Cloud Computing

COEN 233 - COMPUTER NETWORKS

AUTHORS:

Archana Vellanki (7700009833)

Gowthami Edamalapati (00001653636)

Gnana Mounica Jasti(00001632742)

UNDER THE GUIDANCE OF

Dr.Keyvan Moataghed

AUDIENCE

This in-depth manual explores the fundamentals of cloud computing, breaking down its underlying architecture and the function of software in the cloud. It offers readers an in-depth look of the fundamental ideas behind cloud computing, as well as a look at the many cloud service providers and the essential software that drives cloud services.

Although not necessary, having some prior understanding of core concepts like operating systems, virtualization, containerization may help you better understand some of the more sophisticated topics covered here. This resource is intended to be used by those who are unfamiliar with cloud computing as well as by people who have some expertise and want to learn more.

A wide range of users, including IT students, software developers, cloud architects, are the target audience for this guide.

INTRODUCTION

This research project is designed to provide a comprehensive understanding of cloud computing, its architecture, software components, security, and major cloud service providers. The document is organized into several parts, each addressing key aspects of cloud technologies. Each chapter within these parts provides in-depth insights into specific topics, ensuring a comprehensive grasp of cloud computing and related subjects.

Document Organization:

Part 1: Fundamentals of Cloud Computing: This section introduces the essential characteristics, history, service models, deployment models, and enabling technologies of cloud computing.

Part 2: Cloud Computing Architecture: Part 2 delves into various architectural components such as workload distribution, elastic disc provisioning, dynamic scalability, and more.

Part 3: Software in Cloud Computing: Here, we explore virtualization technologies, containerization, cloud-native application development, and the role of DevOps and CI/CD in the cloud.

Part 4: Understanding Cloud Security and Cybersecurity: This section emphasizes the importance of cloud security, shared responsibility models, threat agents, common attacks, and security mechanisms.

Part 5: Cloud Service Providers: The final part focuses on the major cloud service providers, including AWS, Microsoft Azure, and Google Cloud Platform, detailing their services and offerings.

TABLE OF CONTENTS

AUDIENCE **2**
Introduction 3
TABLE OF CONTENTS **4**
Table of Tables: 6
Table of Figures 7
PART 1: FUNDAMENTALS OF CLOUD COMPUTING **9**
1.1. Key characteristics of cloud computing 9
1.2. Brief history of Cloud computing 10
1.3. Cloud Computing Service Models 12
1.3.1. Infrastructure as a Service (IaaS) 12
1.3.2. Platform as a Service (PaaS) 13
1.3.3. Software as a Service (SaaS) 14
1.3.4. Comparison of Cloud Service Models 15
1.4. Cloud Deployment Models 16
1.4.1. Public Cloud 16
1.4.2. Private Cloud 17
1.4.3. Hybrid Cloud 18
1.5.Cloud Enabling Technologies 19
1.5.1.Network and Internet Architecture: 19
1.5.2.Cloud Data Center Technology 20
1.5.3.Modern virtualization: 20
1.5.4.Service Technology 22
PART 2: CLOUD COMPUTING ARCHITECTURE **23**
2.1. Workload Distribution 23
2.2. Elastic Disc Provisioning 24
2.3. Hypervisors clustering 25
2.4. Distributed Data Sovereignty 27
2.5. Dynamic Scalability Architecture 28
2.5.1.Dynamic Horizontal Scaling: 28
2.5.2.Dynamic Vertical Scaling: 29
2.5.3.Dynamic Relocation: 29
2.6 Cloud Bursting Architecture 29
2.7 MultiCloud Architecture 30
2.8 Zero Downtime Architecture 31
2.9 Dynamic Failure Detection and Recovery Architecture 32
2.10 Elastic Resource Capacity Architecture 34
2.11 Redundant Storage Architecture 36
2.12 Virtual Server Clustering Architecture 38
2.13 Rapid Provisioning Architecture 39

PART 3: SOFTWARE IN CLOUD COMPUTING..... 40
3.1 Virtualization Technologies..... 40
3.1.1 Hypervisors..... 41
3.1.2 Containerization..... 43
3.1.2.1 Understanding Containers..... 43
3.1.2.2 Container Networks..... 44
3.1.2.3 Containerization on Physical servers-..... 45
3.1.2.4 Containerization on Virtual Servers..... 45
3.2 Container Orchestration..... 47
3.2.1 Container Network Addresses..... 48
3.2.2 Common Containerization Technologies..... 48
3.2.3 Other Virtualization technologies..... 49
3.3. Cloud-native Application Development..... 49
3.3.1. What is Cloud Native?..... 49
3.3.2. Key Characteristics for Building Cloud-Native Applications..... 50
3.3.3. Benefits of Cloud-Native Applications:..... 51
3.3.4. Challenges of Cloud-Native App Development..... 51
3.4.DevOps and Continuous Integration/Continuous Deployment (CI/CD) in the Cloud..... 52
3.4.1. Best DevOps Practices..... 52
PART 4: UNDERSTANDING CLOUD SECURITY and CYBERSECURITY..... 53
Why is cloud security important?..... 53
Shared Responsibility model..... 53
4.1.Threat Agent..... 54
4.2.Common Attacks..... 54
4.3.Cloud Security mechanisms..... 55
PART 5: CLOUD SERVICE PROVIDERS..... 58
5.1.AWS..... 58
5.1.1.Introduction..... 58
5.1.2.History and Growth of AWS..... 58
5.1.3.AWS Global Outreach..... 59
5.1.4.AWS Services..... 60
5.1.5 AWS Compute Services..... 61
5.1.5.1.AWS Lambda..... 61
5.1.5.2.Amazon EC2..... 62
5.1.6.Amazon Storage Services..... 64
5.1.6.1.Amazon S3..... 64
5.1.7.AWS Database Services..... 65
5.1.7.1.Amazon RDS..... 65
5.1.7.2.Amazon DynamoDB..... 65
5.1.8.Amazon Network Services..... 66
5.1.8.1.Amazon VPC..... 66

5.1.8.2.Amazon ELB..... 67
5.1.9.Amazon Security Services..... 67
5.1.9.1 AWS IAM:..... 68
5.1.9.2.Web Application Firewall (WAF)..... 68
5.2.Microsoft Azure..... 69
5.2.1.Overview..... 69
5.2.1.1.Azure cloud Strategy..... 70
5.2.2.Azure Global Data Center Presence..... 70
5.2.3.Azure Service Offerings..... 71
5.2.4.Storage Services..... 75
5.2.5.Networking Services..... 75
5.2.5.1.Azure Virtual Network (VNet)..... 75
5.2.5.2.Azure Load Balancer..... 76
5.2.6.Data Base Services..... 76
5.2.7.Identity and Access Management..... 77
5.3.Google Cloud Platform..... 77
5.3.1.Introduction to GCP..... 77
5.3.2.History and evolution of Google Cloud Platform:..... 78
5.3.3.Google cloud's Data centers..... 78
5.3.4.Google Cloud Service Offerings..... 79
5.3.4.1.GCP Compute Services..... 80
5.3.4.2.Google Cloud Storage Services..... 82
5.3.4.3.Google Cloud Networking Services..... 83
5.3.4.4.GCP Identity and Access Management..... 84
5.3.4.5.GCP DevOps and CI/CD Tools..... 84
PART 6: Future Developments and Conclusion..... 86
6.1.Edge Computing..... 86
6.1.1.Background..... 86
6.1.2.Edge Computing and Cloud..... 86
6.1.3 Key aspects of edge computing..... 87
6.1.4 General Architecture of Edge Computing..... 87
6.2.Artificial Intelligence..... 89
6.3 Conclusion..... 90
References:..... 90

Table of Tables:

- Table 5.1.5.2.1 - Features of EC2
- Table 6.1.1.1 - Main differences between Edge Computing and Cloud Computing

Table of Figures

- Figure 1.1.1 Key Characteristics of Cloud Computing
- Figure 1.2.1 History of cloud computing
- Figure 1.3.1 Cloud Computing Pyramid - The three service models IaaS, PaaS, SaaS
- Figure 1.3.1.1 IaaS - Leasing Virtual Servers Based on Hardware Specifications
- Figure 1.3.2.1 Accessing a Preconfigured PaaS Environment
- Figure 1.3.3.1 SaaS - Access to Cloud Service without IT Resource Details
- Figure 1.3.4.1 Flexibility and customization management differences for each Service Model
- Figure 1.4.1.1 Public Cloud - Organizations acting as cloud-consumers
- Figure 1.4.2.1 Private Cloud - On-Prem environment
- Figure 1.4.3.1 Hybrid Cloud - utilizing both private and public clouds
- Figure 1.5.1.1 The internetworking architecture of internet-based cloud deployment model
- Figure 1.5.3.1 Operating system-based virtualization
- Figure 1.5.3.2 Hardware-based virtualization
- Figure 1.5.3.3 A containerized application can run on any system where its containerization engine is installed
- Figure 2.1.1 Workload Distribution using Horizontal scaling and Load Balancers
- Figure 2.2.1. Cloud Consumer charged based on the provisioned instead of used storage
- Figure 2.2.2 Elastic Disc Provisioning - Cloud Consumer charged based on the allocated Usage
- Figure 2.3.1 Hypervisor Clustering Architecture
- Figure 2.4.1 Data Governance Manager ensuring data is according to regional data protection regulations.
  Figure 2.7.1 An organization uses different types of resources from different clouds
- Figure 2.8.1 Virtual Server A hosted by physical Server A is moved to Physical Server B by a live VM migration program.
- Figure 2.9.1 The intelligent watchdog monitor keeps track of cloud consumer requests (1) and detects that a cloud service has failed (2).
- Figure 2.9.2 The intelligent watchdog monitor notifies the watchdog system (3), which restores the cloud service based on predefined policies. The cloud service resumes its runtime operation (4).
- Figure 2.9.3 The intelligent watchdog monitor notifies the watchdog system (3), which restores the cloud service based on predefined policies. The cloud service resumes its runtime operation (4).
- Figure 3.1.1.1 Three virtual servers created and run by a hypervisor that exists on two physical servers.
- Figure 3.1.1.2 The physical server hosts only the hypervisor that creates virtual servers, each with its own operating system.
- Figure 3.1.1.3.The physical server hosts its own operating system as well as a hypervisor that creates virtual servers with their own operating system environments.

- Figure 3.1.2.1 The symbol on the left is the container icon. The symbol on the right is also used to represent a container and to show its contents.
- Figure 3.1.2.1 A container engine creating two different containers.
- Figure 3.1.2.3.1 A physical server with an operating system hosts a containerization platform that creates containers, each with an environment that has only a subset of the underlying operating system.
- Figure 3.1.2.4.1 A physical server with no operating system hosts a hypervisor that creates virtual servers with operating systems, each of which hosts a containerization platform that can create containers that only have an operating system subset.
- Figure 3.1.2.4.2 A physical server with an operating system hosts a hypervisor that creates virtual server environments with their own operating systems. Each virtual server hosts a containerization platform that creates containers that host a subset of the operating system.
- Figure 3.3.1.1 Monolithic vs Microservices Architecture
- Figure 3.4.1 DevOps Model
- Figure 4.3.1 In the IAM system, User A undergoes authentication, confirming their identity, and is classified as a member of Role X. Leveraging this role-based identification, the IAM system grants User A authorization to access two designated folders on a physical file server.
- Figure 4.3.2 A security professional with a DLP system blocks a user from storing company data on a USB drive (1), scans a corporate server with files in folders to identify the ones with confidential data (2), and forces an email going outside of the organization boundary to be encrypted (3).
- Figure 5.1.2.1 AWS History
- Figure 5.1.3.1 Global Outreach of AWS
- Figure 5.1.5.1.1 Amazon S3, API Gateway, AWS Lambda and DynamoDB work together to retrieve weather data for a web application
- Figure 5.1.5.2.1: Amazon EC2 Instance in Amazon VPC Architecture
- Figure 5.1.9.1 Different AWS Security Solutions and Services
- Figure 5.1.9.2.1 Basic working of AWS WAF
- Figure 5.2.3.1 Brief overview of how a Azure App service works
- Figure 5.2.3.2 how control plane interacts with nodes
- Figure 5.2.3.3 containerized applications interaction with virtual network and storage
- Figure 5.2.3.4 Azure Functions quick glance
- Figure 5.2.3.5 Chaining together a series of function executions
- Figure 5.2.3.6 Notification workflow
- Figure 5.2.3.7 Serverless workflow topology
- Figure 5.2.5.2.1 Balancing multi-tier applications by using both public and internal Load Balancer
- Figure 5.3.3.1 Google Cloud Datacenters
- Figure:5.3.4.1.1 Google Clouds compute services
- Figure 5.3.4.2.1. Architectural diagram of the Google File System
- Figure 6.1.4.1 Edge Computing reference architecture 3.0

PART 1: FUNDAMENTALS OF CLOUD COMPUTING

Cloud computing is a technology model that allows users to access and use computing resources, including servers, storage, databases, networking, software, and other IT services, over the internet (the "cloud") on a pay-as-you-go basis. In simpler terms, it lets users rent or lease these resources instead of having to own and manage them on their own computers.

1.1. Key characteristics of cloud computing

Figure 1.1.1 Key Characteristics of Cloud Computing

On-Demand Usage: Users can provision and manage computing resources as needed, often through a web-based interface. This means you can scale up or down quickly based on your requirements.

Broad Network Access: Cloud services are accessible over the internet from a variety of devices, such as laptops, smartphones, and tablets.

Resource Pooling: Cloud providers pool their computing resources to serve multiple customers. These resources are dynamically allocated and reassigned according to demand.

Rapid Elasticity: Cloud resources can be quickly scaled up or down to handle changing workloads. This scalability is achieved without requiring major changes in infrastructure.

Measured Service: Cloud usage is metered and billed according to actual consumption. This pay-as-you-go model allows organizations to only pay for the resources they use

Resiliency and Availability: Cloud resiliency refers to how quickly services can recover after problems and Availability refers to how easily people can access them from different places.

Flexibility: Companies need to scale as their business grows. The cloud offers customers the freedom to scale up or down without the need to restart their servers.

1.2. Brief history of Cloud computing

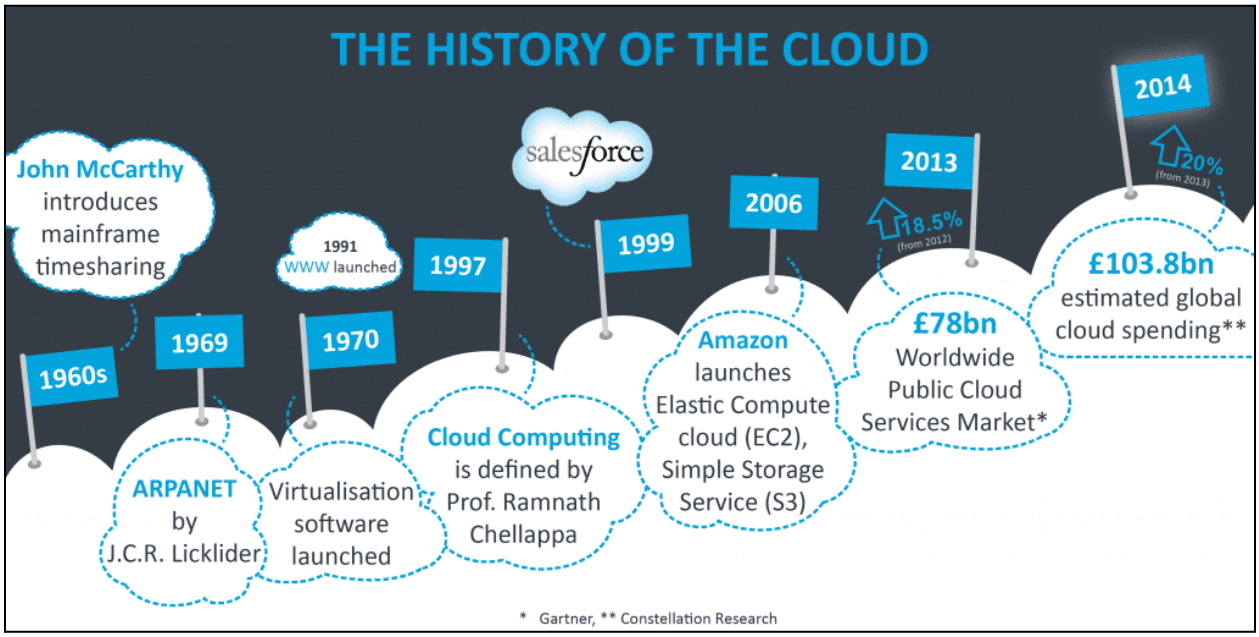

Figure 1.2.1 History of cloud computing

The idea of computing in a "cloud" traces back to the origins of utility computing, a concept that computer scientist John McCarthy publicly proposed in 1961:

"*If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility.… The computer utility could become the basis of a new and important industry*."

In his visionary statement, McCarthy talked about a future where computers, following the footsteps of a public utility model such as the telephone system, would transform into a similar model used by the public. He predicted that computing would evolve into a fundamental and rapidly growing sector. McCarthy's words marked the inception of a groundbreaking idea that would shape the trajectory of cloud computing.

As time went on, these early visions materialized, with the advent of internet-based utilities in the mid-1990s. The emergence of key cloud computing providers like Salesforce.com and

Amazon Web Services (AWS) in the late 1990s and 2006, respectively, allowed organizations to rent computing power. Formal definitions from experts further defined cloud computing as delivering scalable and on-demand IT services over the internet. All of this history has shaped the modern cloud computing we use today for a wide range of services and innovations.

Cloud computing was necessary due to two key factors: cost reduction and business agility. Maintaining IT infrastructure in conventional ways was expensive and resource intensive, involving significant expenditure and overhead. On the other hand cloud computing presented budget-friendly options through resource sharing and adaptable pay-as-you-go models. In addition to that, the need for business to respond to rapidly changing requirements provided a significant push to cloud computing. With its ability to scale based on demand in an automated and rapid manner, cloud computing brought forth a dynamic shift to handling infrastructure.

Several key technical innovations have played a pivotal role in the development of cloud computing:

1. **Clustering**: Clustering involves connecting independent IT resources to function as a single system, reducing system failure rates and enhancing reliability through redundancy and failover features. This concept is core to cloud platforms, with clusters ensuring availability and reliability.
2. **Grid Computing:** Grid computing organizes computing resources into logical pools that collaborate to form a high-performance distributed grid. Grid computing's influence on cloud computing is seen in shared features like networked access, resource pooling, and scalability. Both grid and cloud computing employ distinctive approaches to establish these features.
3. **Capacity Planning:** Capacity planning is the process of managing IT resource demands to achieve efficiency and performance. Various strategies, such as lead, lag, and match strategies, are used to balance peak usage without excessive infrastructure expenditure, which can lead to under-provisioning or over-provisioning.
4. **Virtualization:** Virtualization converts physical IT resources into virtual ones, enabling shared use by multiple users. Virtualization software abstracts physical resources, allowing guest operating systems and applications to run as if on separate physical servers.
5. **Containerization:** Containerization is a form of virtualization that creates virtual hosting environments known as "containers" without the need for individual virtual servers. Containers provide virtual environments for hosting software programs and resources, streamlining deployment and resource management.
6. **Serverless Environments**: Serverless environments eliminate the need for manual server provisioning. They automatically deploy software packages, including the required server components and configurations, simplifying application deployment, scaling, and maintenance. Serverless architectures streamline cloud-based application deployment.

1.3. Cloud Computing Service Models

As numerous applications run on different cloud models, it's crucial to assess if these solutions align with a company's needs. You should ensure that each application in your portfolio is using the appropriate cloud model. Cloud computing consists mainly of three "as a service" models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

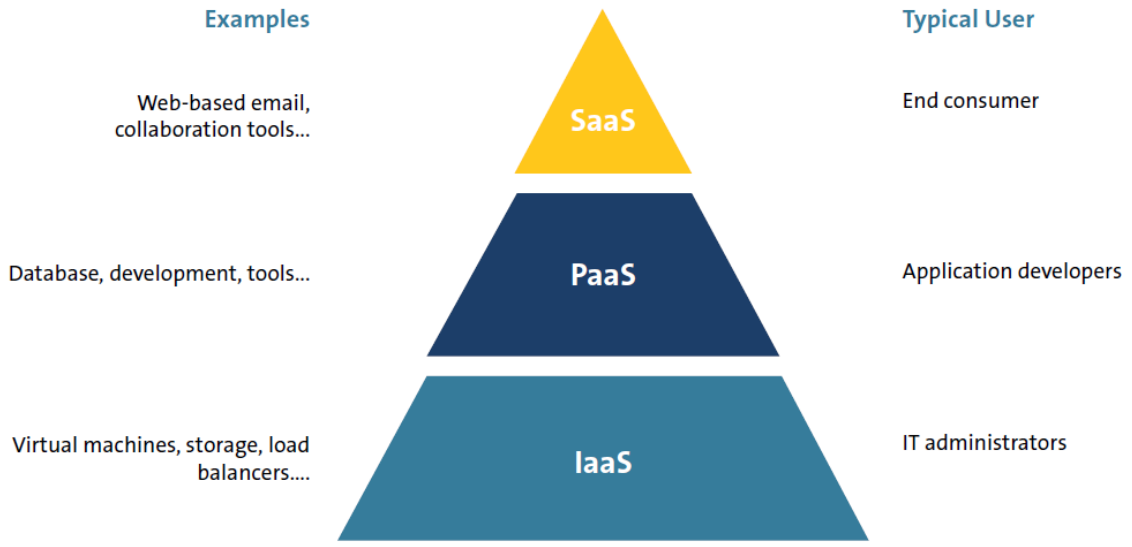

Figure 1.3.1 Cloud Computing Pyramid - The three service models IaaS, PaaS, SaaS

1.3.1. Infrastructure as a Service (IaaS)

IaaS, or Infrastructure as a Service, is a cloud computing model that provides on-demand access to computing resources such as servers, storage, networking, and virtualization. Before the advent of IaaS, organizations had to spend a ton of money on buying and maintaining the infrastructure. In addition to that a separate team of specialists was required to ensure scalability and uninterrupted access. With IaaS, organizations can just focus on implementing the business logic, letting the cloud providers take care of the painstaking work of maintaining and scaling the infrastructure.

