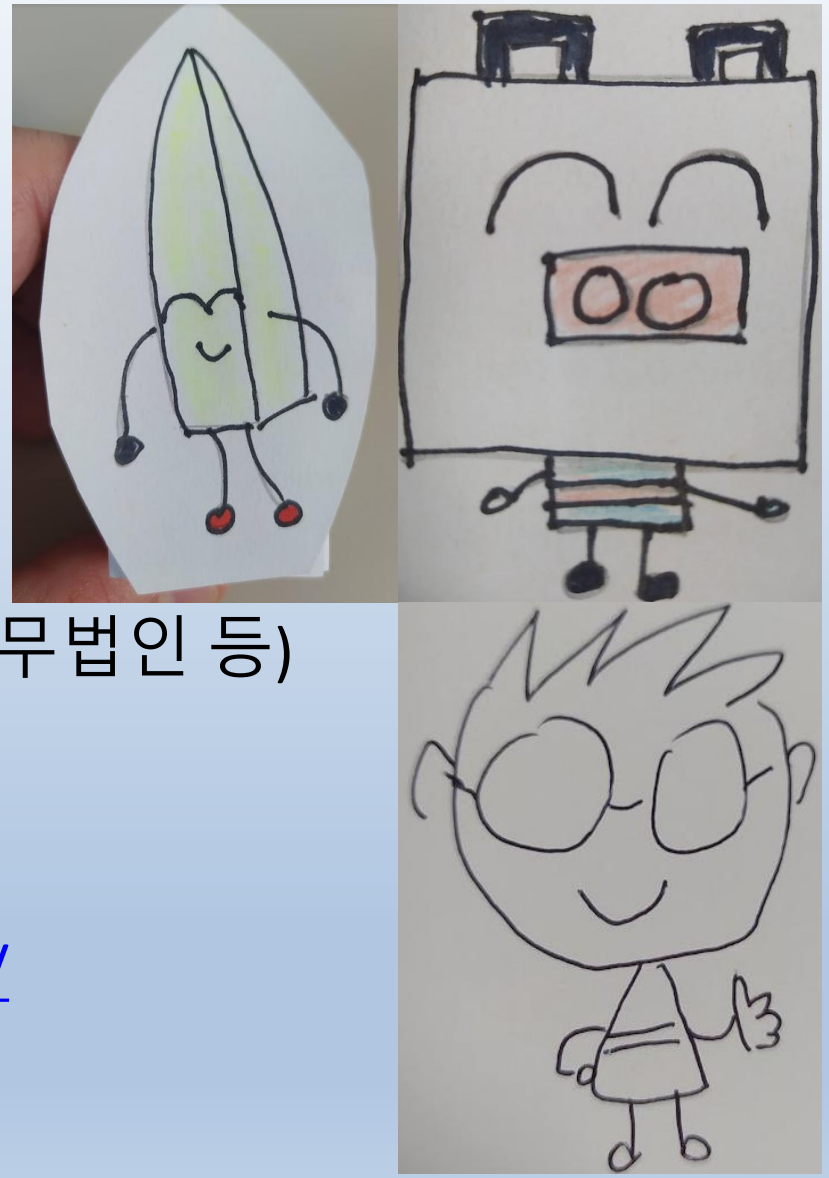
Cloud Computing Overview

High-Level & Practical Insights

AI Big Data Consultant & Financial Advisor

신경준 info@easytoast.net
https://finance.easytoast.net

Self Introduction



학력 및 경력

- 명지대학교 전자공학과 졸업 (2002)
- 하우리 2002 ~ 2004
- YNK Games 2004 ~ 2005
- 안철수 연구소 2005 ~ 2012
- 삼성 SDS 2012 ~ 2015
- SSJ 컨설팅 2016 ~ 현재 (고영테크놀로지, HLK 세무법인 등)

현재 주요 업무

- AI 업무 자동화 (RPA) : https://easytoast.net/
- AI Financial Service : https://finance.easytoast.net/
- YouTube 투자하는 프로그래머
- 대학생, 20~30대 젊은 사람 경제 투자 조언

이름: 신정근
(갱즈니)

특징: 숫자 중독자
숫자는 거짓말 하지
않는다

좋아하는 애니
The simpsons
[redacted] 미래를
보는 걸 알아서

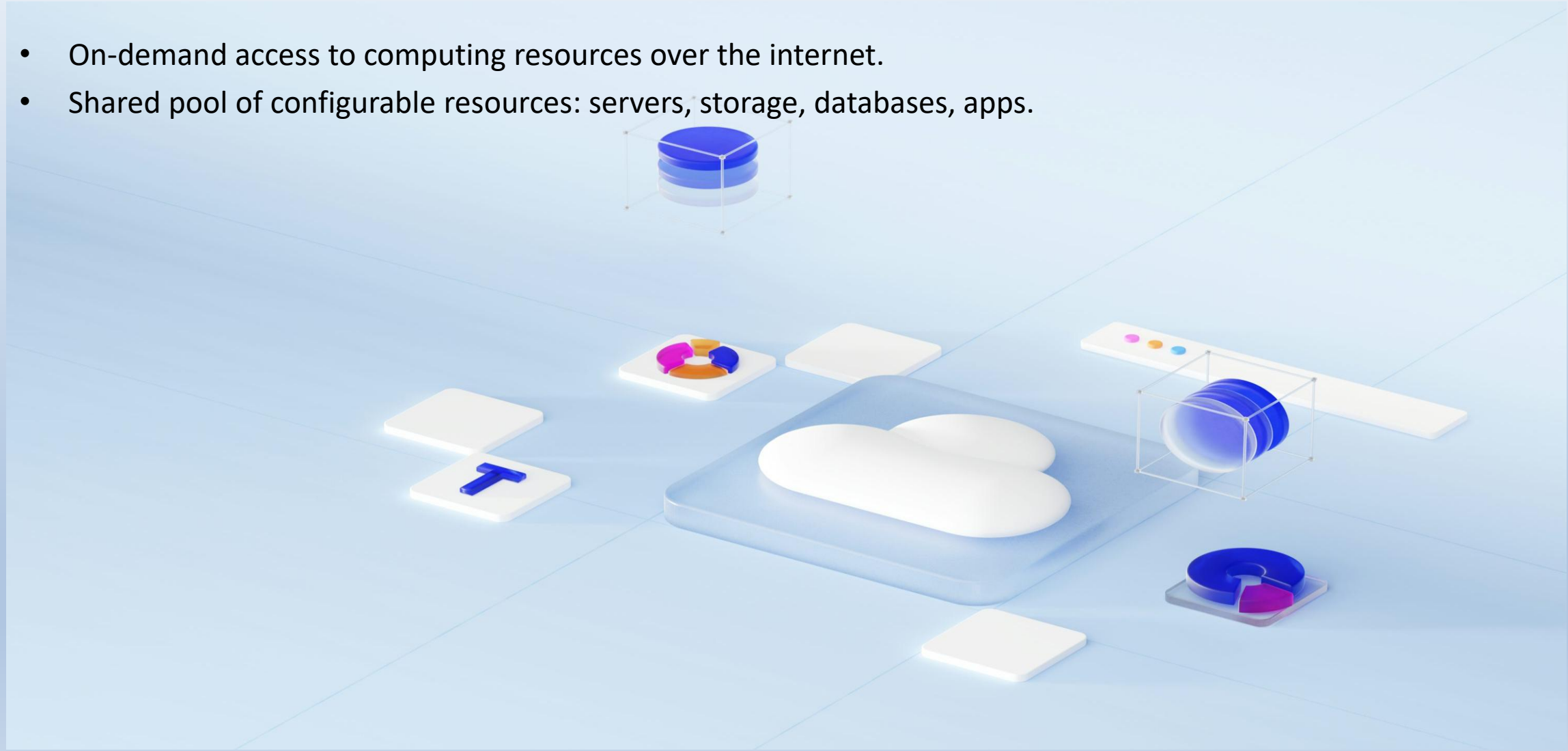
나의 꿈은
1) 경제적 자유
2) 강전어 아닌 내마음
본질을 볼수 있는 사람.