IaaS examples include Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), IBM Cloud, and Oracle Cloud Infrastructure (OCI).

To determine whether IaaS is suitable for your use case, factors like your existing infrastructure, cost savings, redundancy, and compliance requirements need to be considered.

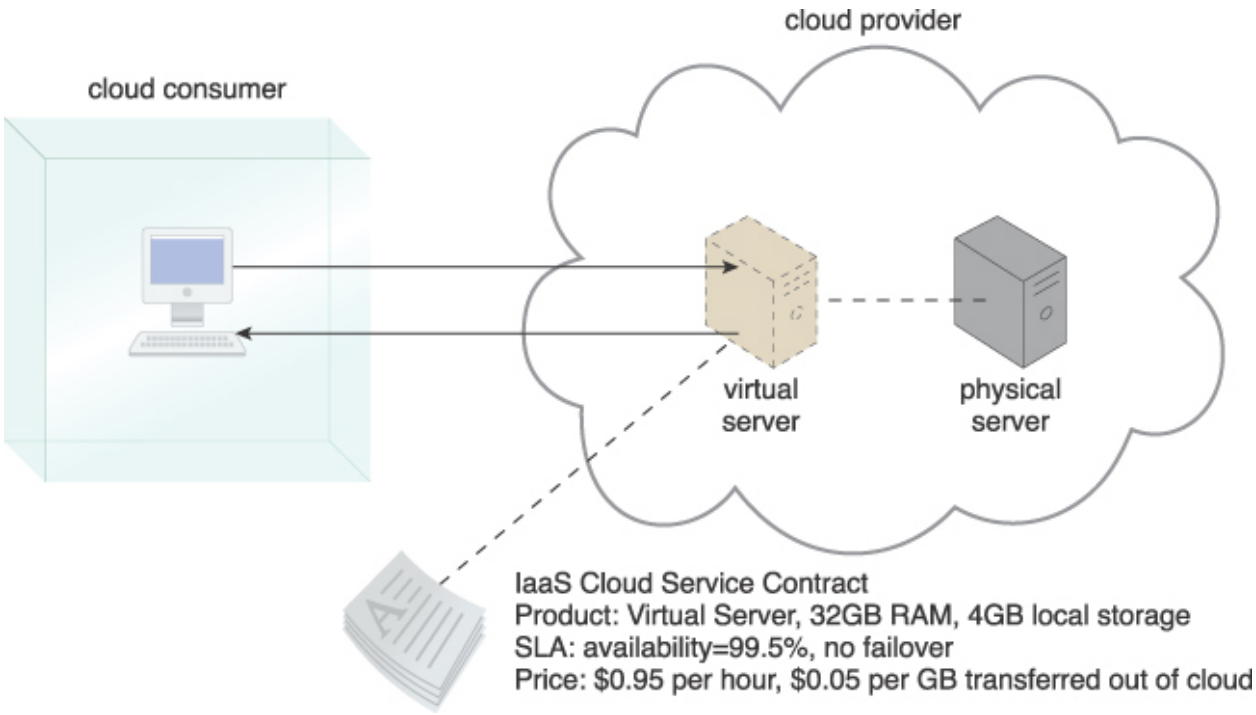

Figure 1.3.1.1 IaaS - Leasing Virtual Servers Based on Hardware Specifications

1.3.2. Platform as a Service (PaaS)

PaaS, or Platform as a Service, offers a preconfigured and ready-to-use environment with pre-deployed and configured IT resources. It relies on this environment to support the entire application delivery process. Cloud consumers opt for PaaS for various reasons, including:

- Scalability and cost-efficiency for extending on-premises environments into the cloud.
- Replacing on-premises setups entirely with a ready-made cloud environment.
- Becoming a cloud provider, offering their cloud services to external users.

With PaaS, cloud consumers avoid the administrative tasks of setting up and maintaining basic infrastructure, which is typically handled in the IaaS model. However, they trade some control over the underlying IT resources for the convenience of a pre-built platform

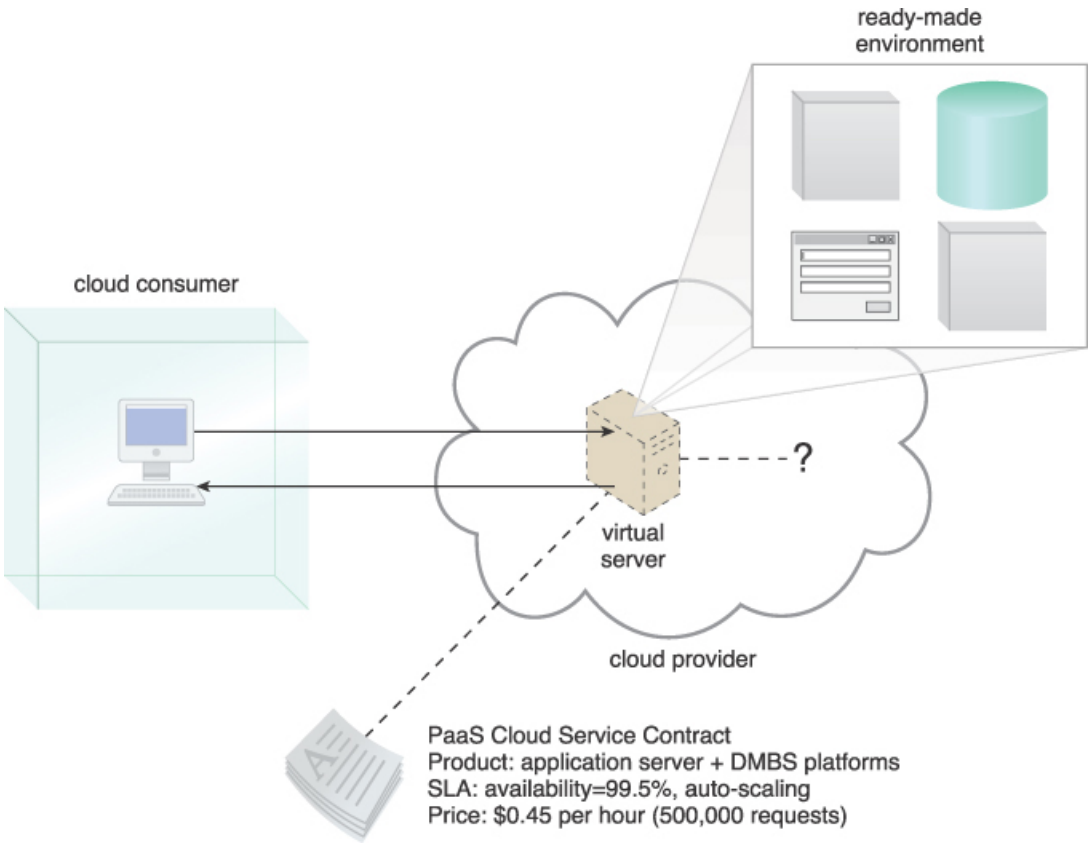

Figure 1.3.2.1 Accessing a Preconfigured PaaS Environment

1.3.3. Software as a Service (SaaS)

SaaS typically refers to a software program delivered as a shared cloud service, functioning as a product or a general utility. This common SaaS model allows for the broad availability of a cloud service, often for commercial purposes, to a wide range of cloud consumers.

Typically, cloud consumers have very limited administrative control over a SaaS setup. The provisioning is usually handled by the cloud provider, although the legal ownership of the SaaS-based cloud service may lie with the entity taking on the role of the cloud service owner. For example, an organization acting as a cloud consumer while using a PaaS environment can create a cloud service within that same environment and offer it as a SaaS service to other organizations, thus transitioning into the role of the cloud provider for those cloud consumers utilizing the service.

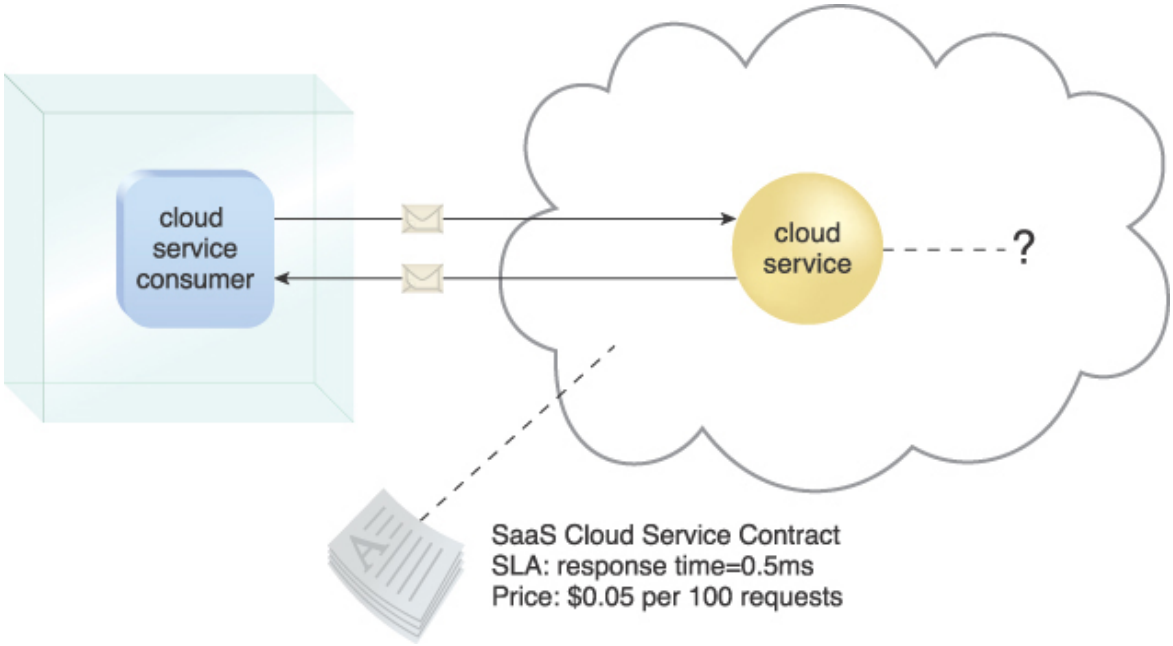

Figure 1.3.3.1 SaaS - Access to Cloud Service without IT Resource Details

1.3.4. Comparison of Cloud Service Models

IaaS, PaaS, and SaaS differ mainly in terms of how much control you have compared to the service provider. The choice depends on your flexibility and customization requirements, each having its advantages and disadvantages.

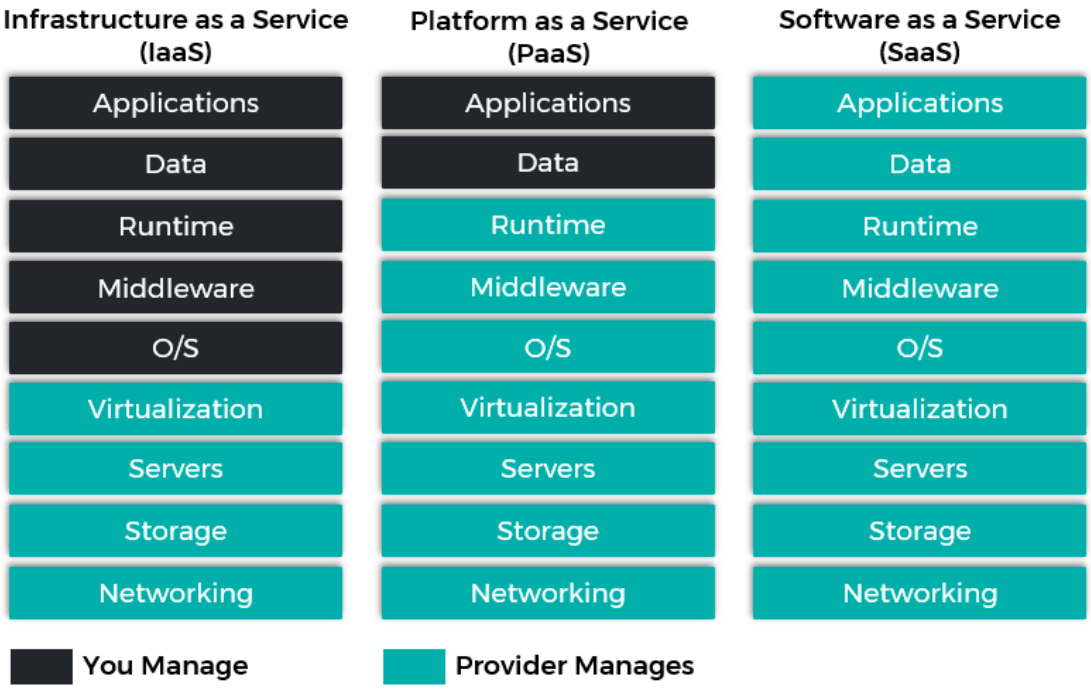

Figure 1.3.4.1 Flexibility and customization management differences for each Service Model

1.4. Cloud Deployment Models

1.4.1. Public Cloud

Public clouds represent the most common form of cloud computing deployment. These cloud resources, such as servers and storage, are owned and operated by a third-party cloud service provider and are accessible via the internet. In a public cloud, the cloud provider owns and manages all the hardware, software, and underlying infrastructure.

Within a public cloud setting, hardware, storage, and network resources are shared among various organizations. With the help of a web browser, professionals can access their services and oversee their accounts. Popular services offered by public clouds include virtual servers, storage, databases, development tools, and solutions for analytics and machine learning.

Advantages of public clouds include:

1. Cost Savings: No need to invest in hardware or software, you pay for the services you use.
2. Maintenance-Free: Your service provider handles maintenance tasks.
3. Scalability: On-demand resources are available to meet your business requirements.
4. Reliability: The extensive server network minimizes the risk of failures.

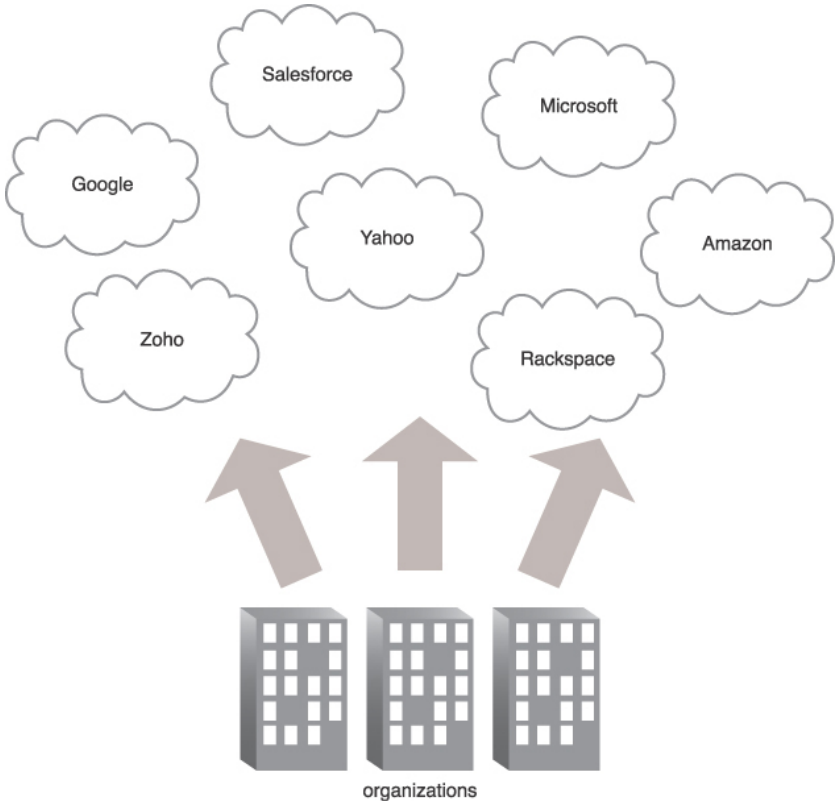

Figure 1.4.1.1 Public Cloud - Organizations acting as cloud-consumers

1.4.2. Private Cloud

A private cloud comprises cloud computing resources exclusively utilized by a single business or organization. This private cloud can either be situated within the organization's own on-site datacenter or hosted by a third-party service provider. However, in a private cloud, all services and infrastructure are consistently maintained on a private network, and both the hardware and software are dedicated solely to that organization.

This setup allows a private cloud to provide a higher degree of customization to meet the organization's specific IT needs. Private clouds are frequently adopted by government agencies, financial institutions, and other medium to large-sized organizations with mission-critical operations that require increased control over their environment.

Advantages of a private cloud include:

1. Enhanced Flexibility: Organization can tailor the cloud environment to suit its unique business requirements.
2. Greater Control: Resources are not shared with external parties, allowing for high control and privacy.
3. Improved Scalability: Private clouds often offer superior scalability compared to on-premises infrastructure.

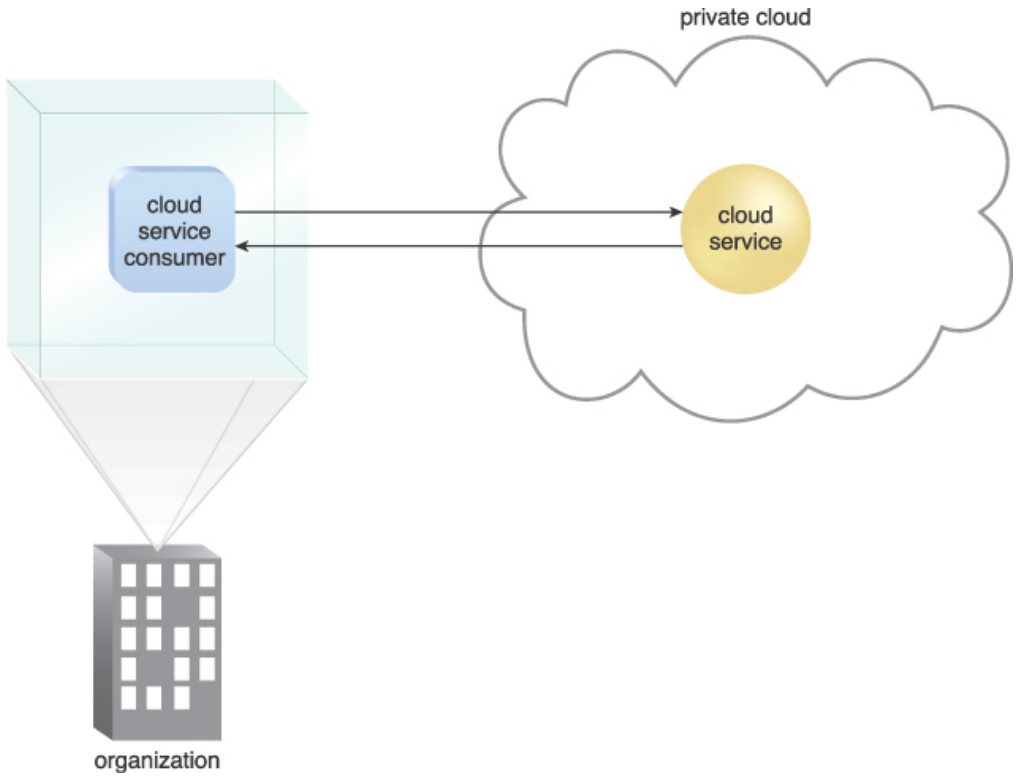

Figure 1.4.2.1 Private Cloud - On-Prem environment

1.4.3. Hybrid Cloud

Hybrid cloud combines on-site or private cloud with a public cloud, allowing data and apps to move between them.

Many organizations opt for hybrid cloud due to regulatory compliance, cost-efficiency, and low-latency requirements. It is evolving to include edge computing, bringing cloud power closer to IoT devices for reduced latency and offline reliability.

Benefits of a hybrid cloud platform include flexibility, deployment options, security, compliance, and cost savings. You can seamlessly scale up on-premises infrastructure to the public cloud as needed, avoiding major capital expenses. Plus, you only pay for resources you temporarily use.

Advantages of the hybrid cloud include:
Control: Keep sensitive assets on your private infrastructure.
Flexibility: Access extra resources in the public cloud when required.
Cost-Effectiveness: Pay for extra computing power as needed.
Easy Transition: Gradual migration allows a smooth transition to the cloud.

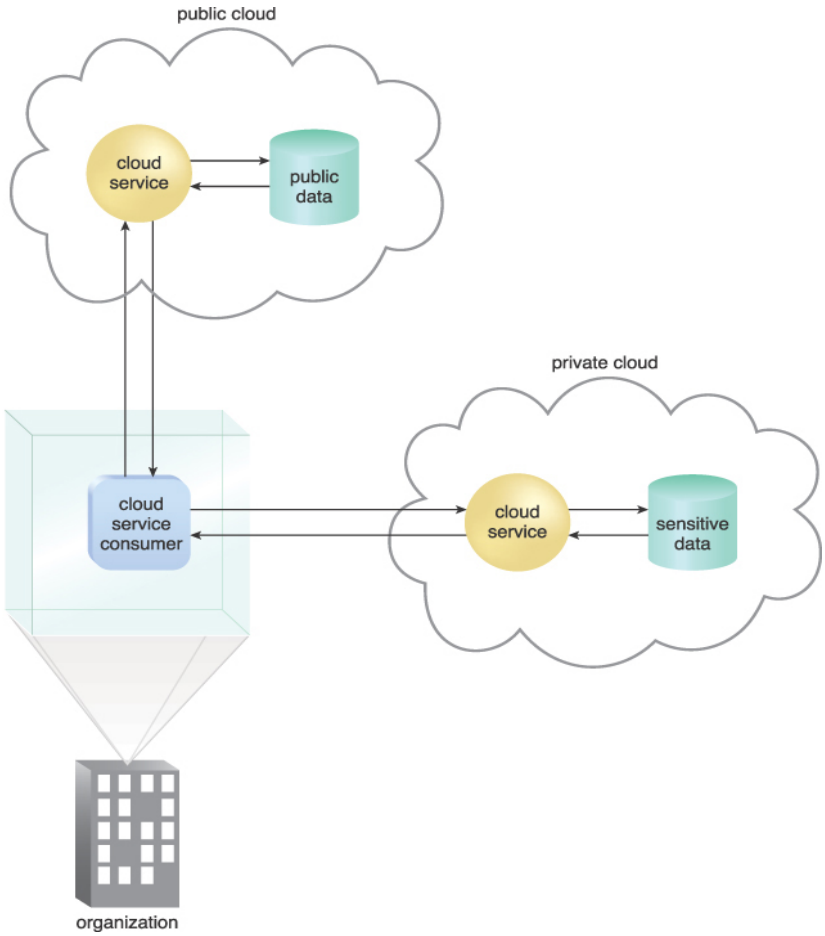

Figure 1.4.3.1 Hybrid Cloud - utilizing both private and public clouds

There’s no one type of cloud computing that’s right for everyone. Several different cloud computing models, types, and services have evolved to meet the rapidly changing technology needs of organizations. Which deployment method to be chosen depends on the business needs.

1.5.Cloud Enabling Technologies

The core technologies that form the foundation of modern clouds have been in existence and reached maturity before the emergence of cloud computing. However, the evolution and development of cloud computing have contributed to advancements in certain aspects of these technologies.

1.5.1.Network and Internet Architecture:

Cloud service providers have the ability to configure the services they offer so that users can access them via the internet from outside of their company as well as from within its internal network. Both external customers who wish to use cloud-based internet services and internal users who require access to company IT resources will benefit from this flexibility. Mobile clients and customers who use cloud services must be able to access them through cellular and wireless networks. This is achieved by using technologies such as mobile edge computing. Internet service providers, or ISPs, are the businesses that offer internet connectivity. Their extensive networks are linked to other networks and institutions to form the internet that we use today.Organizations must take into account their unique bandwidth and latency needs when selecting cloud-based solutions. While certain applications might demand minimal latency for fast reaction times, others might need a lot of bandwidth.

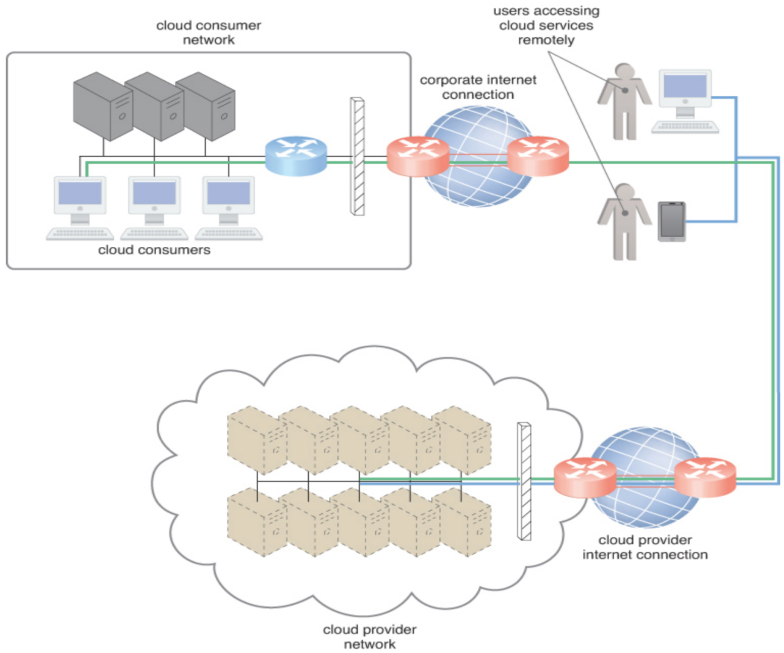

Figure 1.5.1.1 The internetworking architecture of internet-based cloud deployment model

1.5.2.Cloud Data Center Technology

Consolidated IT resources like servers, databases, networking and telecommunications equipment, and software systems are housed in specialized IT infrastructure known as data centers. Data centers for cloud providers often require additional technologies like,

Virtualization: To efficiently abstract and manage physical IT resources, data centers make use of virtualization technologies. This involves the virtualization of networking and computer components, which makes resource allocation, control, and monitoring simpler.

Standardization and Modularity: In order to facilitate scalability, growth, and quick hardware replacements, data centers are constructed using modular architectures and standardized commodity hardware. These building blocks comprise various similar facility infrastructure and equipment components. Reducing investment and operating expenses primarily requires standardization and modularity.

Autonomic Computing: Data centers can perform some activities on their own with the help of autonomous computing, which reduces the need for human interaction. This improves total system stability through the use of self-configuration, self-optimization, self-healing, and self-protecting features.

Storage Hardware: To deliver scalable, fault-tolerant, and high-performance storage solutions, data centers rely on specialized storage systems that make use of technologies like RAID, I/O caching, and storage virtualization.

Serverless environment: A serverless environment is made up of technologies that automatically supply resources for deployed applications without the need to set up the supporting infrastructure. There is no need to worry about capacity planning, administration, resiliency, or elasticity configurations since the serverless environment takes care of these things. The deployed logic still runs on servers, whether they are physical, virtual, containerized, or something else entirely.

1.5.3.Modern virtualization:

Virtualization technology is a foundation of contemporary cloud platforms. It provides a variety of virtualization types and technology layers.

Simplifying Hardware-Software Dependencies The process of virtualization transforms individual IT hardware into uniform, similar copies of software. This procedure ensures hardware independence, making it simple to migrate virtual servers between virtualization hosts. This method automatically fixes a variety of software-hardware incompatibilities.

Servers consolidation Server consolidation, in which several virtual servers share a single physical server, is made easier by virtualization. Among the many advantages of server consolidation are improved hardware utilization, load balancing, and best use of existing IT resources.

Operating system-based virtualization: Operating system-based virtualization allows users to create and manage virtual servers or VMs within their existing host operating system.This approach allows multiple operating systems to run on a single physical machine, making it a valuable tool for various purposes, including software testing, development, and server consolidation.

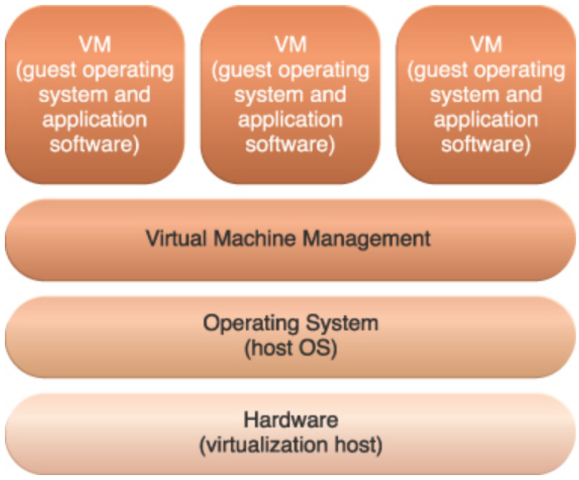

Figure 1.5.3.1 Operating system-based virtualization

Hardware-based virtualization: A hypervisor(virtualization software) is installed directly on the physical host hardware. With this approach, there is less need for the host operating system to function as an intermediary between virtual servers and the hardware.

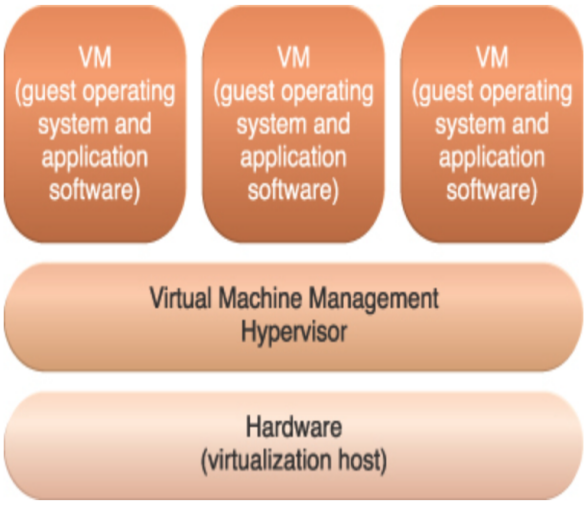

Figure 1.5.3.2 Hardware-based virtualization

Containers and Application-Based Virtualization:
Each individual container contains only the application and its libraries and dependencies. Containers are small, fast, and portable because, unlike a virtual machine, containers do not need to include a guest OS in every instance and can, instead, simply leverage the features and resources of the host OS. Containers are suitable for application-based virtualization because applications running in containers can run on any platform, regardless of the underlying operating system or hardware architecture.

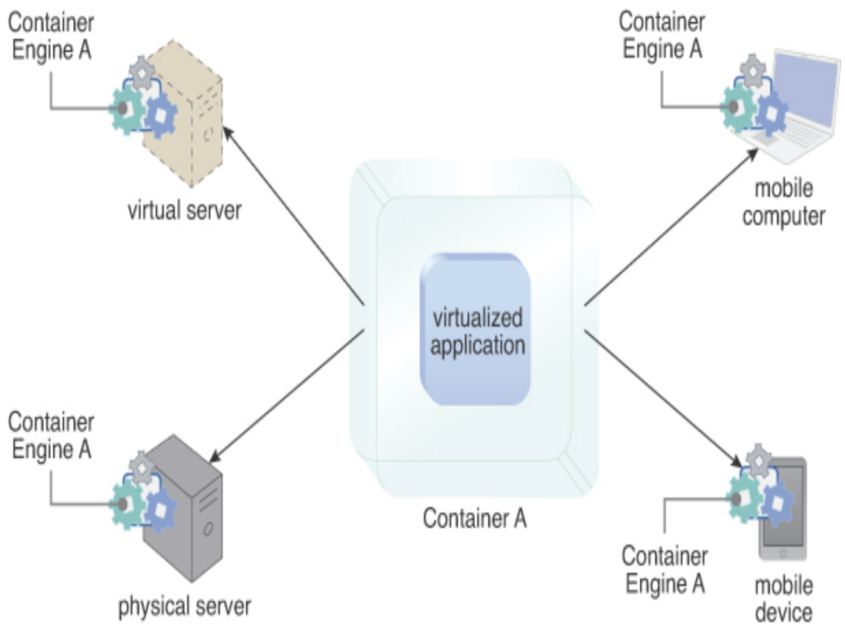

Figure 1.5.3.3 A containerized application can run on any system where its containerization engine is installed

1.5.4.Service Technology

The field of service technology is fundamental to cloud computing and serves as the foundation for the "as a service" cloud delivery models. Several service technologies are used to create and expand upon cloud-based environments.

Service Middleware:Large market of middleware platforms that evolved from messaging-oriented middleware (MOM) platform to support complex service compositions. The most common types are, Enterprise service bus (ESB) platforms provide features like service brokerage, routing, and message queuing, and orchestration platforms host and execute workflow logic for service composition. Both types of middleware can be deployed in cloud-based environments.
Web-Based RPC:Web-Based RPC (Remote Procedure Call): RESTful services are frequently used by cloud providers to distribute resources, although they may have performance issues because of the repeated message exchanges. Web-based RPC protocols like gRPC, GraphQL, and Falcor address these limitations by combining the performance benefits of RPC with web-based communication, overcoming the limitations of traditional RPC frameworks. These service technologies are essential for creating, managing, and providing a range of cloud services in cloud-based systems.

PART 2: CLOUD COMPUTING ARCHITECTURE

2.1. Workload Distribution

A workload refers to the amount of computing tasks, processes, or activities that are being executed or processed by a computer system or network at a given time. It's a measure of the demands placed on a system's resources, such as its central processing unit (CPU), memory, storage, and network capacity.

Workloads can be classified into different types based on User Traffic patterns.

1. **Static/Predictable workloads** - These are workloads with predictable resource usage and no unexpected surges in traffic. Examples include a Learning Management System(LMS) in an organization with employees/students accessing it.
2. **Periodic workloads** - These workloads see traffic surge in specific timelines. For instance, e-commerce websites could see a significant increase in visitors during holiday seasons, because of the numerous special offers that draw in a large customer base.
3. **Unpredictable workloads** - These workloads experience sudden and massive increases in traffic making it very unpredictable and tough to handle. Examples of this type can be Social media platforms like Twitter/Instagram, where user activity can explode during significant events or trending topics.

Apart from the above classification, workloads can also be classified based on Resource Requirements

1. **General Compute** - Workloads that don't have specific computational needs and typically run on the default configuration of the cloud. These include common web apps, web servers, distributed data stores, and containerized microservices.
2. **CPU Intensive** - These workloads have high computational requirements. These are typically deep learning applications, highly scalable multiplayer gaming apps built to handle quite a number of concurrent users, running big data analytics etc.
3. **Memory Intensive** - Workloads that need memory and processing power to execute millions of transactions per second. These include real-time streaming data, caches, and distributed databases
4. **GPU Accelerated Computation** - Workloads such as speech recognition, computational fluid dynamics, autonomous vehicles data processing, require the power of GPUs along with the CPUs to run the accelerated tasks.
5. **Storage Optimized Database Workloads** - Workloads such as in-memory databases, highly scalable NoSQL databases, and data warehouses

There are different customized solutions that address the specific needs for each workload type, Horizontal scaling is one such solution. Horizontal scaling involves increasing or decreasing the number of instances of a particular resource, such as servers or virtual machines, to distribute the incoming traffic across multiple devices. Load balancers are used to evenly distribute incoming network traffic across multiple servers or resources, ensuring efficient utilization and

preventing any server from being overwhelmed. With efficient load balancing you can take care of both underutilization and overutilization of resources.
The ability to scale resources automatically based on real-time demand is a fundamental advantage that cloud computing offers in handling varying workloads while maintaining high availability and performance.

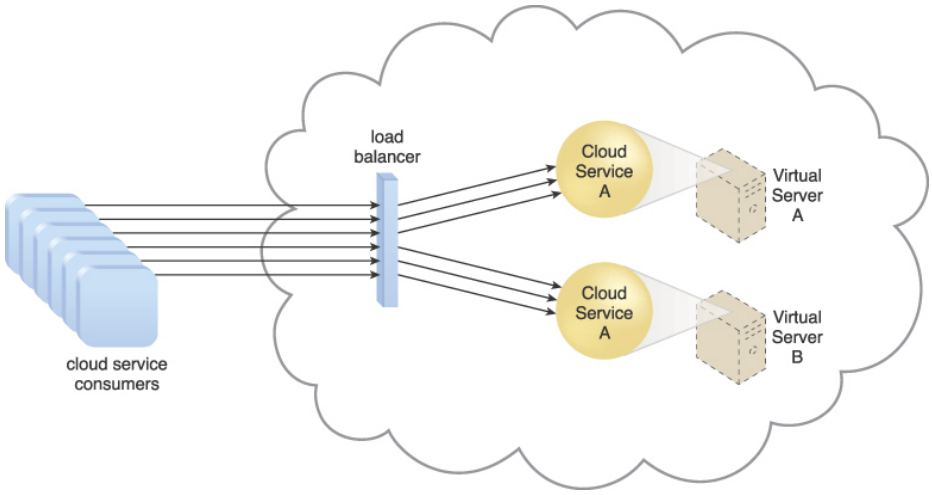

Figure 2.1.1 Workload Distribution using Horizontal scaling and Load Balancers

2.2. Elastic Disc Provisioning

When cloud providers charge for fixed-disk storage allocation, the billing is based on the capacity of the disks, irrespective of their actual storage consumption. As a result, cloud consumers are generally billed for more storage than they consume. The figure illustrates the example where a cloud consumer who requested for three 150GB hard-drives is charged for entire hard drives even though they have not installed any software on those.

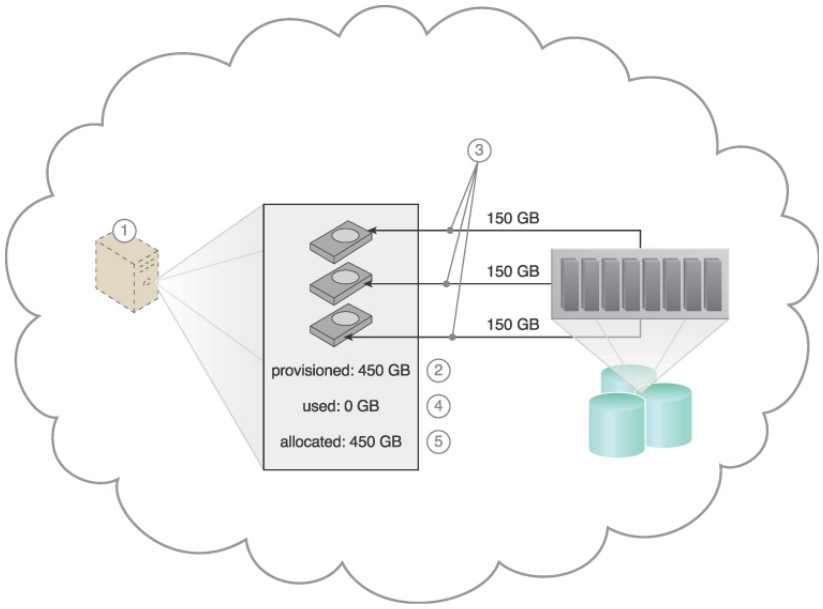

Figure 2.2.1. Cloud Consumer charged based on the provisioned instead of used storage

Elastic Disc Provisioning rectifies this issue by providing dynamic storage provisioning, ensuring that consumers are accurately billed for the specific storage that is used. It employs thin-provisioning technology to allocate storage as needed, aided by real-time usage monitoring for precise billing.

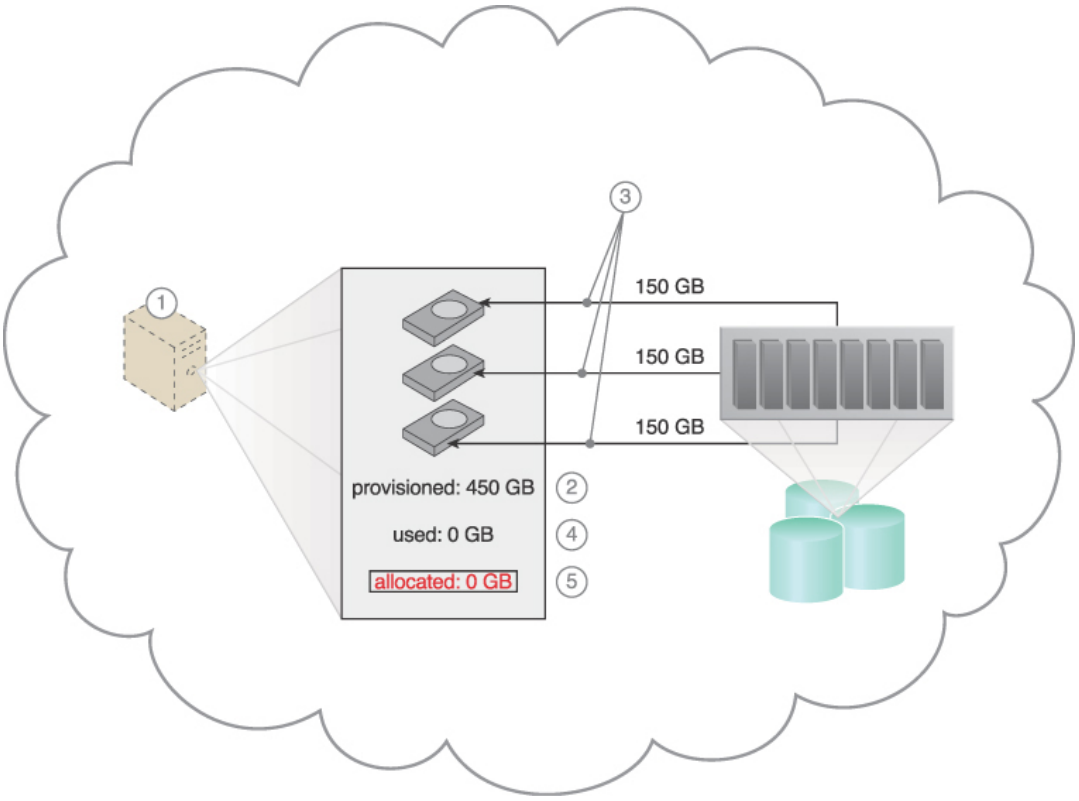

Figure 2.2.2 Elastic Disc Provisioning - Cloud Consumer charged based on the allocated Usage

Thin-provisioning software, implemented through the hypervisor, enables dynamic storage allocation, while a pay-per-use monitor tracks and reports usage data for billing. Supplementary components, such as specialized cloud usage monitors and resource replication, can be integrated into this architecture for monitoring and ensuring the availability of storage resources.

2.3. Hypervisors clustering

Hypervisor: Before diving into clustering, it's essential to understand the concept of a hypervisor. A hypervisor is a piece of software or hardware that creates and manages virtual machines (VMs) on a physical server. It abstracts the underlying hardware and allows multiple VMs to run independently on a single physical server.

If the physical server hosting the hypervisor fails due to hardware issues, software errors, or any other reason, the hypervisor and all the VMs it manages may also become unavailable. This leads to a domino effect where multiple VMs are simultaneously affected, causing downtime for critical workloads. In addition to VMs, the physical server typically hosts other IT resources such

as storage, network services, and management tools. The failure of the physical server disrupts these resources as well, compounding the impact on the entire IT infrastructure and reputation of the company.

Hypervisor Clustering: Hypervisor clustering is the process of grouping multiple hypervisor hosts (physical servers) together into a cluster. It involves using specialized software to provide high availability and fault tolerance for virtualized workloads.

Heartbeat messages are passed between clustered hypervisors and a central VIM(Virtual Infrastructure Manager) to maintain status monitoring. Shared storage is provided for the clustered hypervisors and further used to store virtual server disks.