싫어요
- 감정에 휘둘릴때
- 근거 없이 주장하는 사람
이유

좋아요
자연, 푸른하늘
푸른 바다, 자전거

What is Cloud Computing?

- On-demand access to computing resources over the internet.
- Shared pool of configurable resources: servers, storage, databases, apps.



The Core Concept Cloud (1)

- Imagine you need a powerful computer to run a complex program for just a few hours. Buying one would be expensive, and it would sit idle most of the time. Now, imagine a library where you can rent a powerful computer for exactly as long as you need it, and you only pay for that time.
- Cloud computing is a similar idea, but instead of a physical computer, you are renting computing power, storage, and software over the internet from a massive, shared pool of resources. This is the "cloud."

The Core Concept Cloud (2)

On-demand access to computing resources over the internet.

- **What this means:** You don't have to buy and install physical hardware. You can get what you need instantly, with just a few clicks. Think about when you need to access a document. Do you have to buy a new USB stick and physically move it? No, you can just use Google Drive or Dropbox. Cloud computing is the same, but for much more powerful things like servers, databases, and software.
- **For the students:** In the past, if a company wanted to launch a new website, they would have to buy servers, set up a data center, and install a cooling system. This could take months and cost a fortune. With cloud computing, they can have a new server up and running in minutes, and a complete website ready to go in a few hours.

The Core Concept Cloud (3)

- **Shared pool of configurable resources: servers, storage, databases, apps.**
- **What this means:** Cloud providers have huge data centers filled with thousands of servers and storage devices. They manage all the hardware and software. Instead of each customer having their own dedicated physical server, they share a virtualized pool of resources.
- **For the students:** Think of this like a public utility. Just as the power company provides electricity to many different homes from a central power plant, a cloud provider provides computing resources to many different companies from a central data center.
- **Configurable:** This is a key word. You don't just get a generic server. You can choose the exact specifications you need:
  - **Servers:** How much memory (RAM) and processing power (CPU) do you need? You can scale up or down as your needs change. => cmd nslookup check.
  - **Storage:** Do you need fast, expensive storage for an active database, or slow, cheap storage for long-term backups? You choose.
  - **Databases:** The cloud provider offers a variety of pre-configured databases that you can launch and use immediately, without the hassle of setting them up yourself.
  - **Apps:** You can use services like Microsoft 365 or Salesforce, where all the software is running on the cloud provider's infrastructure.

Google, AWS, MS, Buy Server or Made by themselves?

- Server Computer Buy Or Self Made(Design)

https://finance.yahoo.com/quote/HPE/

https://finance.easytoast.net/

https://www.quora.com/Which-company-does-Amazon-Web-Services-AWS-buy-most-of-its-servers-hardware-from

Quora

Technology | AWS Cloud Services | Servers (computers) | Hardware Products | Companies | Ar | + 5

Which company does Amazon Web Services(AWS) buy most of its servers/hardware from?

All related (34) Sort Recommended

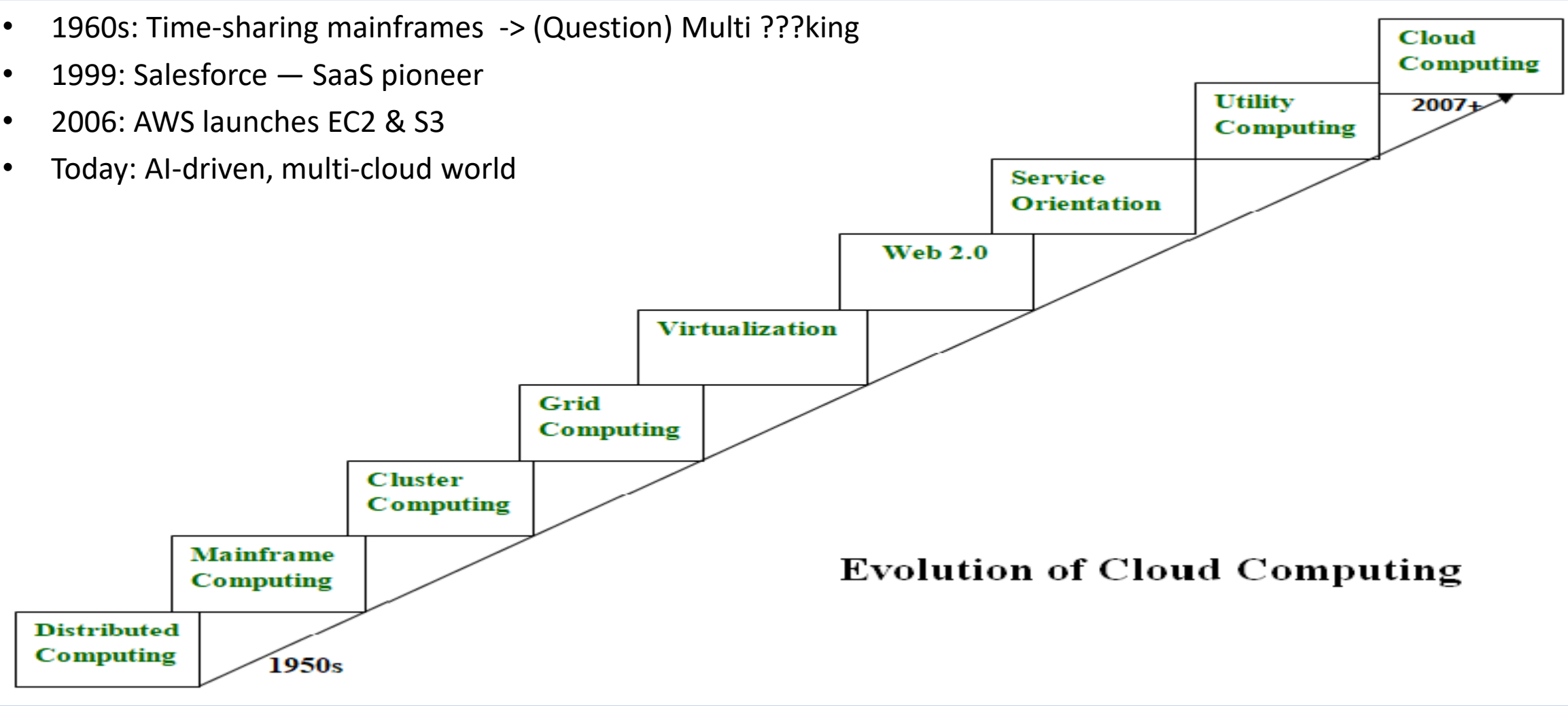
Bernard Golden
Vice President, Cloud Strategy at Capital One · Upvoted by Shane Ryoo, CompE PhD, multiple tech start-up alum and Orly Andico, Solutions Architect at Amazon Web Services (2015-present) · 6y