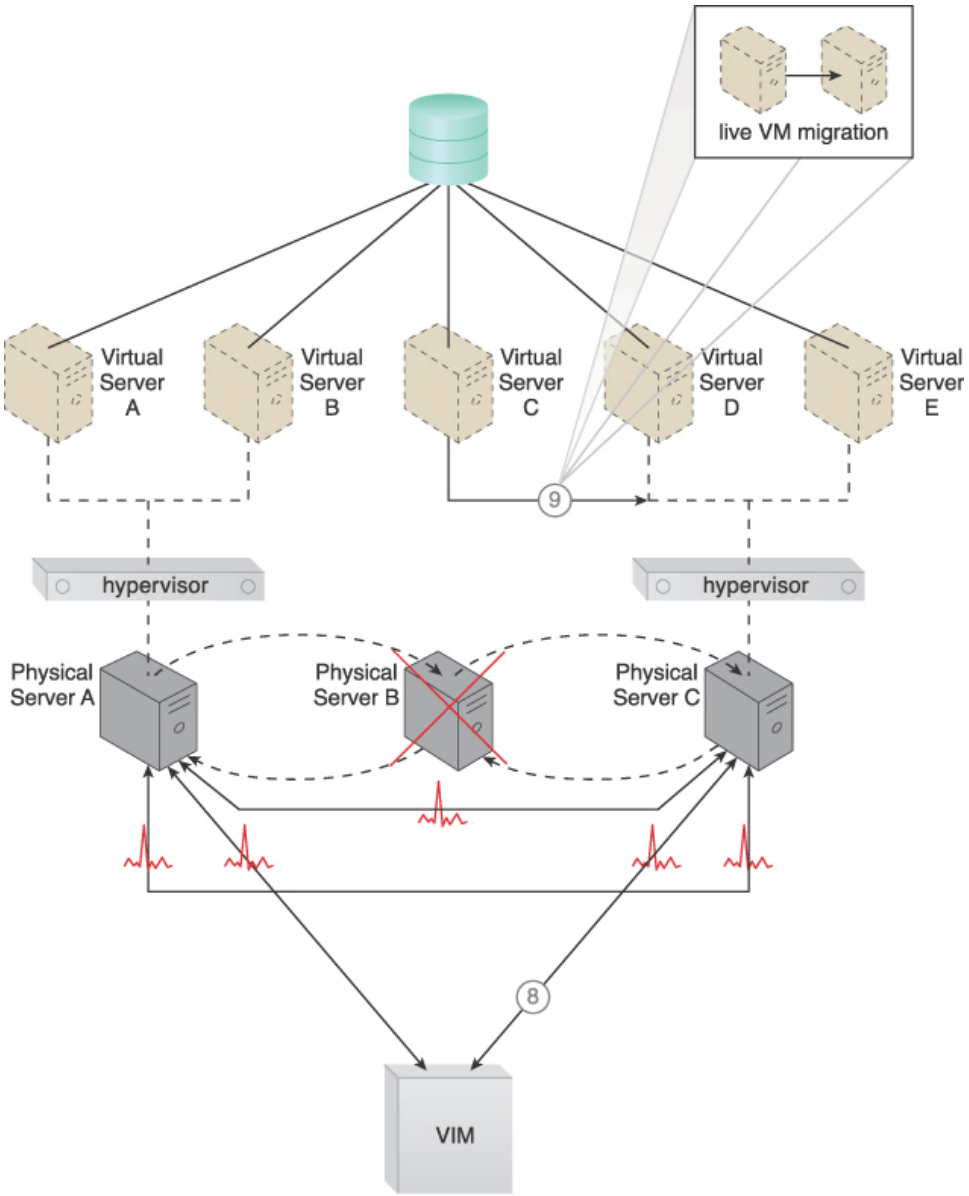

Figure 2.3.1: Hypervisor Clustering Architecture

The Figure illustrates the following:

- Hypervisors are installed on three physical servers (A, B, and C).
- These hypervisors create virtual servers.
- Configuration files for these virtual servers are stored in a shared cloud storage device accessible by all hypervisors.
- A hypervisor cluster is set up on these three physical servers using a central Virtual Infrastructure Manager (VIM).
- The physical servers exchange heartbeat messages with one another and the VIM according to a predefined schedule.
- Physical Server B fails, affecting Virtual Server C. and the other physical servers and VIM stop receiving heartbeat messages from Physical Server B.
- After evaluating the capacity of other hypervisors in the cluster, the VIM selects Physical Server C as the new host for Virtual Server C.
- Virtual Server C is then live-migrated to the hypervisor on Physical Server C, which may require a restart before it can resume normal operations.

2.4. Distributed Data Sovereignty

Distributed data sovereignty means that each country has the right to control and manage the data created within its borders. Governments can oversee how data is collected, stored, and shared within their territory.

In today's digital world, data sovereignty is becoming more important because we generate a lot of data from things like social media, online shopping, and mobile devices. Governments want to protect people's personal information and important data that could affect national security.

Rules about how data should be handled, especially personal data, can vary from one place to another. Usually, these rules require data to be physically stored in a specific location within a country. When data is in the cloud, the people or organizations using the cloud are usually responsible for following these rules, not the cloud providers.

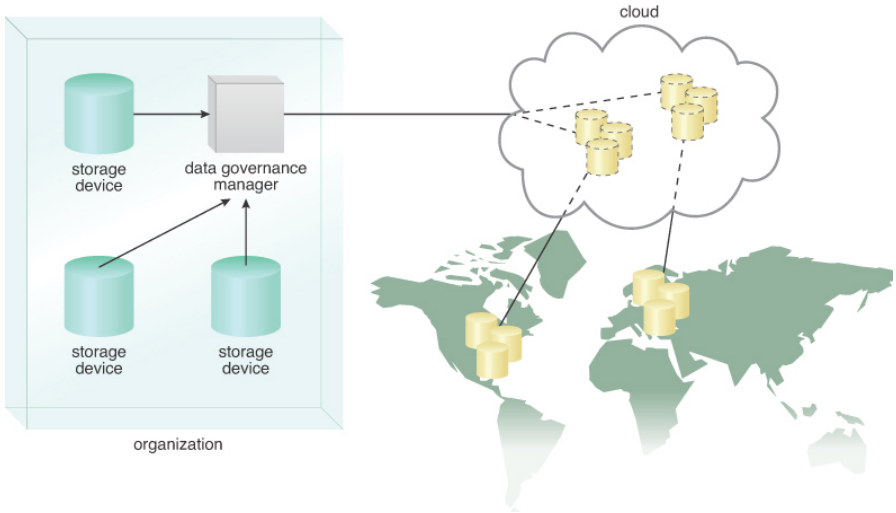

Figure 2.4.1 : Data Governance Manager ensuring data is according to regional data protection regulations.

Distributed data sovereignty has some key parts:

1. **Decentralization:** Data isn't kept in just one place; it's spread out across many servers or locations. This reduces the risk of problems if one location has issues.
2. **Ownership and Control**: People or organizations have more say in who can access their data and when they can do it.
3. **Data Localization**: Data is stored in ways that follow the rules of the place it comes from. This could mean keeping data in a specific country to follow privacy laws.
4. **Data Portability**: People or organizations should be able to easily move their data from one place to another if they want to.
5. **Security**: Using strong security measures, like encryption and access controls, to protect data and keep it private.

Distributed data sovereignty uses technologies like blockchain and decentralized storage to help people and organizations have more control over their data. These technologies help reduce the risk of data breaches and follow the law.

This idea is important as new technologies, like blockchain-based identity systems and decentralized apps, let people have more control over their digital identities and data. It deals with concerns about privacy, security, and control in the digital age, and it gives people and organizations more power to manage and protect their data.

2.5. Dynamic Scalability Architecture

Dynamic Scalability is an architectural model used in cloud computing. Based on predefined scaling parameters and variations in usage demand, it is intended to effectively allocate and deallocate IT resources (such virtual servers and storage). Making sure resources are used as efficiently as possible without requiring human intervention. The Automated Scaling Listener is the part of the system that keeps track of workload thresholds. The dynamic distribution of IT resources is initiated upon the achievement of specified thresholds.

Types of Scalabilities

2.5.1.Dynamic Horizontal Scaling:

Adding extra instances of the same resource to handle the increasing workload is known as horizontal scalability, or scale-out. For instance, more servers can be added to a web application to spread the load and guarantee responsive performance if it is receiving a lot of traffic.
Benefits

1. **High Availability**: By dividing the workload among multiple machines, horizontal scaling ensures that even in the event of a system failure, requests can still be processed.
2. **Enhanced Fault Tolerance**: This occurs when the breakdown of a single machine does not adversely affect the performance of the entire system.
3. **Scalability**: It provides simple scalability by allowing for the addition of more machines as required to handle increasing workloads.

2.5.2.Dynamic Vertical Scaling:

Increasing the processing capacity of an existing server or resource is referred to as vertical scalability, or scale-up. For instance, a virtual machine's CPU or RAM can be increased to do this. Applications that need extra memory or processing power to run well are frequently accommodated by vertical scalability.

Benefits

1. **Immediate Performance Gains:** Without the need to add new computers, vertical scaling offers instant performance gains.
2. **Simplified Management:** It may be easier to oversee one larger machine than several smaller ones.

Considerations

1. **Reliability Limitations:** A single machine's capacity to grow vertically is practically limited, which makes it less appropriate for managing extraordinarily high workloads.
2. **Cost:** Vertical scaling may be expensive, particularly if substantial hardware changes are required.
3. **System downtime:** may arise from doing updates on an already-existing machine.

2.5.3.Dynamic Relocation:

The IT resource is relocated to a host with more capacity. For instance,tape-based SAN storage device with a 4 GB per second I/O capacity, for instance, could need to have its database migrated to a disk-based SAN storage device with an 8 GB per second I/O capacity.A key feature of cloud computing is auto-scaling, which helps businesses attain scalability and elasticity while reducing operational costs.

In the cloud, "scaling" refers to dynamically modifying computing resources in response to demand. It's economical since companies don't need to make upfront investments to add or remove resources like storage and virtual computers. This adaptability is essential for managing traffic peaks and cost optimization in response to shifting business requirements.

2.6 Cloud Bursting Architecture

Cloud bursting architecture enables dynamic scaling by extending on-premises IT resources into the cloud when predefined capacity thresholds are reached. Cloud-based resources are pre-deployed but remain inactive until needed. Once no longer required, they are released, returning to the on-premises environment. This architecture provides flexibility by letting cloud users utilize resources only when there is a lot of demand. For it to work, resource replication techniques and automated scaling listeners are required. Replication keeps on-premises and cloud resources synchronized, while the listener reroutes requests to cloud resources when needed. Additionally, various other mechanisms can automate the burst in and out processes, tailored to the type of IT resource being scaled.

2.7 MultiCloud Architecture

A MultiCloud architecture is a cloud architecture that integrates two or more public clouds. This kind of architecture combines several clouds that can provide their resources via IaaS, PaaS, or SaaS, three alternative cloud delivery methods. The cloud resource administrator uses a centralized remote administration system mechanism that connects to the management systems of each individual cloud provider via their respective APIs, enabling cloud consumers to access IT resources that are distributed across multiple clouds. This enables the cloud user to simply use and access all cloud-based IT services from a single location, managing them all from one site.

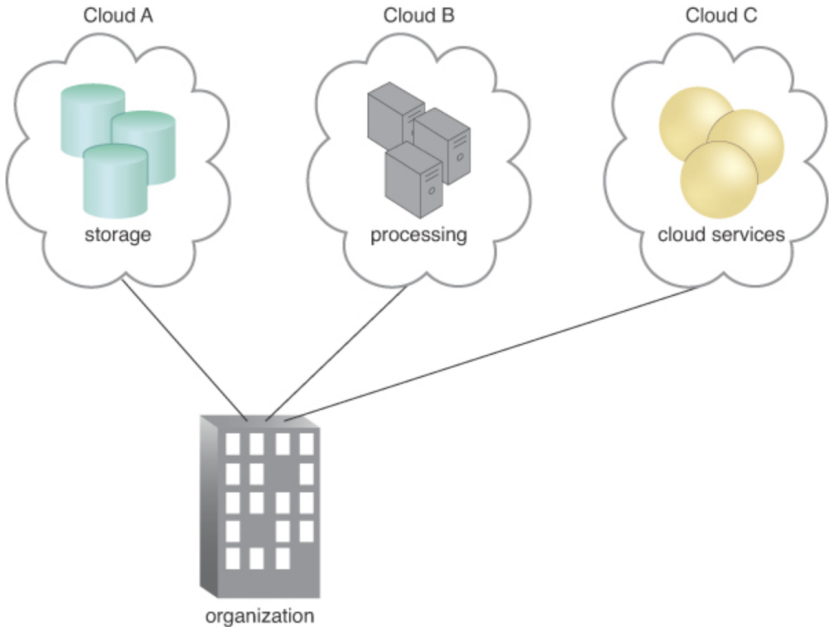

Figure 2.7.1 An organization uses different types of resources from different clouds

Types of multi cloud architecture

1. **Cloudification:** Cloudification enhances performance and elasticity by enabling on-premises applications to take advantage of cloud services from various cloud platforms.

2. **MultiCloud migration:** Depending on their requirements, this architecture style enables enterprises to shift data and apps between several cloud service providers. There are a number of reasons to do it, including to benefit from new features, save money, or enhance performance.

3. **MultiCloud refactoring:** The goal of MultiCloud refactoring is to leverage the high availability, failover capabilities, and cloud bursting that come with having many clouds. Applications must be rearchitected in order for them to be deployed in a MultiCloud

environment. Applications may not need to be modified for multiplatform deployment during multicloud relocation.

4. **MultiCloud rebinding:** Re-architecting apps in order to move them to a MultiCloud architecture is another aspect of MultiCloud rebinding.It is possible to partially deploy a re-architected application on MultiCloud infrastructure using MultiCloud rebinding in conjunction with cloud brokerage. By removing any potential single points of failure from the application, this helps increase availability.

The main goals of implementing a multicloud architecture for enterprises are to reduce vendor lock-in, which arises from relying only on one cloud provider, to boost flexibility and agility, to enhance disaster recovery capabilities, and to maximize cloud expenses.

2.8 Zero Downtime Architecture

Naturally, a physical server that supports virtual servers operates as a single point of failure for those virtual servers.In the case that a virtual server's primary physical server host fails, the zero downtime architecture creates an advanced failover system that enables virtual servers to be dynamically transferred to other physical server hosts. A fault tolerance system that can seamlessly move activity from one physical server to another is used to manage a group of many physical servers. Usually, a major feature of this type of high-availability cloud architecture is the live virtual machine migration component.

The following mechanisms can be part of this architecture

- Audit Monitor: Verifies data compliance during virtual server relocation.
- Cloud Usage Monitor: Tracks resource usage to prevent capacity overages.
- Hypervisor: Hosts virtual servers on physical servers.
- Logical Network Perimeter: Maintains isolation boundaries after server relocation.
- Resource Cluster: Forms active-active clusters to enhance IT resource availability.
- Resource Replication: Creates new instances in case of primary server failure.

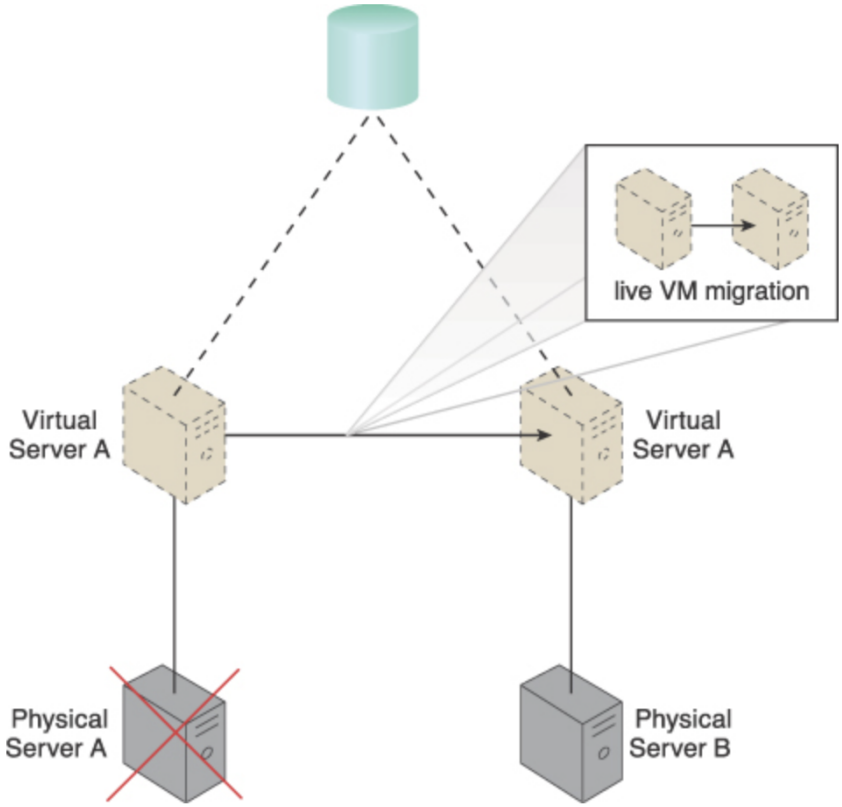

Figure 2.8.1 Virtual Server A hosted by physical Server A is moved to Physical Server B by a live VM migration program.

2.9 Dynamic Failure Detection and Recovery Architecture

A resilient watchdog system is established by the dynamic failure detection and recovery architecture to monitor and react to a variety of predetermined failure scenarios. When a failure condition arises that this system is unable to handle automatically, it alerts users and escalates the issue. Resilient watchdog systems' primary functions include monitoring, determining when an event occurs, responding to an event, reporting, and escalation.For every IT resource, sequential recovery policies can be established that specify what actions the intelligent watchdog monitor must do in the event of a failure scenario. When an issue is escalated, the intelligent watchdog monitor frequently runs a batch file, sends a console message, sends a text message, sends an email, sends an SNMP trap, and logs a ticket.

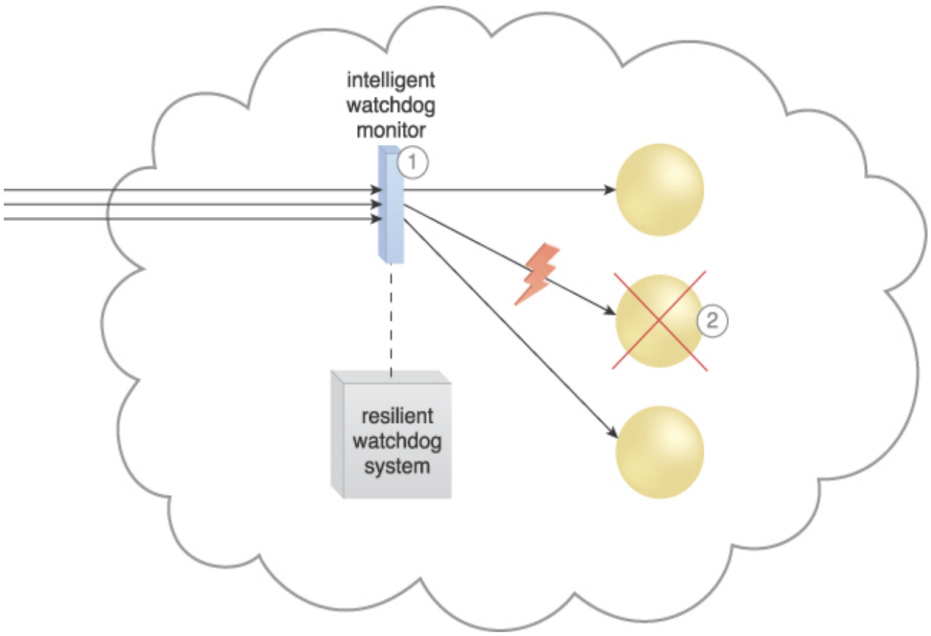

Figure 2.9.1 The intelligent watchdog monitor keeps track of cloud consumer requests (1) and detects that a cloud service has failed (2).

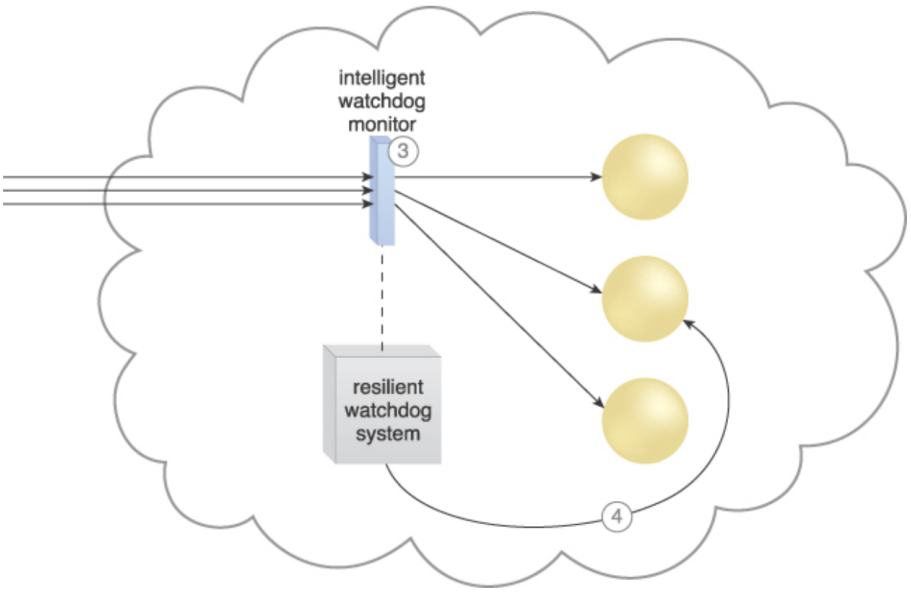

Figure 2.9.2 The intelligent watchdog monitor notifies the watchdog system (3), which restores the cloud service based on predefined policies. The cloud service resumes its runtime operation (4).

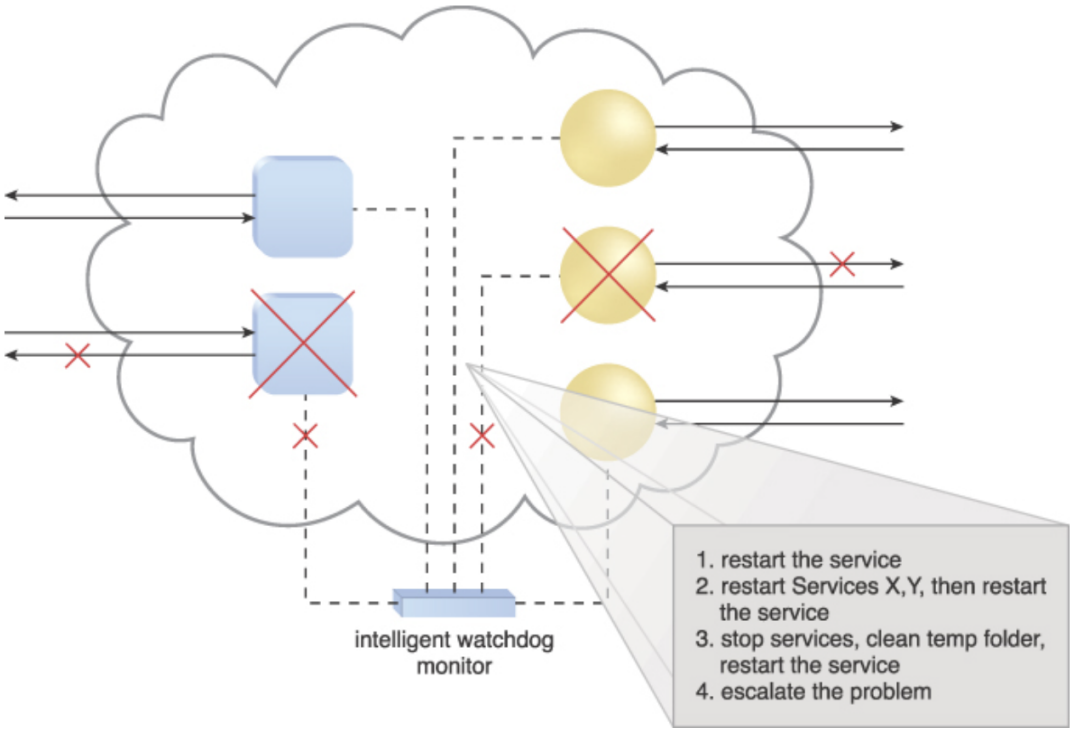

Figure 2.9.3 The intelligent watchdog monitor notifies the watchdog system (3), which restores the cloud service based on predefined policies. The cloud service resumes its runtime operation (4).

The following mechanisms can also be incorporated into this architectural model:

Audit monitor: This mechanism is employed to monitor if the process of data recovery complies with applicable laws and policies.
Failover System: When trying to recover lost IT resources, the failover system mechanism is employed.
SLA Management System and SLA Monitor: The system often depends on the data that is handled and controlled by these mechanisms since the functionality attained through the use of this design is closely linked to SLA guarantees.

2.10 Elastic Resource Capacity Architecture

Traditional architectures of various solutions are based on an approach where applications make resource reservations on an end-to-end and per-session basis. These approaches are useful for real time applications like video-conferencing which are long-lived and require delay and jitter bounds. However, with the advent of the world wide web and the cloud, the utilization of resources has become short-lived. Moreover, we need to typically tolerate variations in the utilization of resources.

Elasticity aims at matching the amount of resource allocated to a service with the amount of resource it actually requires, avoiding over- or under-provisioning.

The elastic resource capacity architecture dynamically provisions the virtual servers using a system that allocates and reclaims CPUs and RAM in a response to the demand of resources, enabling automatic scaling up and down both horizontally and vertically . (Figures 2.10.1 and 2.10.2).

Resource pools, managed by scaling technology, dynamically retrieve and allocate CPU and RAM resources for virtual servers at runtime. This monitoring system adjusts resource allocation, enhancing processing power before capacity limits are reached. The virtual server and its hosted applications are vertically scaled to meet demand.
In this cloud architecture, an automation engine script routes scaling requests through the VIM (“Understanding Virtual Infrastructure Management (VIM)”) instead of directly to the hypervisor.

Additional mechanisms in this architecture include:

- Cloud Usage Monitor: Collects resource usage data for defining future processing capacity thresholds of virtual servers before, during, and after scaling.
- Pay-Per-Use Monitor: Tracks fluctuating resource usage costs with elastic provisioning.
- Resource Replication: Generates new instances of scaled IT resources within this architectural model.

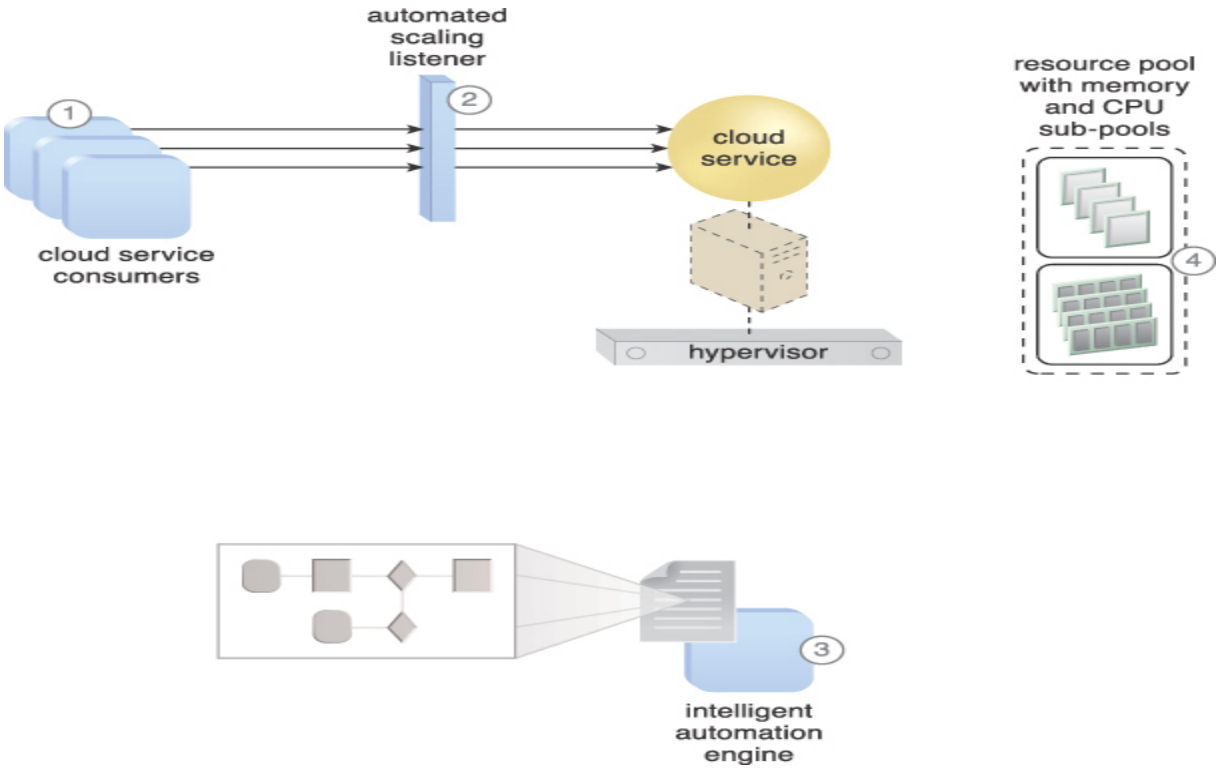

Figure 2.10.1 Cloud service consumers are actively sending requests to a cloud service (1), which are monitored by an automated scaling listener (2). An intelligent automation engine script is deployed with workflow logic (3) that is capable of notifying the resource pool using allocation requests (4)

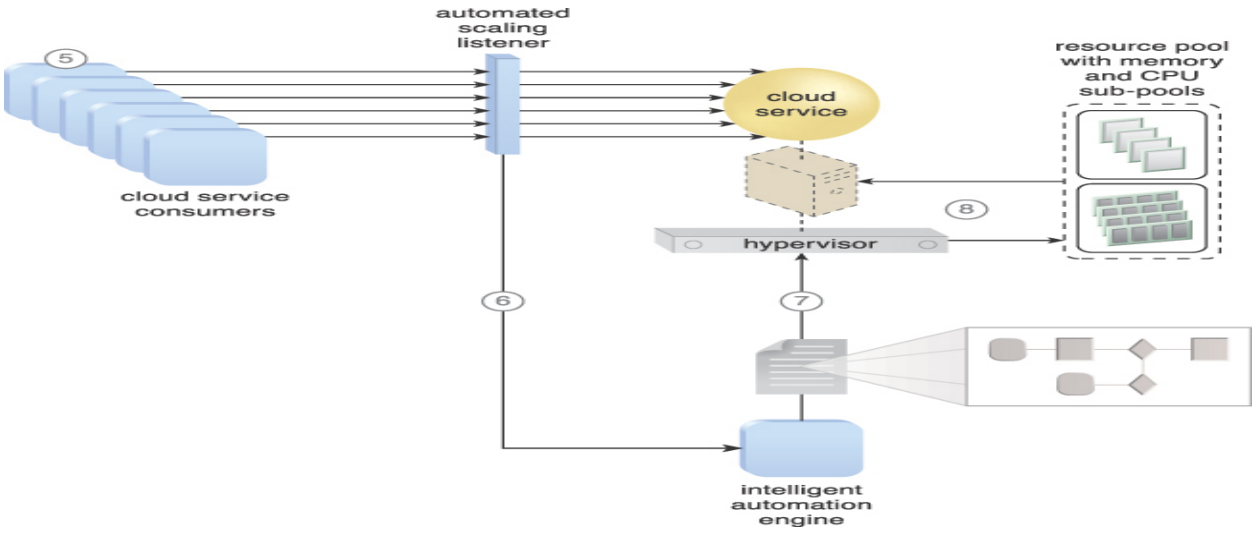

Figure 2.10.2 Cloud service consumer requests increase (5), causing the automated scaling listener to signal the intelligent automation engine to execute the script (6). The script runs the workflow logic that signals the hypervisor to allocate more IT resources from the resource pools (7). The hypervisor allocates additional CPU and RAM to the virtual server, enabling the increased workload to be handled (8).

2.11 Redundant Storage Architecture

Cloud storage devices can fail due to network issues, hardware malfunctions, or security breaches. Compromised devices can disrupt and impact the availability of services, applications, and infrastructure within the cloud.

Storage Service Gateway

The storage service gateway serves as the external interface for cloud storage services, adept at automatically redirecting consumer requests when the data's location has changed.

Redundant storage architecture involves a secondary duplicate cloud storage device synchronized with the primary one for failover. In case of primary device failure, a storage service gateway redirects consumer requests to the secondary device Figure 2.11.1 and Figure 2.11.2

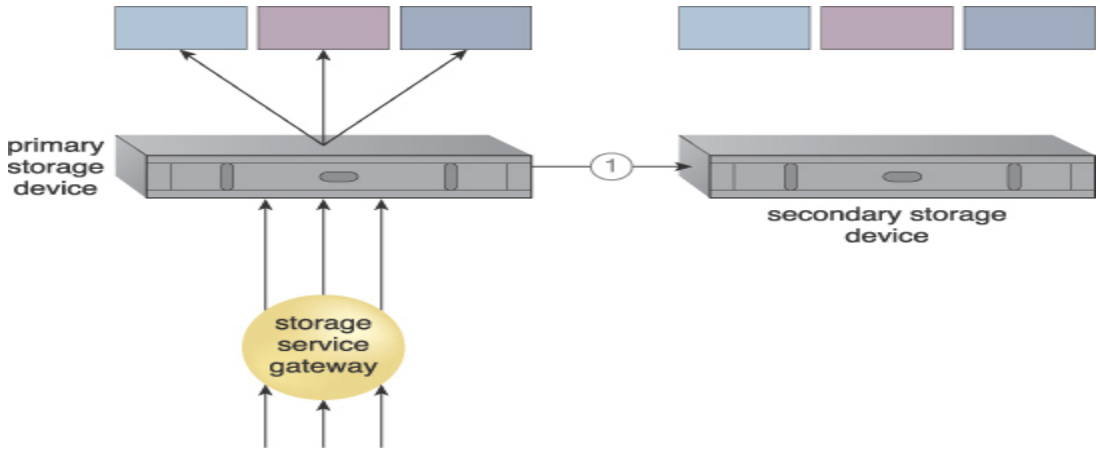

Figure 2.11.1 The primary cloud storage device is routinely replicated to the secondary cloud storage device (1).

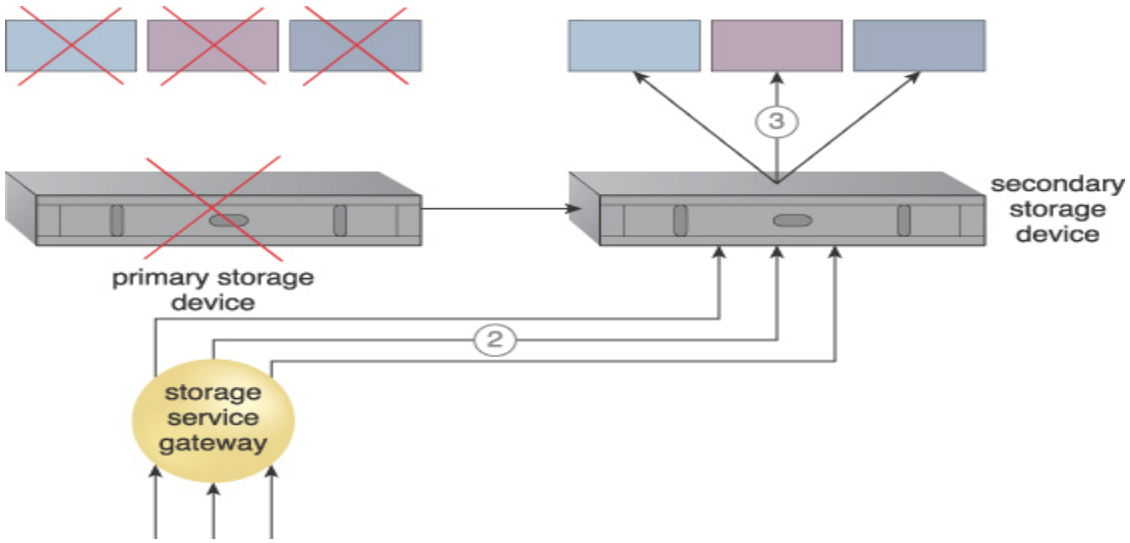

Figure 2.11.2 The primary storage becomes unavailable and the storage service gateway forwards the cloud consumer requests to the secondary storage device (2). The secondary storage device forwards the requests to the LUNs(A logical unit number is a logical drive that represents a partition of a physical drive.), allowing cloud consumers to continue to access their data (3).

This cloud architecture primarily relies on a storage replication system that keeps the primary cloud storage device synchronized with its duplicate secondary cloud storage devices (Figure 2.11.3).

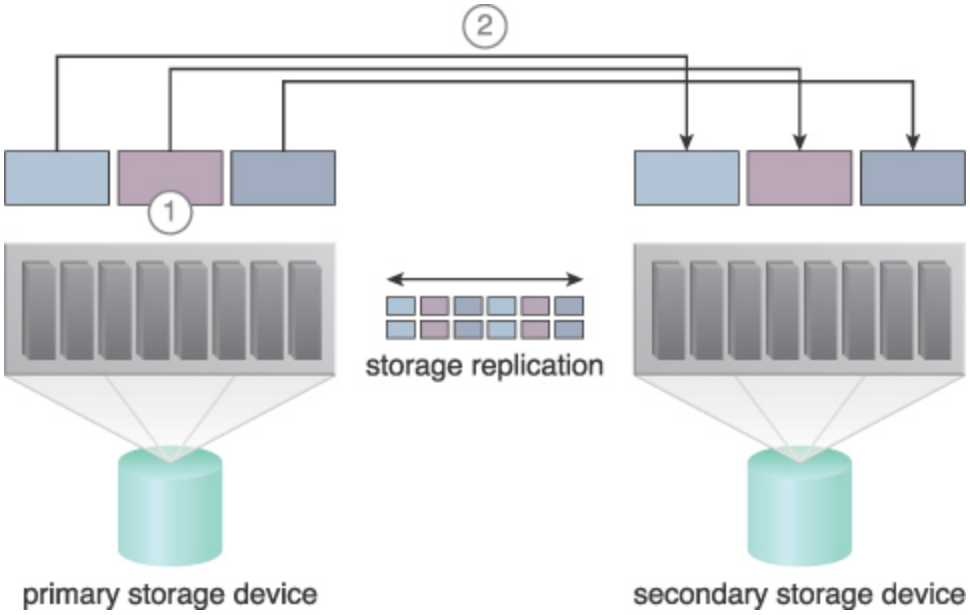

Figure 2.11.3 storage replication system

Cloud providers may position secondary storage in different regions for cost-efficiency, but this may raise legal concerns for certain data types. The location of secondary storage dictates the synchronization protocol, as some methods have distance limitations.

Certain cloud providers employ dual-array storage devices for enhanced redundancy, positioning secondary storage in different locations for load balancing and disaster recovery. They might lease a network connection from a third-party provider to facilitate replication between these devices.

2.12 Virtual Server Clustering Architecture

The virtual server clustering architecture deploys clusters of virtual servers on physical hosts using hypervisors. It capitalizes on cloud efficiency, resiliency, and scalability through virtualization. It allows creation of diverse clusters for various purposes like big data, service architectures, NoSQL databases, and container management platforms.(Figure 2.12.1)

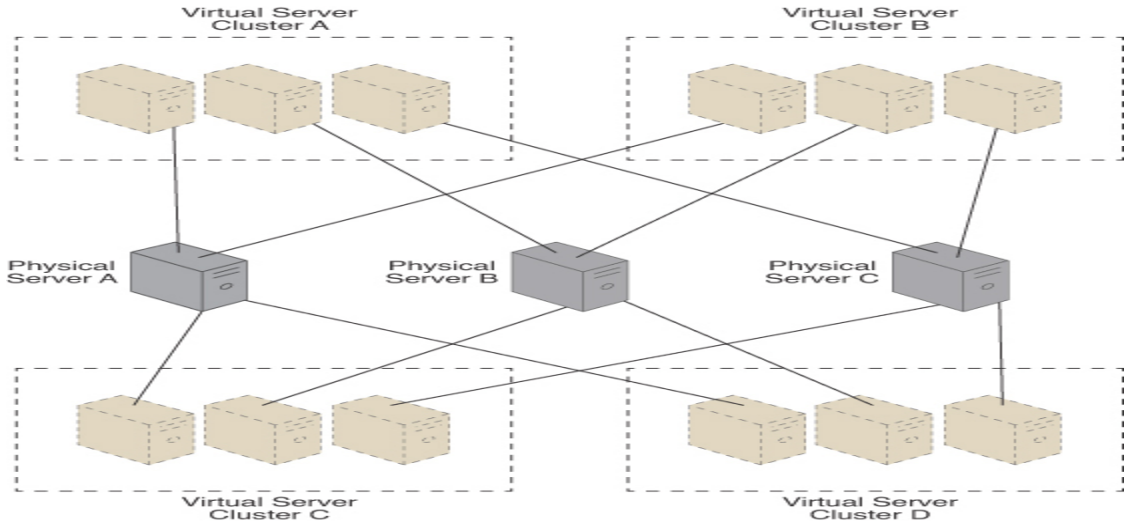

Figure 2.12.1 Physical Servers A, B and C are running hypervisors that allow multiple virtual servers to be hosted on each. (These virtual servers are then configured by a resource cluster mechanism into clusters of virtual servers.)

Additional components in this architecture include:

- Logical Network Perimeter: Secures communication within the virtual server cluster, ensuring interconnected nodes can communicate securely.

- Resource Replication: Ensures virtual servers share status and updates within the cluster, such as creating or deleting virtual switches, for consistent information across all servers.

2.13 Rapid Provisioning Architecture

The manual provisioning process in traditional IT involves human-driven tasks, prone to errors, especially in cloud environments serving higher volumes of clients demanding larger IT resources. For instance, installing and configuring multiple servers with various applications requires significant time and risks human error.

Rapid provisioning architecture automates IT resource provisioning, relying on an intricate system with components like automated provisioning programs, engines, and provisioning templates. Additional components, such as server templates, application packages, custom scripts, and sequence management tools, coordinate and automate the provisioning process. These components streamline tasks like server and application deployment, configuration, and maintenance, ensuring efficiency and reducing human error in cloud environments (Figure 2.13.1).

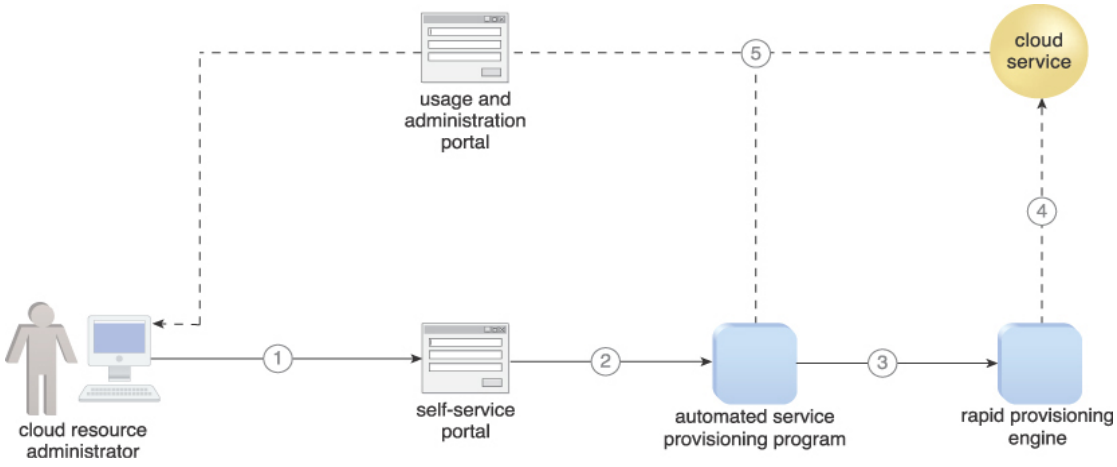

Figure 2.13.1 A cloud resource administrator requests a new cloud service through the self-service portal (1). The self-service portal passes the request to the automated service provisioning program installed on the virtual server (2), which passes the necessary tasks to be performed to the rapid provisioning engine (3). The rapid provisioning engine announces when the new cloud service is ready (4). The automated service provisioning program finalizes and publishes the cloud service on the usage and administration portal for cloud consumer access (5).