The hyperscale cloud providers (AWS, Microsoft, Google, Alibaba) do not purchase servers (or, for that matter, network devices) from traditional vendors (e.g., HP). Instead, they have in-house hardware designers design user-specific devices for them, which they then have manufactured to their specification by what are called ODMs (Original Device Manufacturers), typically located in Taiwan. One widely-used ODM is Quanta.

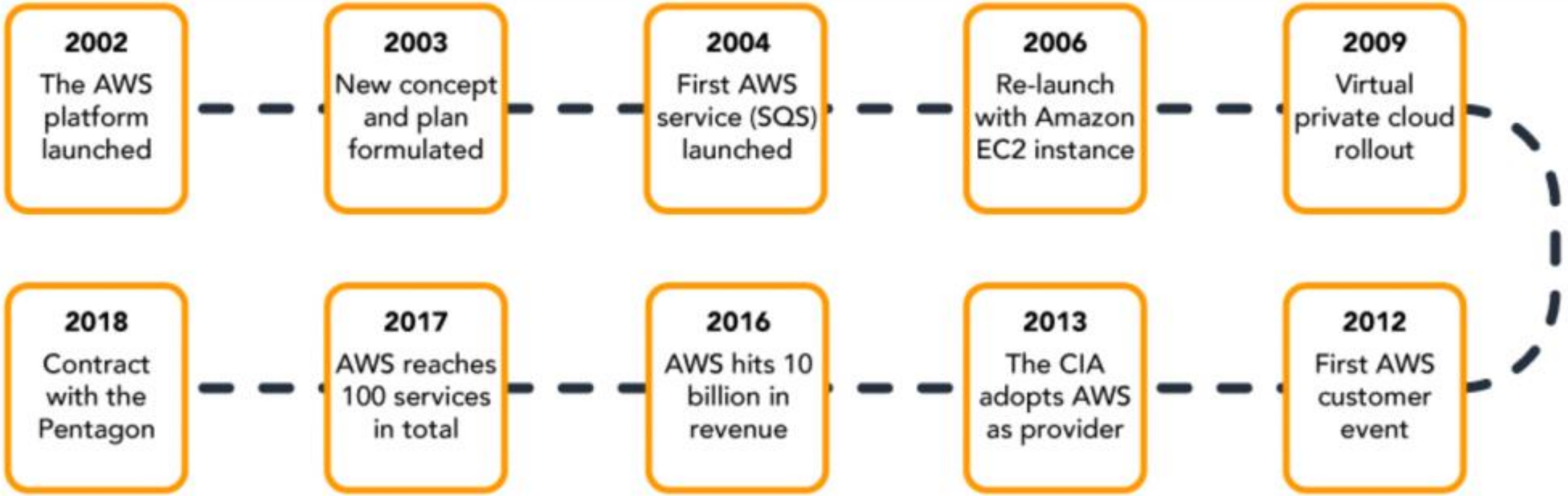
James Hamilton, a senior infrastructure executive at AWS, gave a talk at Reinvent 2016 about how AWS managed its infrastructure. He discusses the company's approach to servers and network devices. Well worth viewing.

Evolution of Cloud

- 1960s: Time-sharing mainframes -> (Question) Multi ???king
- 1999: Salesforce — SaaS pioneer
- 2006: AWS launches EC2 & S3
- Today: AI-driven, multi-cloud world



aws History


2002
The AWS platform launched
2003
New concept and plan formulated
2004
First AWS service (SQS) launched
2006
Re-launch with Amazon EC2 instance
2009
Virtual private cloud rollout
2012
First AWS customer event
2013
The CIA adopts AWS as provider
2016
AWS hits 10 billion in revenue
2017
AWS reaches 100 services in total
2018
Contract with the Pentagon

- Early Foundations (1950s-1990s)
  - Mainframe computing: The concept of centralized computing resources accessible by multiple users via time-sharing emerged in the 1950s.
  - Distributed Systems: This led to systems that shared resources across networks.
  - Grid Computing: Technologies connected geographically dispersed computing resources to tackle complex tasks, though with challenges like inconsistent bandwidth.
  - The Internet: The wide adoption of the internet in the 1990s provided the crucial network infrastructure for web-based services and applications.

- The Rise of Cloud Services (2000s)
  - Virtualization: The introduction of technologies like VMware in 2001 allowed for multiple virtual machines to run on a single physical server, dramatically improving resource efficiency.
  - Infrastructure as a Service (IaaS): Amazon Web Services (AWS) launched in 2006, providing on-demand access to computing infrastructure like servers, storage, and networking, marking the start of modern cloud computing.
  - Platform as a Service (PaaS): In 2009, Heroku introduced PaaS, providing platforms for developers to build and deploy applications without managing the underlying infrastructure.
  - Software as a Service (SaaS): Companies like Salesforce.com began offering applications over the internet, eliminating the need for on-premises installations and maintenance.

- Modern Cloud Evolution (2010s-Present)
  - Containers: Docker's introduction in 2013 revolutionized application deployment by packaging applications and their dependencies together into portable containers.
  - Serverless Computing: This paradigm allows developers to run code without provisioning or managing servers, with cloud providers handling all underlying infrastructure and charging only for actual execution time.
  - Hybrid and Multi-Cloud: Organizations now combine public cloud services with their own on-premises infrastructure (hybrid cloud) or use services from multiple cloud providers (multi-cloud) to diversify and access specialized capabilities.
  - Cloud Native: The Cloud Native Computing Foundation (CNCF), established in 2015, promotes technologies and practices that enable building and running scalable applications in modern, dynamic environments, such as containers and microservices

Key Characteristics

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service (pay-as-you-go)