The rapid provisioning engine functions through a step-by-step process:

- Cloud consumers request a new server via the self-service portal.
- Sequence manager forwards the request to the deployment engine for OS preparation.
- Deployment engine uses templates for OS provisioning (virtual/physical).
- If the requested OS image is unavailable, the regular deployment process is executed.
- Deployment engine signals sequence manager upon OS readiness and updates logs.
- Operating system baseline is applied to the provisioned OS as requested.
- Deployment engine deploys applications and informs the sequence manager.
- Application configuration baseline is applied by the deployment engine.
- Storage mechanism houses application baseline info, templates, and scripts.
- Hypervisor swiftly creates, deploys, and hosts virtual servers.
- Resource replication generates replicated IT resource instances as needed for rapid provisioning.

PART 3: SOFTWARE IN CLOUD COMPUTING

3.1 Virtualization Technologies

Virtualization is the technology that enables a physical IT resource to provide multiple virtual images of itself so that its underlying processing capabilities can be shared by multiple solutions.

Virtualization plays a crucial role in simplifying resource management within cloud computing environments. By abstracting and isolating the underlying hardware and networking resources, it enhances the management of cloud components. This technology increases the security of cloud computing by safeguarding the integrity of guest virtual machines and the various components of the virtualized infrastructure.

Virtualization allows for the creation of individual, dedicated virtual servers, each offering a fresh and isolated copy, or image, of the hosting environment - often referred to as a guest operating system. These virtual servers can cater to different sets of consumer applications or services without necessitating any knowledge of the underlying physical server's existence or operations.

The flexibility of virtualization permits the scaling of physical servers to accommodate fluctuating consumer usage demands. Virtualized machines can be seamlessly scaled up or down on demand, ensuring reliability, efficient resource sharing, and high utilization of pooled resources. This technology also enables rapid provisioning and workload isolation, empowering the cloud infrastructure to swiftly adapt to changing demands in a highly dynamic environment.

There are several approaches to virtualisation in cloud environments to provide various capabilities and address specific use cases. Some of them are discussed below.

3.1.1 Hypervisors

A hypervisor is a software that creates and runs virtual machines (VMs) on physical hardware. It allows multiple operating systems to run concurrently on a single physical machine (Figure 3.1.1.1).

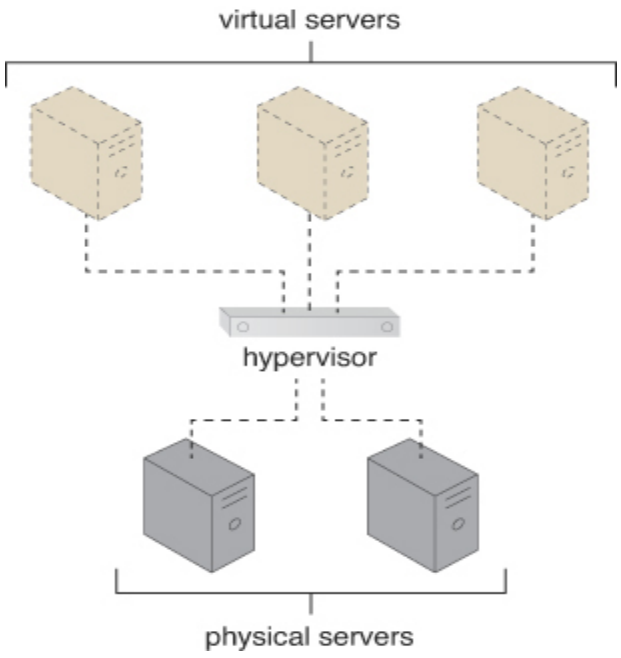

Figure 3.1.1.1 Three virtual servers created and run by a hypervisor that exists on two physical servers.

There are two types of Virtualization utilizing hypervisor :

Type 1 : Physical server does not have an operating system installed. Hypervisor, which is installed on the physical server is responsible for creating the virtual servers and providing them with virtualized operating system environments (Figure 3.1.1.2).

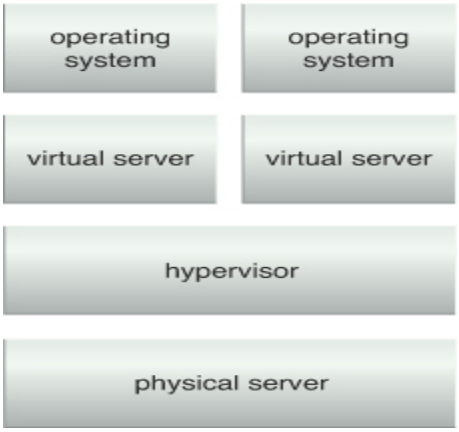

Figure 3.1.1.2 The physical server hosts only the hypervisor that creates virtual servers, each with its own operating system.

In Type 2 Virtualization the physical server has an operating system installed and the hypervisor remains responsible for creating the virtual servers and providing them with their virtualized operating system environments Figure 3.1.1.3.

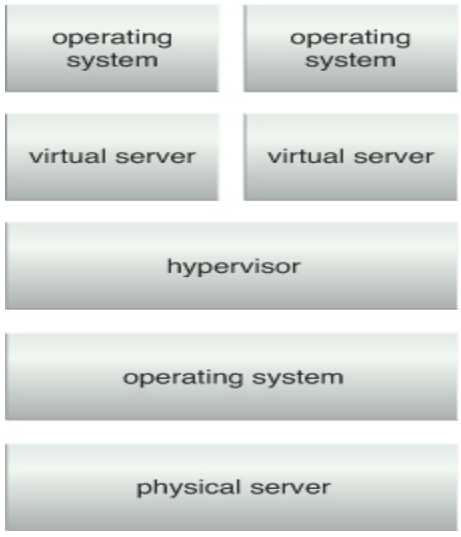

Figure 3.1.1.3.The physical server hosts its own operating system as well as a hypervisor that creates virtual servers with their own operating system environments.

3.1.2 Containerization

It is a form of lightweight virtualization where applications and their dependencies are packaged together as containers. These containers share the same host operating system kernel but are isolated from each other. Containers package the application, along with its dependencies, libraries, and configurations, so they don't need an operating system(Figure 3.1.2.1) .

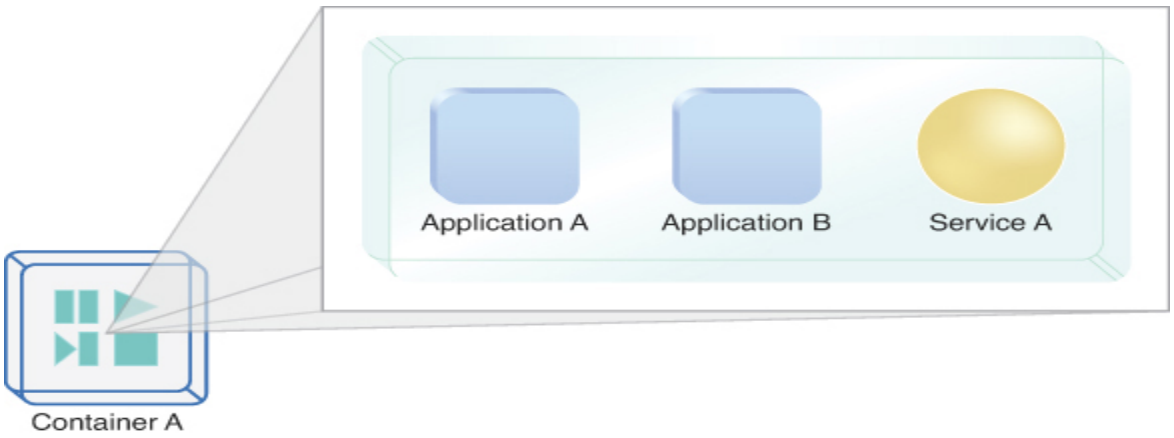

Figure 3.1.2.1 The symbol on the left is the container icon. The symbol on the right is also used to represent a container and to show its contents.

3.1.2.1 Understanding Containers

A ***container image*** is similar to a predefined template that is used to create deployed containers.

The ***container engine*** is deployed in a physical or virtual server's operating system from where it can abstract the resources required for a given container and is responsible for creating containers based on predefined container images (Figure 3.1.2.1).

Its implementation is is organized into two planes as follows

- Management Plane – the GUI and command-line tools made available to enable human administrators to configure and maintain the container engine environment
- Control Plane – all remaining container engine functions and features that the container engine carries out automatically and in response to settings and commands issued via the management plane

A given container engine can create multiple containers

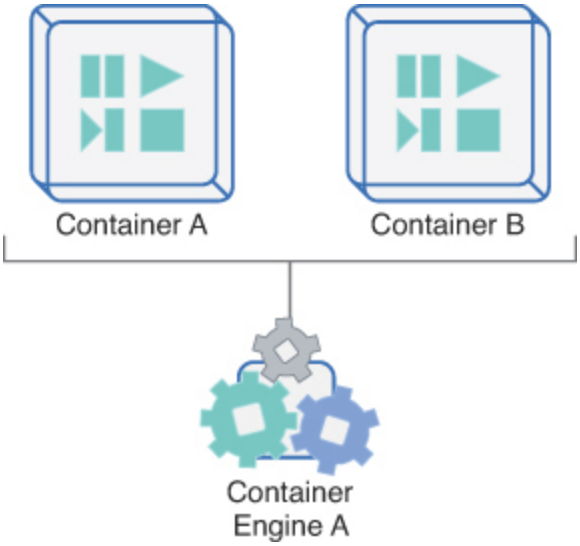

Figure 3.1.2.1 A container engine creating two different containers.

A ***pod**,* also known as a logical pod container, is a special type of system container that can be used to host a single container or a group of containers that have shared storage and/or network resources, and also share the same configuration that determines how the containers are to be run. Containers inside a pod can find and discover each other via the host on which the pod is deployed and can communicate with each other using standard interprocess communication methods, such as shared memory. They also share a file system, dataset, or data storage device.

A ***Host***, which is an environment where a container can be deployed is a server or a node and Multiple containers can be deployed and run on a single host.

Hosts commonly exist as physical servers, but a host can also be a virtual server. When a container is deployed on a virtual server, it is considered a form of *nested virtualization* because one virtualized system is deployed on another.

Common types of host clusters include :

Load-balanced Cluster - Specializes in distributing workloads among hosts to increase resource capacity while preserving the centralization of resource management.
High Availability Cluster - Specializes in maintaining the availability of systems in the event of multiple failures. It provides redundant implementations of most or all of the resources, and monitors, failure conditions and automatically redirects workloads from failed environments.
Scaling cluster- Specializes in scaling horizontally and vertically.

3.1.2.2 Container Networks

Container Engine generates container images, deploys and runs containers on a host. All the related containers within a host can communicate with each other using a host network .

Related containers and container engines on different hosts can communicate with each other via an overlay network.

3.1.2.3 Containerization on Physical servers-

When deploying containers on a physical server, the containerization platform requires no virtualization environment since virtual servers are not required. The underlying physical server has an operating system installed, and the containerization platform can create containers that each only abstract the subset of the operating system relevant to the software programs it hosts (Figure 3.1.2.3.1).

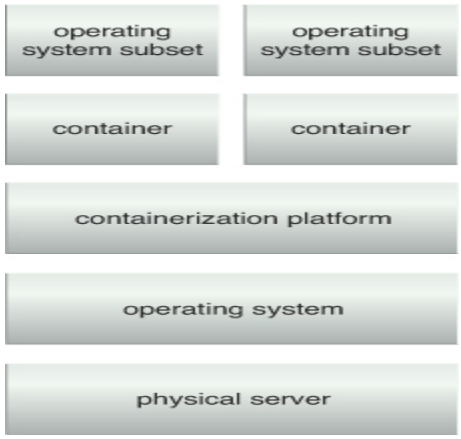

Figure 3.1.2.3.1 A physical server with an operating system hosts a containerization platform that creates containers, each with an environment that has only a subset of the underlying operating system.

3.1.2.4 Containerization on Virtual Servers

When deploying containers on one or more virtual servers, the containerization platform can be implemented on a Type 1 virtualization environment (Figure 3.1.2.4.1) or a Type 2 virtualization environment with a hypervisor (Figure 3.1.4.1.2). Both types of virtualization environments allow for the creation of virtual servers that can host containerization engines.

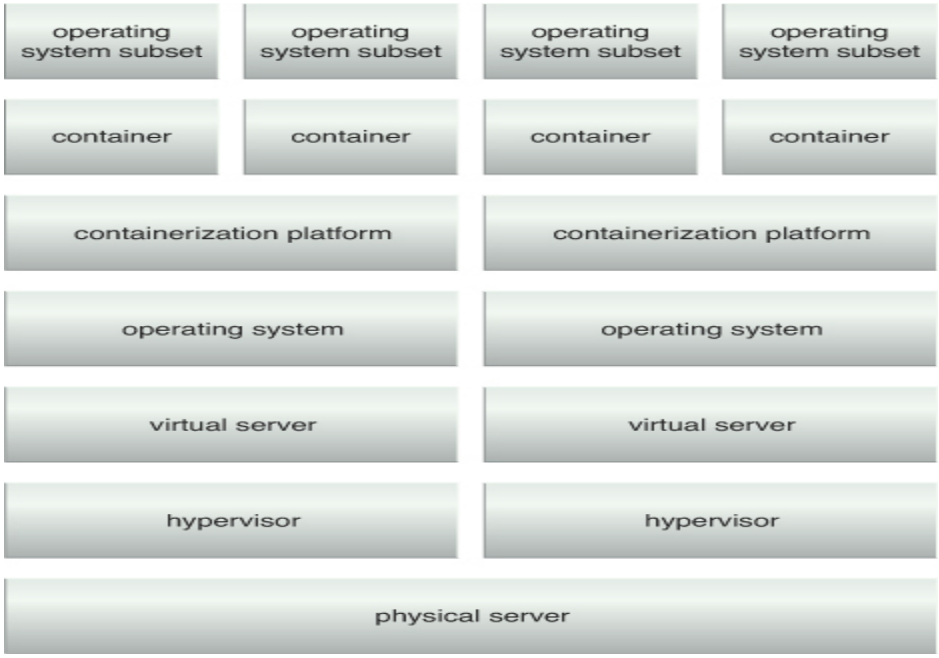

Figure 3.1.2.4.1 A physical server with no operating system hosts a hypervisor that creates virtual servers with operating systems, each of which hosts a containerization platform that can create containers that only have an operating system subset.

The motivation behind deploying containers on virtual servers is often related to security vulnerabilities that exist when the physical server has an operating system installed. As a result, the Type 1 virtualization environment is more common in most production environments. Type 2 virtualization is typically used in development environments when containerized solutions are being built and tested.

Type 2 virtualization can also be used for smaller solutions or for smaller organizations when the underlying physical server needs an operating system in order to host additional programs and systems alongside the containerization platform.

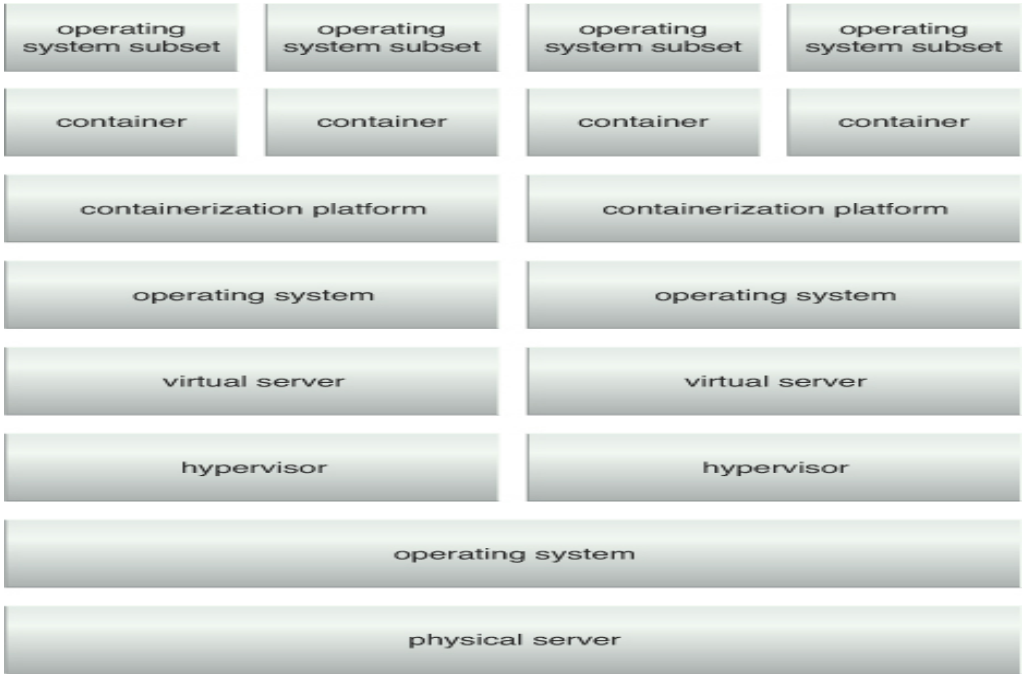

Figure 3.1.2.4.2 A physical server with an operating system hosts a hypervisor that creates virtual server environments with their own operating systems. Each virtual server hosts a containerization platform that creates containers that host a subset of the operating system.

3.2 Container Orchestration

The process of automating the deployment, scaling, and management of containerized applications in a distributed computing environment is known as *container orchestration.* A container orchestrator typically consists of several components that work together. Some of the key components of a container orchestrator are:

- Deploying containers across multiple nodes in a cluster and ensuring they are properly configured and networked.
- Distributes traffic across multiple containers running the same application to ensure high availability and scalability .
- Automatically scales up, down, in, or out the number of containers running an application based on demand, to ensure optimal resource utilization and cost efficiency.
- Monitors the health of containers and can automatically restart failed containers or replace them with healthy ones.
- Maintains a service registry so applications can discover and communicate across the network .
- Provides each container with a unique IP address and routes network traffic between containers.

3.2.1 Container Network Addresses

Each deployed container receives an IP address that enables it to participate in a container network. If a container needs to participate in multiple container networks, it will require a separate network address for each container network. For example, if a container hosting a software program is being reused across two container networks then that container will need two network addresses.

Network addresses are usually assigned by the container engine subsequent to container deployment. They can also be manually assigned by the administrator in a deployment package. Containers residing in the same pod share the same network address and are individually identified through different network ports.

3.2.2 Common Containerization Technologies

Docker and Kubernetes are essential technologies in the realm of containerization and orchestration, enabling efficient deployment, management, and scaling of applications within a computing environment.

Docker:

- Docker is a platform for developing, shipping, and running applications using containerization technology. It allows developers to create, deploy, and run applications within containers. A container packages an application and its dependencies, ensuring it runs consistently across environments.
- Key features of Docker:
    - Containerization: It encapsulates applications and their dependencies, creating lightweight, portable, and self-sufficient units.
    - Portability: Docker containers can run on any system that supports the Docker engine, maintaining consistency from development to production.
    - Efficiency: It uses system resources more efficiently compared to virtual machines and enables faster application deployment.

Kubernetes:

- Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It was originally developed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF).
- Key features of Kubernetes:
    - Orchestration: It automates the management of containers, ensuring the efficient deployment and scaling of applications.
    - Scaling: Kubernetes can automatically scale the number of containers based on demand, ensuring applications are always available and responsive.

- High Availability: It provides mechanisms for ensuring that applications remain available even in the case of hardware or software failures.

3.2.3 Other Virtualization technologies

Serverless computing abstracts away the infrastructure management. It allows developers to run code without provisioning or managing servers explicitly. Services like AWS Lambda, Google Cloud Functions, and Azure Functions fall under this category.

NFV(Network Function Virtualization) virtualized network services traditionally run on dedicated hardware, making it easier to manage and scale network functions. It is particularly relevant in telecommunications and networking within the cloud.

3.3. Cloud-native Application Development

3.3.1. What is Cloud Native?

The traditional methodology for creating a software system is the monolithic approach, wherein all the required functionalities are enclosed as a single unit. Every component in a software system behaves differently, some might require more resources than the rest. In this case it becomes difficult to scale that particular component separately as everything is enclosed as a single unit. Moreover, scaling the entire monolithic application for a single component is not a viable solution.

In contrast to the traditional approach we have the cloud-native approach. Cloud-native applications are software systems composed of numerous small, interrelated services known as Microservices.

This segmentation enhances the agility of cloud-native applications, as these microservices operate autonomously and require minimal computational resources to function.

Comparing Cloud-Native Applications to Traditional Enterprise Software:

- Traditional enterprise software development relied on less adaptable methods, where developers typically worked on a sizable batch of software functionalities before releasing them for testing. Consequently, traditional enterprise applications exhibited longer deployment times and lacked scalability.
- In contrast, cloud-native applications adopt a collaborative approach and offer high scalability across various platforms. Developers employ software tools to extensively automate the building, testing, and deployment processes within cloud-native applications. This enables the rapid setup, deployment, or replication of microservices, a level of agility unattainable with traditional applications.