On-Demand Self-Service

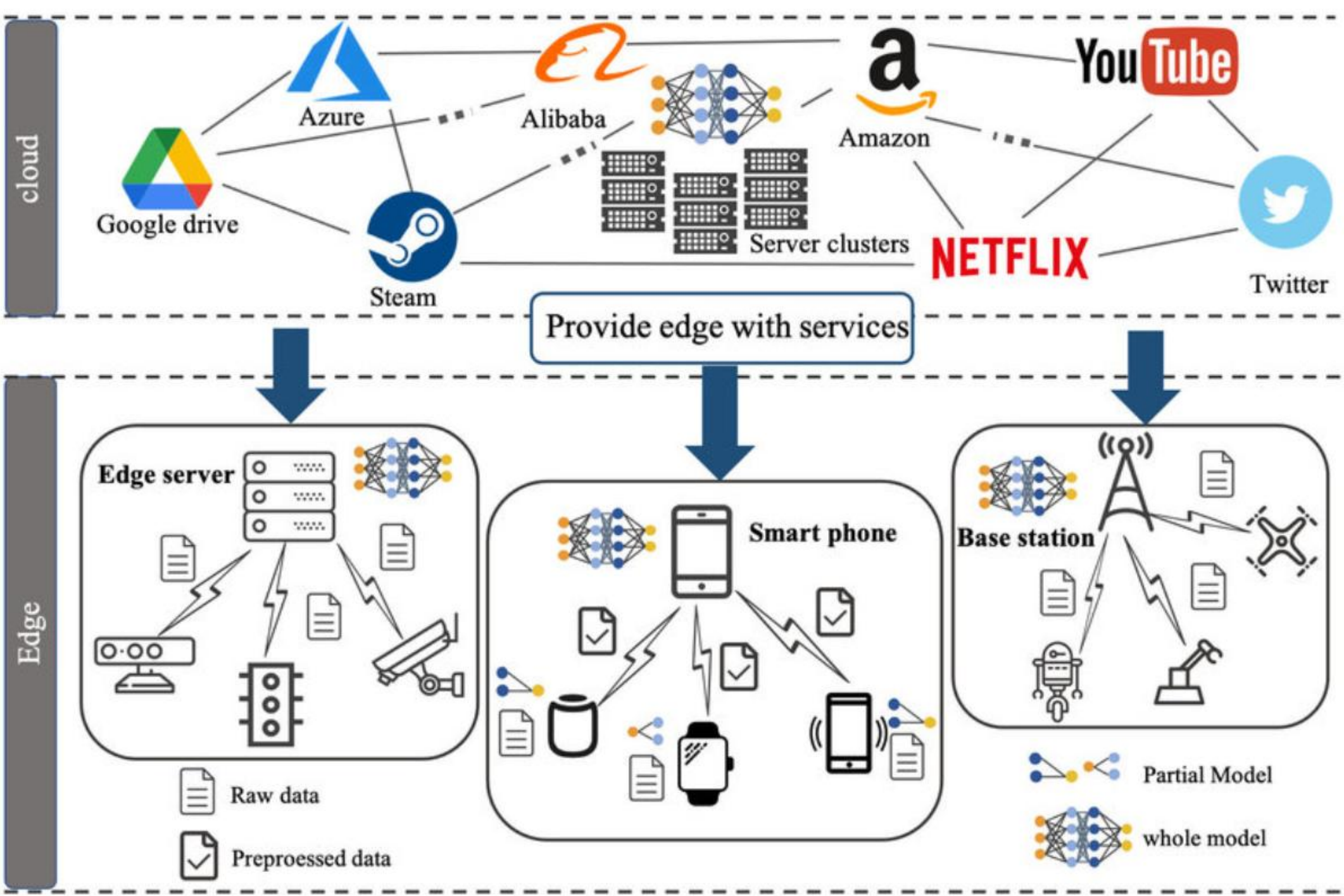
- **Meaning**: Users can provision computing resources (like servers, storage, or databases) automatically without human interaction with the service provider.
- **Why it matters**: No need to wait for IT departments or vendors; users get resources instantly.
- **Example**: In AWS, you can spin up an **EC2 instance** (virtual machine) within minutes from the console.

Broad Network Access

- Meaning: Services are available over the network (usually the Internet) and accessible from diverse devices (PC, laptop, mobile, thin client(?), IoT).
- Why it matters: Accessibility from anywhere improves flexibility and enables remote collaboration.
- Example: Google Drive or Microsoft OneDrive lets you store files in the cloud and access them on your phone, tablet, or desktop seamlessly.

Resource Pooling

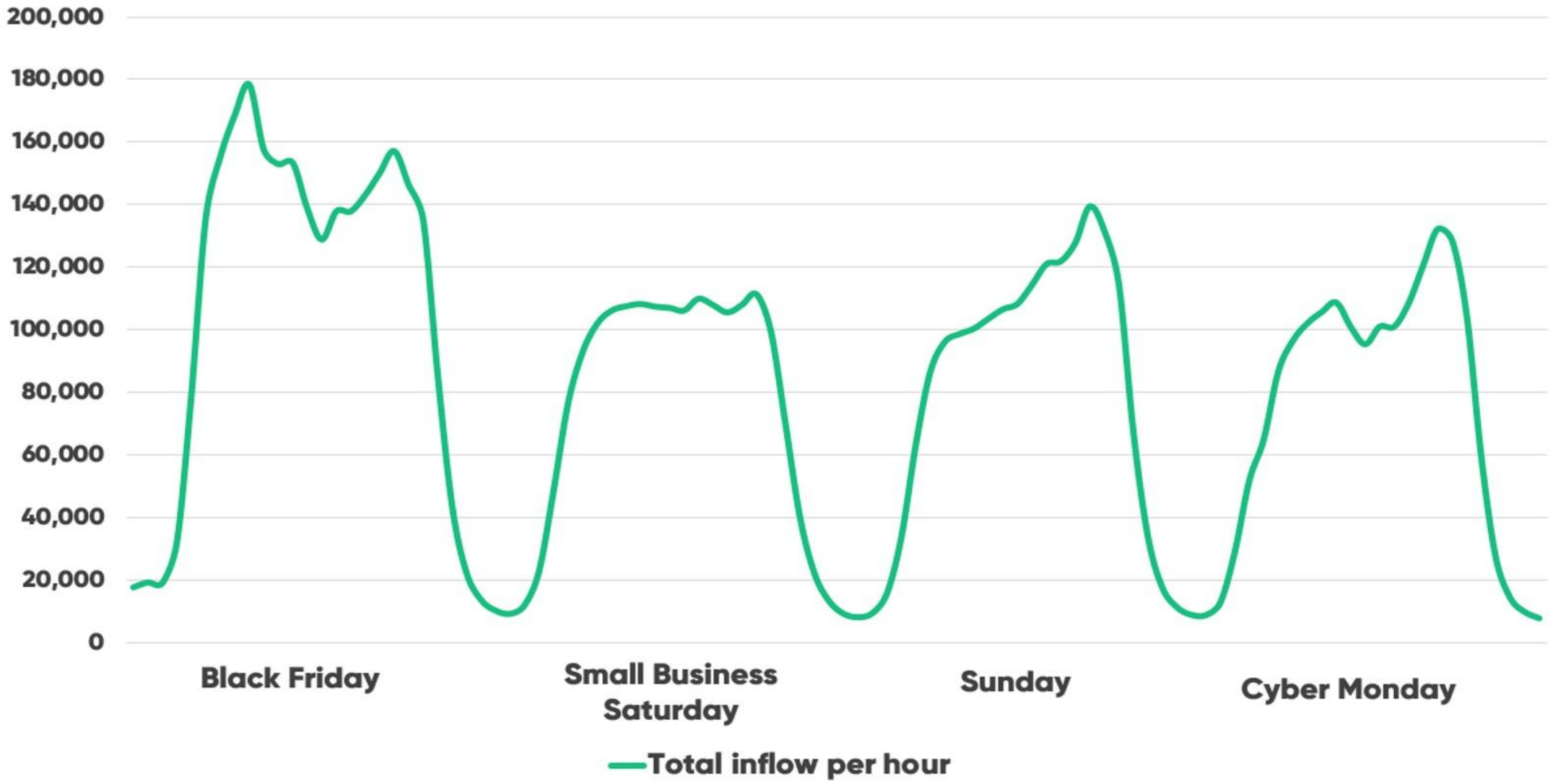
- Meaning: Cloud providers use multi-tenancy — multiple customers share a pool of computing resources (servers, storage, network), dynamically allocated based on demand.
- Why it matters: Efficient use of infrastructure lowers costs and enables scalability.
- Example: AWS data centers serve thousands of companies simultaneously, but resources are logically isolated to ensure security.


cloud
Azure
Alibaba
Amazon
YouTube
Google drive
Server clusters
NETFLIX
Steam
Twitter
Provide edge with services
Edge
Edge server
Smart phone
Base station
Raw data
Preproessed data
Partial Model
whole model

Rapid Elasticity

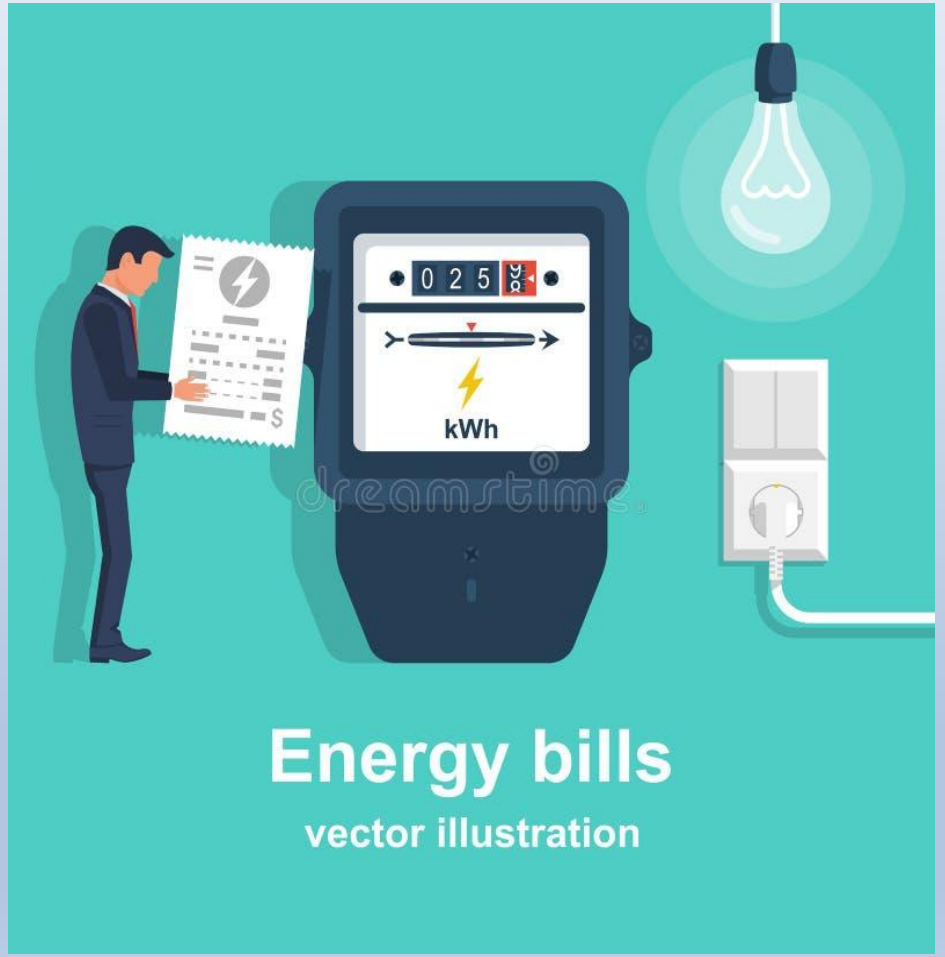
- **Meaning**: Resources can be quickly scaled up or down, automatically or manually, to match workload demand.
- **Why it matters**: Supports unpredictable workloads and prevents over-provisioning or under-provisioning.
- **Example**: An e-commerce website automatically adds more servers during **Black Friday** traffic and scales down afterward.

Holiday season traffic spike: British retailer



Measured Service (Pay-as-You-Go)