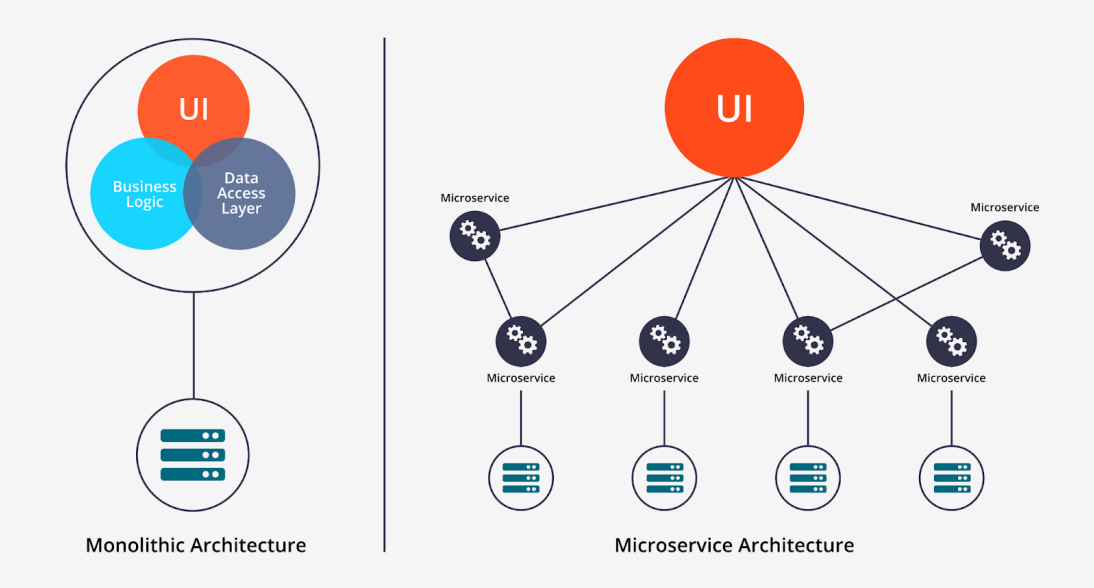

Figure 3.3.1.1 Monolithic vs Microservices Architecture

Cloud-native applications are intentionally engineered to maximize the scalability, flexibility, resiliency, and adaptability offered by cloud platforms. According to the Cloud Native Computing Foundation (CNCF), cloud-native technologies empower organizations to build and run scalable applications across diverse cloud environments. This approach is exemplified by features such as containers, service meshes, microservices, immutable infrastructure, and declarative application programming interfaces (APIs). These characteristics foster the development of loosely interconnected systems that are robust, easy to oversee, and offer enhanced observability, enabling engineers to effect impactful changes frequently and with minimal effort.

3.3.2. Key Characteristics for Building Cloud-Native Applications

Cloud-native applications are constructed upon a foundation of key building blocks and principles that enable them to fully exploit the advantages of cloud computing and distributed systems. These building blocks encompass:

1. **Microservices**: Microservices are compact, independently deployable components that fulfill specific functions within an application. They facilitate modularity, easing development, deployment, and scaling.
2. **Containers**: Containers, often managed with Docker, encapsulate application code and its dependencies, ensuring consistency across development, testing, and production environments.
3. **API based Communication:** It stands for Application Programming Interface, is a mechanism employed by two or more software programs to facilitate the exchange of data. API informs you about the data a microservice requires and the results it can provide.
4. **DevOps & Continuous Integration and Continuous Delivery (CI/CD)**: DevOps is a set of practices, principles, and cultural philosophies that emphasize collaboration and

communication between software development (Dev) and IT operations (Ops) teams. CI/CD pipelines automate the testing, integration, and deployment of code changes, enabling frequent and reliable releases.

Tools like Kubernetes automate container deployment, scaling, and management, offering features like load balancing, auto-scaling, and self-healing.

3.3.3. Benefits of Cloud-Native Applications:

1. **Scalability**: Cloud-native applications can seamlessly scale horizontally, enabling them to handle sudden spikes in traffic or growth without major overhauls.
2. **Agility**: Designed for rapid development and deployment, cloud-native applications leverage microservices, containerization, and CI/CD pipelines to facilitate faster and more frequent updates, enabling organizations to promptly respond to market changes and customer demands.
3. **Cost Efficiency**: Cloud-native architectures optimize resource utilization, reducing infrastructure costs. Automatic scaling down during periods of low demand can result in cost savings compared to traditional on-premises solutions.
4. **Resilience and High Availability:** Cloud-native applications are engineered for resilience and fault-tolerance, with the ability to recover from failures. Many cloud providers offer high-availability features, minimizing downtime and enhancing system reliability.
5. **Improved Resource Utilization:** Containerization and container orchestration platforms enable efficient resource utilization by packaging applications into containers, reducing overhead and optimizing hardware usage.
6. **Portability**: Cloud-native applications are often more portable, as they are less reliant on specific hardware or infrastructure configurations. This facilitates easier migration between different cloud providers or on-premises environments.
7. **Enhanced Security**: Many cloud providers offer robust security features, which cloud-native applications can leverage. Furthermore, a well-architected cloud-native application can implement security best practices, including zero-trust architecture and secure APIs.

3.3.4. Challenges of Cloud-Native App Development

The adoption of cloud-native practices has introduced new challenges for developers, operations teams, and organizations as a whole. These challenges include:

1. Managing various software versions across different cloud providers.
2. Navigating the complexities of scaling applications.
3. Dealing with increased complexity due to the integration of additional services and components.
4. Guaranteeing efficient resource usage, taking into account the cost factors associated with the pay-as-you-go model of cloud services.
5. Fostering seamless collaboration among all application components.

3.4.DevOps and Continuous Integration/Continuous Deployment (CI/CD) in the Cloud

The goal of DevOps is to streamline and automate the software delivery process, enabling organizations to develop, test, and deploy software more rapidly and reliably.

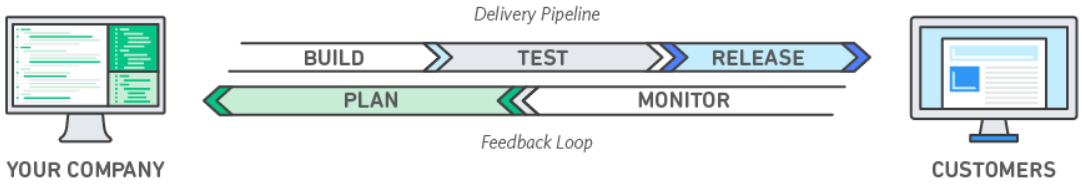

Figure 3.4.1 DevOps Model

3.4.1. Best DevOps Practices

Here are some essential ways that help organizations become more innovative by making their software development smoother and automated. These practices often involve using the right tools.

1. **Making small, frequent updates instead of large, occasional updates:** This speeds up innovation for organizations, and these updates are usually small improvements made regularly, rather than occasional large ones as seen in traditional methods. This reduces the risk of problems and helps teams fix errors faster. Organizations following a DevOps approach deploy updates more often than those using traditional methods.
2. **Adopting a microservices architecture:** Using a microservices architecture makes applications more flexible and encourages quicker innovation. This architecture breaks down large, complex systems into smaller, independent projects. Each part of the application, called a service, serves a specific purpose and operates independently. This simplifies the process of updating applications. When small, agile teams take charge of each service, organizations can work faster.
3. **Implementing CI-CD pipelines:** Combining microservices with more frequent updates can lead to operational challenges. DevOps practices like Continuous Integration and Continuous Delivery solve these problems and allow organizations to deliver updates quickly and reliably.
4. **Maintaining infrastructure through Infrastructure as Code (IaC):** Infrastructure automation practices, such as using code to manage infrastructure and configurations, help keep computer resources flexible and responsive to changes.
5. **Setting up monitoring and logging:** Monitoring and Logging tools help engineers track the performance of applications and infrastructure so they can quickly respond to issues.

By adopting these practices, organizations can provide faster and more dependable updates to their customers, fostering innovation and better meeting their needs.

PART 4: UNDERSTANDING CLOUD SECURITY and CYBERSECURITY

Cloud security, which is often referred to as cloud computing security, is the process of protecting data, applications, and infrastructure that are hosted on the cloud from threats and attacks.

Why is cloud security important?

Important business applications and data are moving to reputable third-party cloud service providers (CSP) as the use of enterprise cloud services keeps growing. Although the majority of large CSPs include monitoring and alerting capabilities in their baseline cybersecurity solutions, internal IT security teams may run into scenarios where these tools fall short of the enterprise's requirements for comprehensive cybersecurity coverage. This raises the possibility of data loss and security breaches by exposing possible cybersecurity flaws.

Shared Responsibility model

Cloud service providers follow a shared security responsibility model, which is a framework that outlines which security tasks that CSPs are responsible for and the security tasks which are customers responsibility, although customers shift workloads, data, containers, and apps to the cloud. Under this shared responsibility framework, security ownership must be clearly specified between both the parties. Both parties must retain total security control over the assets, procedures, and functions that they own. By sharing security responsibilities, cloud environments can be maintained in a secure environment with less operational overhead.

The shared responsibility security framework has several definitions depending on the service provider and whether you are using platform-as-a-service (PaaS) or infrastructure-as-a-service (IaaS).
AWS claims it is responsible for "protecting the hardware, software, networking, and facilities that run AWS Cloud services" in the AWS Shared Security model.
Microsoft Azure asserts that "physical hosts, networks, and data centers" are under its security ownership.
Which services you choose will determine your retained security responsibilities, according to both AWS and Azure.

Customers are always in charge of protecting the things that are directly under their control, whether in the data center, utilizing a server-based IaaS instance, a serverless system, or a PaaS cloud service. These things include information and data, application logic and code, identity and access, platform and resource configuration. Furthermore, customers are still in charge of protecting anything in their business that links to the cloud.

4.1.Threat Agent

An entity that can launch an attack and hence poses a threat is known as a threat agent. Threats to cloud security might come from software, people, or other sources, both inside and outside.

1. **Anonymous attacker:** An anonymous attacker is an unauthorized party with no authorization operating in a cloud environment. These attackers are frequently external. Usually, these attackers use public networks to launch network-level attacks. Their efficacy may be hampered by a lack of understanding of security rules and countermeasures, which may force them to turn to anonymous means of identity theft or account bypassing.
2. **Trusted Attacker:** A trusted attacker seeks to take advantage of valid credentials while working in the same cloud environment as the cloud consumer. Attackers like these could go after cloud service providers and other tenants who share IT resources. Trusted attackers, as opposed to anonymous ones, initiate their attacks from inside the cloud's trust boundaries, frequently by gaining unauthorized access to critical data or by misusing legitimate credentials.
3. **Malicious Service Agent:** In a cloud context, a malicious service agent has the ability to intercept and forward network communications. It might be present in the cloud as a malicious or corrupted service agent application. As an alternative, it can appear as an outside application that can remotely intercept messages and perhaps alter their contents.
4. **Malicious Insider:** A human threat agent connected to or functioning on behalf of the cloud provider is known as a malicious insider. These threat agents, who are usually present or former workers or outsiders with access to the cloud provider's facilities, represent serious dangers since they may have administrative privileges that might be exploited to compromise cloud user IT resources.

4.2.Common Attacks

1. Virtualization Attack: A virtualization attack aims to breach the confidentiality, availability, and integrity of IT resources by exploiting vulnerabilities in a virtualization platform. It could occur if users of cloud services who possess administrative access to virtualized resources misuse that authority to launch an attack on the physical infrastructure underneath. Since several users share physical resources in public clouds, there is a very high chance of serious repercussions.

2. Overlapping Trust Boundaries: In a cloud setting, overlapping trust boundaries arise when various cloud service users share physical IT resources. This shared environment could be used by malicious users to compromise resources or other users within the same trust boundary. This could have a broad effect on several users or resources that share the compromised trust boundary.

3. Containerization Attack: Because containers operating on the same host share the host's operating system, containerization, however effective, can provide security issues. Every container on the host could be vulnerable if it becomes infected. Containers can be installed inside virtual servers to reduce this risk by guaranteeing that an attacker can only access a

single virtual server or its containers. As an alternative, implementing a one-service per physical server approach can improve security, but it will require more management and resources.
A few Common attacks in Cloud environments are Traffic Eavesdropping, Malicious Intermediary, Denial of Service, Insufficient Authorization,Phishing.

4.3.Cloud Security mechanisms

There are two types of cloud security mechanisms, CSP supplied and customer-implemented. It is crucial to remember that neither the CSP nor the client are usually alone in charge of addressing security. Usually, it involves teamwork. While cloud-specific versions of many tools may exist, many of the same tools used in on-premises setups should be used in cloud environments as well.

A few security mechanisms are listed below.
1. Encryption: Is a crucial digital coding system that safeguards data confidentiality and integrity by converting plaintext into a secure, unreadable format, preventing unauthorized access during network transmission.

2. Hashing: Is a one-way, irreversible data protection method that securely locks a message without providing a key for unlocking. It's commonly employed for password storage and data integrity verification.

3. Firewall: A firewall is a type of network gateway that controls network traffic by imposing rules on what packets can and cannot pass through, according to predetermined standards. It can be virtual or physical and is essential for safeguarding a company's network against dangers. Within virtualized environments, virtual firewalls protect virtual networks, and some solutions use firewall agents to offer more individualized security.

4. Hardened Virtual Server Image: To generate virtual service instances, a secure template called a hardened virtual server image is utilized. Hardening is the process of deleting unneeded software, blocking unused server ports, and turning off redundant services. In comparison to the original, this produces a virtual server template that is more secure.

5. Virtual Private Network (VPN): A VPN functions similarly to a secure tunnel and enables users to access networks that are firewall-protected from a distance. Data is encrypted to ensure secure network transmission. Secure VPN encrypts and authenticates traffic; on the other hand, Trusted VPN depends on provider confidence to guarantee secure communication.

6. SSO (Single Sign-On) System: SSO makes it easier for users to authenticate for various cloud services. Through a security broker, users authenticate once and retain their security context. This improves usability by removing the requirement for repeated authentication when logging into different cloud services.

7. Public Key Infrastructure (PKI): Asymmetric keys are securely managed via the Public Key Infrastructure system. It authenticates public keys and links them to their owners. An important component of PKI is digital certificates, which are signed by certificate authority and guarantee the security of key pairs used in encryption.

8. Identity and Access Management (IAM): system is a comprehensive mechanism designed to manage user identities and access privileges for IT resources, systems, and environments. It consists of four key components:
Authentication: This component manages user authentication credentials, which involve usernames and passwords. However, it can also support various other methods like digital signatures, digital certificates, biometrics, and IP/MAC address-based access control.
Authorization: This process establishes the connections between user identities, access permissions, and the availability of IT resources as well as the access controls. It guarantees that users have the right amount of access.
User Management: Responsible for administrative tasks such as putting up new user identities, controlling access groups, changing passwords, establishing password regulations, and monitoring user privileges is the user management department.
Credential Management: Establishes identities and access control rules for user accounts, reducing the risk of insufficient authorization.

In contrast to the PKI system, IAM includes policies and access controls and is mostly used to counter threats pertaining to denial of service, overlapping trust boundaries, virtualization, and containerization, as well as inadequate authorization. IAM systems employ preset roles and privileges to check, authenticate, and authorize users.

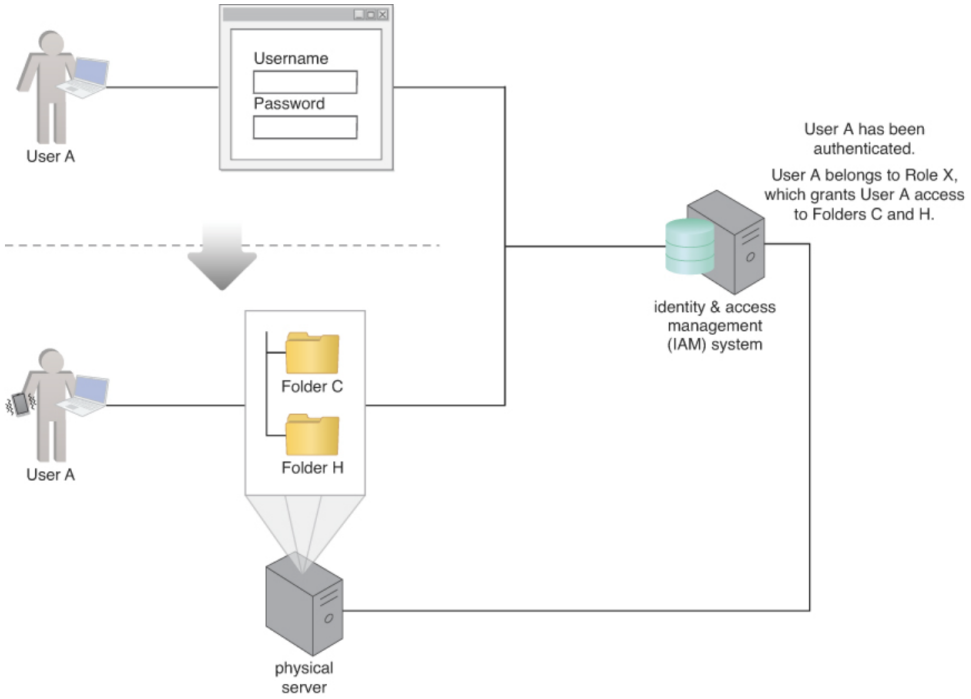

Figure 4.3.1 In the IAM system, User A undergoes authentication, confirming their identity, and is classified as a member of Role X. Leveraging this role-based identification, the IAM system grants User A authorization to access two designated folders on a physical file server.

9. Data Loss Prevention (DLP) System:

A Data Loss Prevention (DLP) system is a crucial tool for security professionals managing and configuring access to distributed information assets. Its primary purpose is to prevent unauthorized or accidental sharing of confidential data, especially as the workforce becomes more remote.

Key capabilities of DLP systems include:

Device Control: Admins can control which devices users are allowed to store or copy data on, like blocking USB drives for storing sensitive data.

Content-Aware Protection: Monitoring and controlling files, emails, and other artifacts to prevent extraction of confidential information.

Data Scanning: Scanning files, emails, and digital documents across devices to identify and label confidential information.

Forced Encryption: Ensuring that any outbound content is encrypted for authorized access only.

DLP systems can also be deployed as cloud-based services to monitor file-sharing applications and sites.

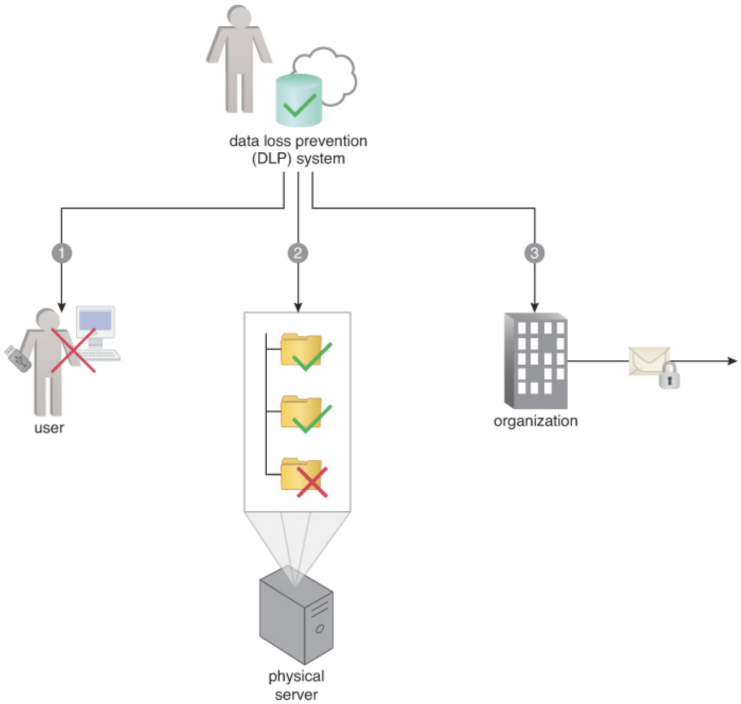

Figure 4.3.2 A security professional with a DLP system blocks a user from storing company data on a USB drive (1), scans a corporate server with files in folders to identify the ones with confidential data (2), and forces an email going outside of the organization boundary to be encrypted (3).

PART 5: CLOUD SERVICE PROVIDERS

5.1.AWS

5.1.1.Introduction

Amazon Web Services (AWS) is a subsidiary of Amazon providing a comprehensive and evolving cloud computing platform. It was officially launched in 2006 and it has been offering IT infrastructure services to businesses as web services and has become a dominant force in the cloud services industry. AWS offers a wide range of cloud-based services, including computing power, storage, databases, machine learning, analytics, and more. These services are designed to help individuals, businesses, and organizations deploy applications and services with greater speed, scalability, and flexibility while minimizing infrastructure costs.

5.1.2.History and Growth of AWS

AWS was initially developed to provide Amazon with a more scalable and reliable infrastructure to support its rapidly growing e-commerce platform. Amazon's experience in handling high volumes of web traffic and computing needs paved the way for AWS's cloud services. In the early 2000s, Amazon's engineers developed the foundational technologies that have later become AWS. This includes the use of virtualization, a key concept in cloud computing, to create flexible and scalable server infrastructure.

Figure 5.1.2.1 AWS History

Launch: AWS officially launched as a commercial service in March 2006, initially offering simple storage and computing services namely: Amazon Simple Queue Service (SQS) and Amazon Simple Storage Service (S3), a highly durable and scalable object storage service, was one of

AWS's groundbreaking offerings, fundamentally changing how businesses stored and accessed data. This marked the beginning of AWS's mission to provide businesses with on-demand, scalable, and cost-effective computing resources.

Over the years, AWS has experienced remarkable growth by continuing to expand its service portfolio. It introduced key offerings such as Amazon Elastic Compute Cloud (EC2) in 2008. EC2 allowed users to run virtual machines in the cloud, providing them with immense computing power on demand. Additional services like Amazon RDS (Relational Database Service), Amazon DynamoDB (NoSQL database), and AWS Lambda (serverless computing) were introduced in the following years. AWS rapidly expanded its global presence by launching data centers in different regions around the world, referred to as Availability Zones (AZs). This global footprint ensured that AWS could offer low-latency access to its services, high availability, and redundancy. AWS's revenue has consistently grown, making it the leading provider of cloud services globally.