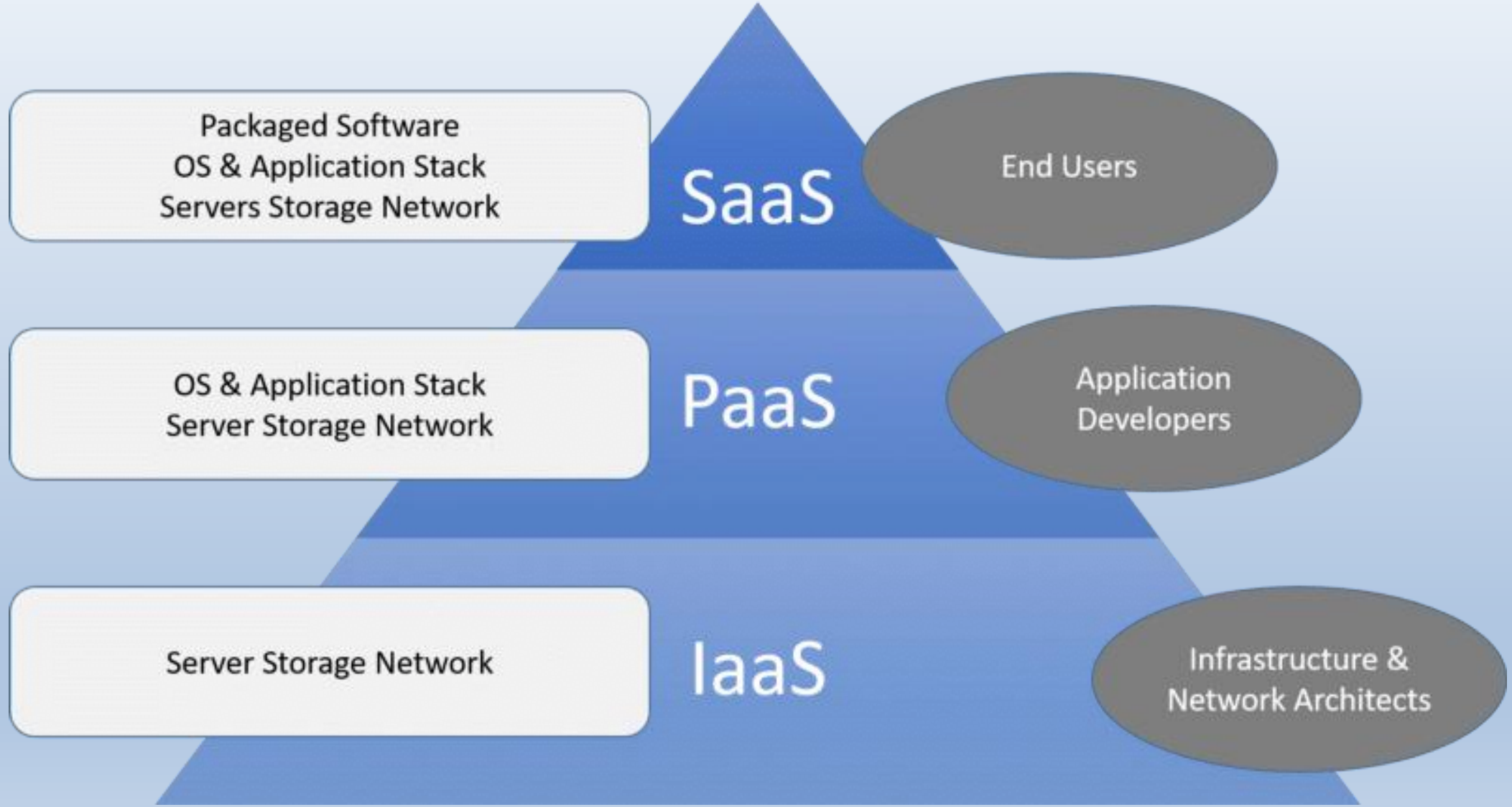
- **Meaning**: Cloud systems automatically control and optimize resource usage by metering — customers pay only for what they consume (compute hours, GB of storage, data transfer).
- **Why it matters**: Shifts from **CapEx (capital expense)** to **OpEx (operational expense)**; very cost-efficient.
- **Example**: AWS S3 charges per **GB stored per month** and per **GB retrieved**, not for unused capacity.



Cloud Service Models (IaaS, PaaS, SaaS)

- IaaS — Infrastructure (VMs, networking, storage) — e.g., AWS EC2
- PaaS — Platforms for developers — e.g., Google App Engine
- SaaS — End-user applications — e.g., Microsoft 365

Cloud Service Models



Infrastructure as a Service (IaaS)

- Think of **IaaS** as renting the most basic building blocks of IT infrastructure. It's like renting an empty apartment. You get the building (servers, storage, networking) and the basic utilities, but you're responsible for everything else: the furniture, decorations, appliances (operating system, middleware, applications).
- With IaaS, a cloud provider gives you on-demand access to fundamental computing resources over the internet. You have the most control and flexibility at this level.
- **You Manage:** The operating system, applications, data, and middleware.
- **The Provider Manages:** The physical hardware, servers, data centers, storage, and networking.
- **Analogy:** You're a home builder. The cloud provider gives you all the raw materials and a plot of land. You build the house exactly how you want it, from the foundation up.
- **Examples:** Amazon Web Services (AWS) EC2, Microsoft Azure Virtual Machines, Google Compute Engine.

Platform as a Service (PaaS)

- **PaaS** is the next layer up the stack. It's like renting a fully furnished apartment, complete with kitchen appliances. You don't have to worry about building the infrastructure or even installing the operating system. The cloud provider gives you a ready-to-use platform where you can develop, run, and manage applications.
- **You Manage:** Your applications and data.
- **The Provider Manages:** The operating system, middleware, and all the underlying infrastructure.
- **Analogy:** You're a chef. The cloud provider gives you a fully equipped kitchen with all the utensils, pots, and pans you could ever need. You just bring your ingredients (code) and start cooking.
- **Examples:** AWS Elastic Beanstalk, Heroku, Google App Engine.

Software as a Service (SaaS)

- **SaaS** is the highest level of service. It's like using a hotel room. Everything is ready for you; you just need to show up and use the service. You don't manage any part of the infrastructure, nor do you install any software. The application is hosted and managed by the cloud provider and delivered over the internet, typically through a web browser.
- **You Manage:** Only your data and user access.
- **The Provider Manages:** Everything. The applications, middleware, operating systems, and all the underlying infrastructure.
- **Analogy:** You're a diner. You walk into a restaurant and order a meal. You don't worry about where the ingredients came from, who cooked the food, or what's happening in the kitchen. You simply enjoy the finished product.
- **Examples:** Google Workspace (Gmail, Docs), Microsoft 365, Salesforce.

Deployment Models

- Public Cloud — AWS, Azure, GCP
- Private Cloud — VMware, OpenStack
- Hybrid Cloud — mix of public & private
- Multi-Cloud — using multiple providers

Public Cloud

- **Public Cloud**
- **Definition**: Cloud resources are owned and operated by third-party providers and delivered over the internet.
- **Key Features**: Shared infrastructure, massive scalability, cost efficiency, pay-as-you-go pricing.
- **Use Cases**: Startups, app development, AI/ML workloads, global web apps.
- **Examples**: AWS, Microsoft Azure, Google Cloud Platform (GCP).
- **Pros**:
  - No upfront infrastructure costs.
  - High scalability and reliability.
  - Wide range of services (compute, storage, databases, AI, etc.).
- **Cons**:
  - Less control over infrastructure.
  - Potential compliance or data sovereignty concerns.

Private Cloud

- **Definition**: Cloud infrastructure is dedicated to a single organization, either hosted on-premises or by a third-party provider.
- **Key Features**: Exclusive use, high control, customizable security and compliance.
- **Use Cases**: Government agencies, banks, healthcare, organizations with strict data requirements.
- **Examples**: VMware vSphere, OpenStack, IBM Cloud Private.
- **Pros**:
  - Full control over infrastructure.
  - Enhanced security and compliance.
  - Customization for organizational needs.
- **Cons**:
  - Expensive to build and maintain.
  - Limited scalability compared to public cloud.

Hybrid Cloud

- **Definition**: A combination of public and private cloud environments, connected for data and application portability.
- **Key Features**: Flexibility to move workloads between private and public clouds as needs change.
- **Use Cases**: Enterprises needing both scalability and regulatory compliance.
- **Examples**:
  - Storing sensitive data in a private cloud, while running front-end apps in AWS or Azure.
  - Cloud bursting: overflow traffic handled by the public cloud.
- **Pros**:
  - Balance between cost efficiency and security.
  - Improved agility and disaster recovery options.
- **Cons**:
  - Complex management and integration. (Finding Human Resource So difficult…)
  - Requires strong networking and orchestration.

Multi-Cloud

- **Definition**: Use of multiple public cloud providers simultaneously (e.g., AWS + Azure + GCP).
- **Key Features**: Avoid vendor lock-in, leverage best-of-breed services, increase resilience.
- **Use Cases**: Enterprises optimizing performance, pricing, and availability across regions/providers.
- **Examples**:
  - Running analytics on GCP's BigQuery while hosting applications in AWS.
  - Using Azure for Microsoft Office 365 integration and AWS for AI workloads.
- **Pros**:
  - No dependency on a single vendor.
  - Access to unique features from different providers. (AI Google Cloud Tensor …)
  - Better redundancy and disaster recovery.
- **Cons**:
  - Increased complexity in governance, security, and monitoring.
  - Higher training and management overhead.

Summary

- **Public Cloud** = scalable, cost-efficient, general-purpose.
- **Private Cloud** = controlled, secure, for sensitive data.
- **Hybrid Cloud** = best of both worlds, but complex.
- **Multi-Cloud** = flexibility, no lock-in, but requires strong management.

Serverless / FaaS (Function-as-a-Service)

- Event-driven computing (e.g., AWS Lambda)
- No server management
- Pay only per execution

Major Cloud Providers

- AWS — market leader, broad services
- Microsoft Azure — enterprise & hybrid focus & Windows Server & MS SQL DB
- Google Cloud (GCP) — data (Big Query) & AI strengths
- Others: Oracle Cloud (Cheap & Oracle DB), IBM Cloud(?), Alibaba Cloud(?)

Core Cloud Services

- Compute: VMs, containers, serverless
- Storage: Object(File, Image etc), block, databases
- Networking: VPC, CDN, load balancers
- Security: IAM, encryption, compliance

Compute

- **Definition**: The processing power that runs applications and workloads in the cloud.
- **Main Types**:
  - **Virtual Machines (VMs)** → Traditional cloud servers with customizable CPU, memory, and OS. (e.g., AWS EC2, Azure Virtual Machines).
  - **Containers** → Lightweight, portable, isolated environments for applications. (e.g., Kubernetes, AWS ECS/EKS, Azure AKS, GCP GKE).
  - **Serverless** → No server management; code runs on-demand and scales automatically. (e.g., AWS Lambda, Azure Functions, Google Cloud Functions).
- **Use Cases**: Web hosting, batch processing, microservices, event-driven apps.

Storage

- **Definition**: Cloud storage services manage data in various formats with scalability and durability.
- **Main Types**:
  - **Object Storage** → Stores unstructured data (images, videos, backups). (e.g., AWS S3, Azure Blob Storage, GCP Cloud Storage).
  - **Block Storage** → Low-latency storage for databases and VMs. (e.g., AWS EBS, Azure Disk Storage, GCP Persistent Disks).(Data Fixed does not change in the future.) => Back up Data.
  - **Databases** → Managed relational (SQL) and non-relational (NoSQL) services. (e.g., AWS RDS, DynamoDB, Azure Cosmos DB, Google Cloud SQL).
- **Use Cases**: File storage, backups, big data analytics, mobile app backends.

Networking

- **Definition**: Services that connect and deliver applications across cloud and user environments.
- **Main Types**:
  - **VPC (Virtual Private Cloud)** → A logically isolated section of the cloud to control networking (subnets, routing, firewalls).
  - **CDN (Content Delivery Network)** → Distributes content globally with low latency (e.g., AWS CloudFront, Azure CDN, Cloudflare).
  - **Load Balancers** → Distribute incoming traffic across multiple servers for reliability and scalability. (e.g., AWS ELB, Azure Load Balancer, GCP Load Balancing).
- **Use Cases**: Secure private networking, faster global delivery, handling millions of users.

Security

- **Definition**: Cloud security services ensure identity management, data protection, and compliance.
- **Main Types**:
  - **IAM (Identity and Access Management)** → Controls user and service permissions. (e.g., AWS IAM, Azure AD, GCP IAM).
  - **Encryption** → Protects data in transit and at rest (TLS, KMS, HSM).
  - **Compliance Tools** → Ensure alignment with regulations (GDPR, HIPAA, SOC2). (e.g., AWS Artifact, Azure Policy, GCP Compliance Reports).
- **Use Cases**: User access control, securing sensitive data, regulatory requirements.

Summary

- **Compute** = The “brains” (VMs, containers, serverless).
- **Storage** = The “memory” (object, block, databases).
- **Networking** = The “nervous system” (VPC, CDN, load balancing).
- **Security** = The “immune system” (IAM, encryption, compliance).

Pricing Models

- Pay-as-you-go (on-demand)
- Reserved / Committed use (discounts for commitment)
- Free tiers / trial credits

Pay-As-You-Go (On-Demand)

- **Definition**: Customers pay only for the resources they actually use, with no upfront commitment.
- **How it works**:
  - Compute → billed per second/minute/hour of VM/container/serverless runtime.
  - Storage → billed per GB stored per month.
  - Networking → billed per GB of data transferred.
- **Pros**:
  - Maximum flexibility.
  - Good for unpredictable or burst workloads.
  - No upfront costs.
- **Cons**:
  - More expensive in the long run compared to reserved/committed options.
- **Analogy**: Like paying for electricity or water — you pay for what you consume.
- **Example**: AWS EC2 on-demand instance, Azure pay-as-you-go VMs.

Reserved / Committed Use (Discounts for Commitment)

- **Definition**: Customers commit to using certain resources for a period (1–3 years) in exchange for significant discounts.
- **How it works**:
  - Reserved Instances (AWS), Committed Use Discounts (GCP), Reserved VM Instances (Azure).
  - Discounts can be **30–70% cheaper** than on-demand.
  - Some offer flexible options (convertible reserved instances, savings plans).
- **Pros**:
  - Big cost savings for predictable workloads.
  - Budgeting becomes easier.
- **Cons**:
  - Requires upfront planning.
  - Less flexible if workloads change.
- **Analogy**: Like signing a 1- or 2-year lease for an apartment — you pay less rent than short-term stays.
- **Example**: An enterprise running a production database 24/7 gets major savings by committing to reserved pricing.

Free Tiers / Trial Credits

- **Definition**: Cloud providers offer limited free usage to attract new customers and allow experimentation.
- **How it works**:
  - AWS Free Tier → 12 months of free usage (e.g., 750 hours/month of t2.micro instance, 5GB S3 storage).
  - Azure Free Account → $200 in credits + 25+ services free for 12 months.
  - Google Cloud Free Tier → $300 in trial credits + always-free services (e.g., 1 f1-micro VM, 5GB storage).
- **Pros**:
  - No cost entry point.
  - Great for students, startups, or proof-of-concept projects.
- **Cons**:
  - Limited capacity (not for production workloads).
  - Once trial expires, you must switch to pay-as-you-go.
- **Analogy**: Like a free sample at a supermarket — enough to try, not enough to live on.
- **Example**: A student building a small website on AWS using only free-tier resources.

Benefits of Cloud

- Scalability and elasticity
- Global reach and lower latency
- Cost efficiency (capex → opex)
- Faster innovation & shorter time-to-market

Google Global Cloud Data Center Locations

Challenges of Cloud

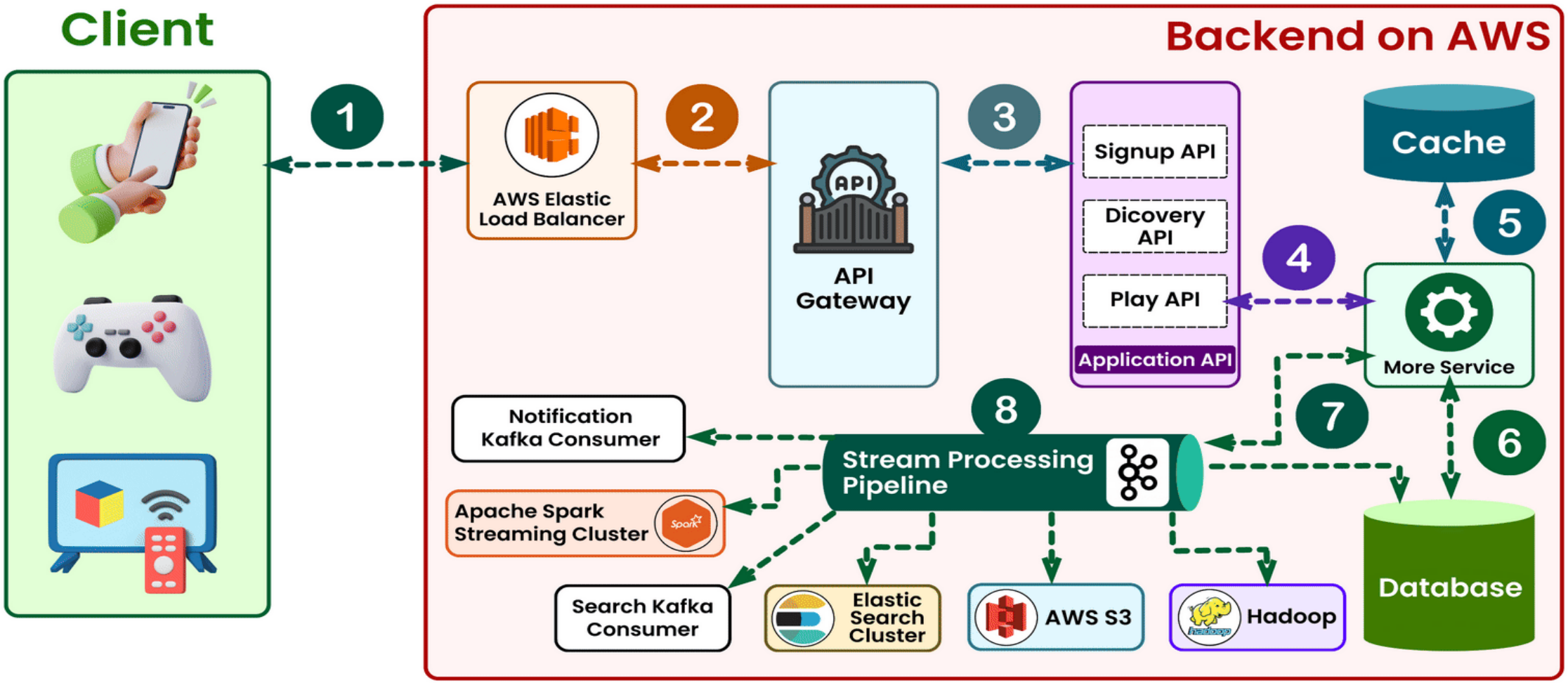
- Security & compliance responsibilities
- Risk of vendor lock-in
- Cost visibility and optimization needed
- Skills gap within teams

Real-World Use Cases

- Netflix: streaming scalability on AWS
- Startups: IaaS(VM) & PaaS for rapid product iterations
- Enterprises: hybrid cloud for legacy systems

Created by :
Rocky Bhatia

Microservice Architecture At NETFLIX



Cloud Trends & Future

- AI/ML integrated cloud services
- Edge computing & IoT (Waymo, Tesla)
- Green / sustainable cloud
- Industry-specific cloud offerings (healthcare, finance)

AI/ML Integrated Cloud Services

- **Definition**: Cloud platforms increasingly embed Artificial Intelligence (AI) and Machine Learning (ML) services directly into their offerings.
- **What's happening**:
  - Pre-trained ML models (vision, speech, NLP) are available as APIs.
  - Cloud-native AI platforms enable custom training at scale.
  - Generative AI (like ChatGPT, Bard, Claude) is offered as managed services.
- **Why it matters**: Democratizes AI — companies no longer need in-house ML experts to leverage AI capabilities.
- **Examples**:
  - AWS SageMaker, Bedrock (Generative AI).
  - Azure Cognitive Services, OpenAI on Azure.
  - Google Cloud Vertex AI, Gemini API(So good!) .
- **Use Cases**: Chatbots, fraud detection, predictive analytics, medical image recognition.

Edge Computing & IoT

- **Definition**: Processing data closer to where it is generated (at the “edge” of the network) instead of sending everything to central cloud data centers.
- **What’s happening**:
  - IoT devices (sensors, smart cars, wearables) produce massive data.
  - Edge reduces latency and bandwidth usage.
- **Why it matters**: Real-time applications (autonomous cars, AR/VR, industrial automation) require milliseconds of response.
- **Examples**:
  - AWS Greengrass, AWS IoT Core.
  - Azure IoT Hub, Azure Stack Edge.
  - Google Distributed Cloud Edge.
- **Use Cases**: Smart cities, connected vehicles, industrial IoT, healthcare monitoring devices.

Green / Sustainable Cloud

- **Definition**: Cloud providers are adopting environmentally friendly practices to reduce carbon footprints.
- **What's happening**:
  - Data centers powered by renewable energy (wind, solar, hydro).
  - Efficient cooling systems, carbon tracking dashboards.
  - Providers committing to net-zero emissions targets.
- **Why it matters**: Sustainability is now a business and regulatory priority.
- **Examples**:
  - Google Cloud → Carbon-neutral since 2007, aiming for 24/7 carbon-free energy.
  - Microsoft Azure → 100% renewable by 2025, carbon-negative by 2030.
  - AWS → Targeting 100% renewable energy by 2025.
- **Use Cases**: Companies choosing providers aligned with ESG (Environmental, Social, Governance) goals.

Green Eenergy

- Denmark's data center popularity stems from its cool climate for energy-efficient cooling, access to abundant renewable energy, stable and reliable power grid, and robust fiber optic connectivity. The country also benefits from strong government support for the sector, a focus on circular economy principles like heat reuse for district heating, and a stable geopolitical environment, making it an attractive location for sustainable data operations

Summary

- Cloud is the backbone of modern IT
- Service models: IaaS, PaaS, SaaS (plus FaaS)
- Deployment: Public, Private, Hybrid, Multi
- Prepare for AI, edge, and sustainability

Questions?

Thanks — I'm happy to answer your questions.

Practice Cloud Architecture: Typical 3-tier App