5.1.3.AWS Global Outreach

Figure 5.1.3.1 Global Outreach of AWS

Amazon Web Services (AWS) operates a global infrastructure that serves as the backbone of its cloud computing services. The AWS Cloud currently spans 102 Availability Zones within 32 geographic regions, 550+ Points of Presence and 13 Regional Edge Caches around the world. These regions are strategically located to provide cloud resources to customers in North America, South America, Europe, Asia Pacific, and the Middle East, ensuring proximity to users and data residency compliance.

The Availability Zones(AZs) offer redundancy and fault tolerance, meaning that if one AZ experiences an issue, services and data can seamlessly fail over to another within the same region. This architectural design ensures high availability, business continuity, and minimal downtime for applications and services hosted on AWS.

In addition to AZs, AWS maintains a network of Edge Locations, which are strategically distributed to support Amazon CloudFront, AWS's Content Delivery Network (CDN) service. Edge Locations cache and deliver content closer to end-users, reducing latency and enhancing the performance of web applications, video streaming, and other content delivery services.

AWS's global network is highly interconnected, ensuring efficient data transfer across regions and AZs. It utilizes a high-speed, low-latency global backbone network that facilitates the swift and reliable transmission of data. Security and compliance are paramount in AWS's global infrastructure. The company invests significantly in physical data center security, employs stringent access controls, and follows best practices for disaster recovery planning. This robust network architecture supports high availability and low-latency access to AWS services, making it possible for businesses to deploy their applications with confidence, regardless of their geographic location.

5.1.4.AWS Services

AWS offers a vast array of services, broadly categorized into:

- **Compute**: This category includes services like Amazon EC2 for virtual servers, AWS Lambda for serverless computing, and AWS Elastic Beanstalk for application deployment.
- **Storage**: AWS provides scalable storage options like Amazon S3 (Simple Storage Service) for object storage, Amazon EBS (Elastic Block Store) for block storage, and Amazon Glacier for archival storage.
- **Databases**: AWS offers managed database services such as Amazon RDS, Amazon DynamoDB, and Amazon Redshift.
- **Networking**: AWS includes services like Amazon VPC (Virtual Private Cloud), AWS Direct Connect, and Amazon Route 53 for domain name system (DNS) management.
- **Analytics**: AWS provides data analytics tools, such as Amazon EMR, Amazon Athena, and Amazon QuickSight.
- **Machine Learning and AI:** AWS offers machine learning services like Amazon SageMaker, as well as AI services like Amazon Rekognition and Amazon Lex.
- **Developer Tools**: Services like AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline support development and deployment.
- **Security and Identity**: AWS Identity and Access Management (IAM) and AWS Key Management Service (KMS) help manage security and access control.
- **Management and Governance**: AWS services like AWS CloudFormation and AWS Config aid in infrastructure management and governance.

5.1.5 AWS Compute Services

Numerous organizations leverage the AWS compute platform to run a wide array of workloads. AWS holds the distinction of being the leader in Gartner's Magic Quadrant for Cloud Infrastructure and Platform Services for a remarkable 12 consecutive years. Renowned entities like Lyft, Netflix, Coca-Cola, and Moderna have harnessed AWS to reduce infrastructure expenses and expedite innovation on a cloud renowned for its reliability, security, and capabilities.

When it comes to selecting the right compute services for your workloads, AWS provides an extensive and versatile array of options. Amazon Elastic Compute Cloud (EC2) empowers users with meticulous control over infrastructure management, allowing them to choose processors, storage, and networking configurations. AWS's container services offer a ton of options for running containers, ensuring flexibility and choice. AWS Lambda stands ready to execute code in response to events originating from over 150 AWS-integrated sources and various software-as-a-service (SaaS) platforms.

AWS compute services are highly flexible, enabling cost optimization without the need for long-term commitments or intricate licensing agreements. It offers automated recommendations to enhance price performance, along with innovative pricing models and tools for further cost optimization.

5.1.5.1.AWS Lambda

AWS Lambda is a serverless compute service. It enables the execution of code without the need to provision or manage servers. Lambda automates operational tasks such as scaling, patching, and monitoring, allowing developers to concentrate solely on application code.

Key Features:

1. **Serverless Computing:** Lambda adheres to the serverless computing model. This means applications can be built without the responsibility of infrastructure management. Code is uploaded, triggers are defined, and Lambda automatically handles the code's execution in response to events. AWS handles tasks like patching, scaling, and ensuring high availability, freeing developers by eliminating the need for server management
2. **Event-Driven**: AWS Lambda is designed for event-driven applications. It can be configured to respond to various types of events from AWS services, custom applications, or third-party services. For example, Lambda can react to events like file uploads to Amazon S3, changes in DynamoDB tables, or HTTP requests via Amazon API Gateway.
3. **Automatic Scaling**: Lambda functions scale automatically based on the number of incoming events. This implies that Lambda can manage increased workloads without manual intervention when applications experience a surge in events or traffic.
4. **Programming Language Support**: AWS Lambda supports multiple programming languages and runtimes, including Node.js, Python, Java, C#, and custom runtimes. Developers can select the language that best suits the application.

5. **Microservices and Decoupled Architecture**: Lambda is well-suited for microservices architecture and decoupled applications. Developers can create functions that perform specific tasks and have them interact with other Lambda functions or AWS services as needed.
6. **Cost-Efficient Billing**: With AWS Lambda, the billing model charges only for the compute time consumed during code execution. There are no charges when code is idle, making it a cost-efficient choice for many workloads.
7. **Monitoring and Logging**: AWS CloudWatch can be utilized for monitoring and logging Lambda functions, providing insights into their performance and behavior.

Use Cases:

- Efficiently Process Large-Scale Data
- Develop Interactive Web and Mobile Backends
- Harness Advanced Machine Learning Insights
- Build Event-Driven Applications

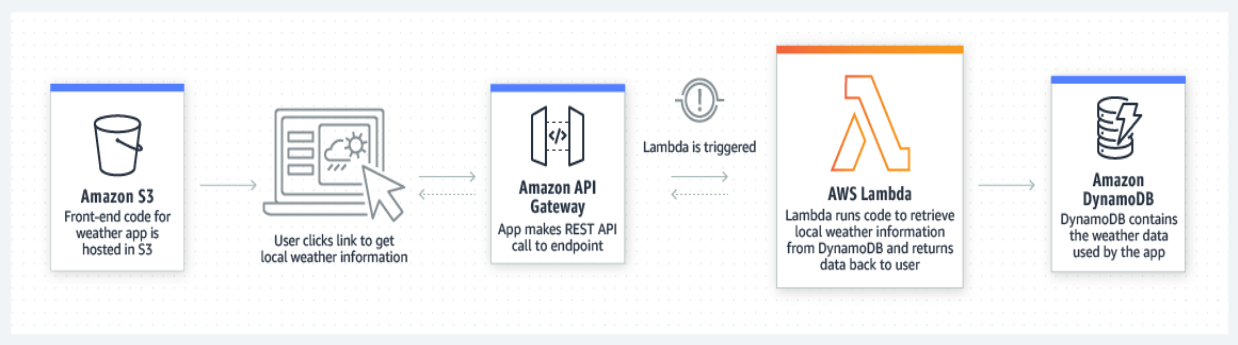

Figure 5.1.5.1.1 Amazon S3, API Gateway, AWS Lambda and DynamoDB work together to retrieve weather data for a web application

5.1.5.2.Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) offers flexible and instantly scalable computing resources within the AWS Cloud. Leveraging Amazon EC2 not only reduces hardware expenses but also expedites the development and deployment of applications. With Amazon EC2, you have the freedom to launch any number of virtual servers required for your workloads. You can effortlessly configure security, network settings, and manage storage. And when the need for resources diminishes, you can readily reduce capacity (scale down) to optimize cost-efficiency.

Table 5.1.5.2.1 - Features of EC2

Feature	Description
Instances	Virtual servers
Amazon Machine Images (AMIs)	Templates for your servers that come preconfigured with everything you need, like the operating system and software.

Instance types	Different configurations for your instances, including CPU, memory, storage, network capacity, and graphics hardware.
Key pairs	Secure login credentials for your instances. AWS holds the public key, and you keep the private key safe.
Instance store volume	Storage for temporary data that gets deleted when you stop, hibernate, or terminate your instance.
Amazon EBS volumes	Permanent storage for your data using Amazon Elastic Block Store (EBS).
Security groups	A virtual firewall that lets you control which protocols, ports, and source IPs can reach your instances and where your instances can connect.
Physical Locations	Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones for the resources
Elastic IP addresses	Static IPv4 addresses for your dynamic cloud computing needs.
Tags	Information you can create and attach to your Amazon EC2 resources.
Virtual private clouds (VPCs)	Virtual networks that you can create, which are isolated from the rest of AWS Cloud. You can optionally connect them to your own network.

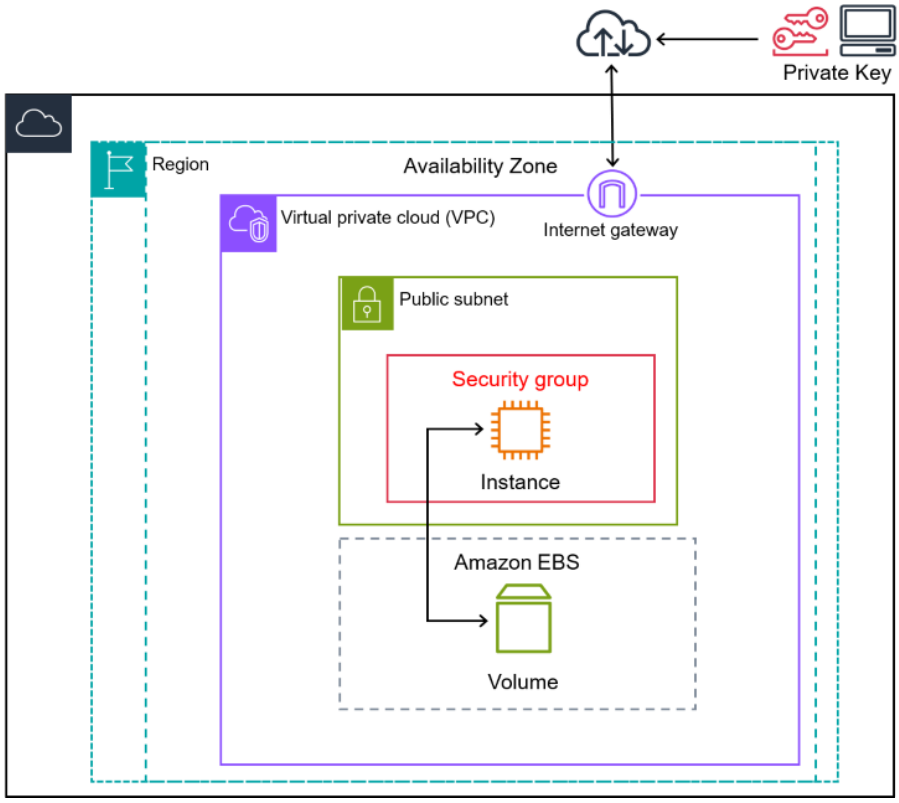

Figure 5.1.5.2.1: Amazon EC2 Instance in Amazon VPC Architecture

The figure illustrates the following:

- The diagram illustrates the basic setup of an Amazon EC2 instance in an Amazon Virtual Private Cloud (VPC).
- The EC2 instance is located within an Availability Zone in a specific AWS Region.
- Security for the EC2 instance is ensured through a security group, acting as a virtual firewall that manages incoming and outgoing traffic.
- To establish the user's identity, a key pair is used, consisting of a private key stored on the user's local computer and a public key stored on the EC2 instance.
- The EC2 instance is associated with an Amazon Elastic Block Store (EBS) volume for data storage.
- The VPC connects to the internet via an internet gateway for external communication.

5.1.6.Amazon Storage Services

AWS offers a variety of storage services to meet different data storage and retrieval needs. These services are used in a wide range of use cases, from simple file storage to high-performance databases. Some of the key storage services are S3 (Simple Storage Service), EBS (Elastic Block Store), EFS (Elastic File System), Amazon File Cache and so on.

5.1.6.1.Amazon S3

Amazon S3 is an object storage service used to store and protect any amount of data. It is highly scalable, faster to access and secure. Customers use it in a wide variety of use cases such as data lakes, websites, mobile applications, enterprise applications, IoT devices, and big data analytics.

Key Features of S3:

- S3 has storage management features such as object lock, replication and batch operations that can be used to manage costs, meet regulatory requirements, and reduce latency.
- Provides features like IAM, ACLs and bucket policies for managing access to buckets and objects. By default, S3 buckets and the objects in them are private.
- Features like S3 Object Lambda, event notifications can be used to transform data and trigger workflows to automate a variety of processing activities at scale.
- S3 provides tools like CloudWatch metrics and CloudTrail for monitoring actions and events in your storage.
- Gain insights into your storage with S3 Storage Lens, Storage Class Analysis and more.
- S3 also supports Versioning to keep multiple variants of an object in the same bucket. It helps preserve, retrieve, and restore every version of all objects stored in a bucket to recover from both unintended user actions and application failures.
- S3 offers strong consistency for data, ensuring that it is reliable and consistent.

5.1.7.AWS Database Services

AWS provides a variety of database services designed for different needs and tasks. These services are built to be easily scalable, always available, and fully managed, making it simpler for developers and businesses to set up, run, and expand their databases. Some of these services include Amazon RDS (Relational Database Service), Amazon Aurora, DynamoDB, Redshift, ElastiCache, and more.

5.1.7.1.Amazon RDS

Amazon RDS is a set of managed services that simplifies the process of creating, operating, and expanding databases in the cloud. It offers various features that make it a popular choice for many situations.

Key Features of Amazon RDS:

- Managed database service that supports multiple relational database engines like MySQL, PostgreSQL, Oracle, and others.
- Automated backups enable you to restore your database to any point within a specified period, aiding in disaster recovery.
- Allows easy scaling of databases to adapt to changes in your application's demands.
- Strong security features, including network isolation, data encryption, and the ability to control access using IAM (Identity and Access Management) and database-level permissions.
- Provides detailed performance and resource usage data through Amazon CloudWatch, facilitating the monitoring of your database's health and performance.

Use Cases:

- Commonly used for storing data in web applications, such as e-commerce websites, content management systems, and blogs.
- Suitable for internal business applications like Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), and Human Resources Management systems (HRMS).
- Ideal for online stores and marketplaces that need dependable and scalable databases to manage product information, orders, and customer details.
- Used by websites and content platforms to organize articles, images, and other content systematically.
- Can serve smaller data warehousing needs, but Amazon Redshift is often preferred for larger-scale data warehousing.
- When combined with tools like Amazon QuickSight or third-party BI tools, RDS can be used for creating reports and data analysis dashboards.

5.1.7.2.Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service designed for high performance, seamless scalability, and quick access to data. It is a popular choice for modern applications that require a flexible, highly available, and fast database.

Key Features of Amazon DynamoDB:

- It's a NoSQL database, not relying on traditional SQL for querying data. This makes it suitable for unstructured or semi-structured data with flexible data models.
- Built for easy and seamless scalability to handle changing workloads.
- Supports various data models, including key-value, document, and wide-column stores, making it adaptable to different application types.
- Known for low-latency and high-performance read and write operations, capable of handling millions of requests per second for applications with heavy workloads.
- Offers two consistency models: strong consistency and eventual consistency, allowing you to choose the one that suits your application's requirements.
- Easily integrates with other AWS services like AWS Lambda, Amazon S3, and Amazon EMR for data analytics.

Use Cases:

- Versatile for various use cases, including user profiles, session management, and product catalogs.
- Ideal for managing and analyzing data from IoT devices.
- Suitable for storing and serving content for websites and apps.
- Effective for collecting and analyzing data for real-time dashboards and reports.
- Used in e-commerce and personalized recommendation systems.
- Useful for serving and tracking online advertisements.

5.1.8.Amazon Network Services

Within AWS, a comprehensive array of network services is available to support the development and management of cloud-based applications and infrastructure. These network services play a pivotal role in assisting users in configuring, safeguarding, and optimizing their network resources within the AWS cloud. Among the essential Amazon network services are Virtual Private Cloud (VPC), Elastic Load Balancing (ELB), Route 53, Cloudfront, and more.

5.1.8.1.Amazon VPC

This networking service empowers users to establish and oversee a virtual network in the AWS cloud. This virtual network is both isolated and customizable, granting the ability to host AWS resources in a controlled and private environment.

Key Features & Components of Amazon VPC

- VPC segments into multiple subnets, each situated in a specific Availability Zone, enabling resource distribution for redundancy and high availability.
- The capability to define security groups and network Access Control Lists (ACLs) to govern inbound and outbound traffic to and from instances.
- The presence of an internet gateway, facilitating internet connectivity for VPC-based instances, thus allowing access to external resources.

- VPC Peering: This feature permits the establishment of connections between VPCs, enabling private communication between them, even if they belong to different AWS accounts.
- VPN Connections: The creation of Virtual Private Network (VPN) connections to extend on-premises data center networks into VPCs.

5.1.8.2.Amazon ELB

Elastic Load Balancing is a service that automates the distribution of incoming network traffic across multiple targets, such as Amazon EC2 instances and containers. This ensures that applications can effectively manage fluctuating levels of traffic and maintain high availability. ELB offers three types of load balancers.

1. **Application Load Balancer (ALB):** Designed for routing HTTP/HTTPS traffic at the application layer (Layer 7) of the OSI model. It can route requests based on content, host, path, and other application-specific information.
2. **Network Load Balancer (NLB):** Optimized for handling TCP and UDP traffic at the transport layer (Layer 4). It is often used for high-throughput, low-latency workloads.
3. **Classic Load Balancer**: Provides basic load balancing capabilities for both HTTP/HTTPS and TCP traffic.

Key features of ELB:

- Seamless integration with AWS Auto Scaling for automatic traffic distribution to healthy instances as applications scale.
- Periodic health checks of instances to ensure that traffic is routed solely to healthy resources.
- High availability across multiple Availability Zones, bolstering redundancy and fault tolerance.
- Configurability to employ security groups and SSL/TLS encryption for securing incoming traffic.

5.1.9.Amazon Security Services

AWS enables organizations to rapidly develop and scale applications with robust security measures. But, the constant incorporation of new tools and services brings a new set of security complexities. As per findings, 70% of enterprises, 61% of small to medium-sized companies expressed concerns about their cloud security. AWS offers a variety of security services and features to help organizations protect their cloud infrastructure, applications, and data. These services are designed to help customers meet compliance requirements. Some of the key AWS security services include Identity and Access Management (IAM), Web Application Firewall (WAF), CloudTrail and so on.

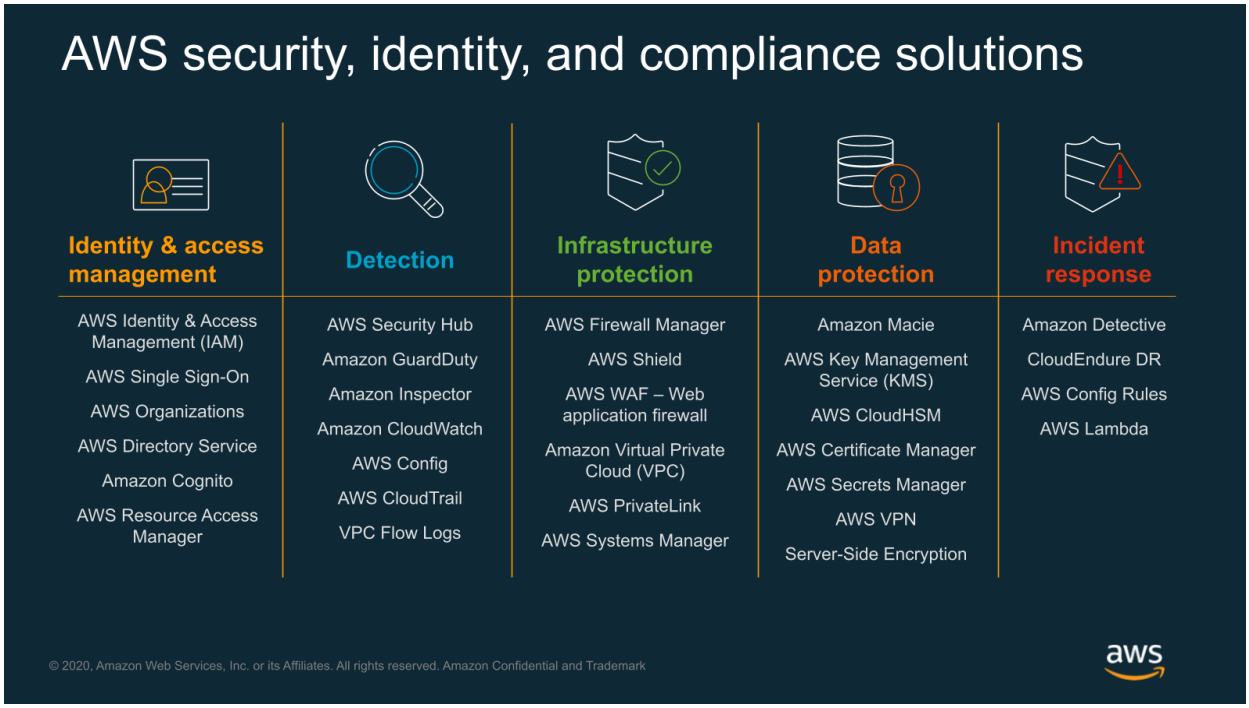

Figure 5.1.9.1 Different AWS Security Solutions and Services

5.1.9.1 AWS IAM:

IAM (Identity and Access Management) is a fundamental service within AWS that enables users to securely control access to AWS resources. IAM offers capabilities to manage users, groups, roles, and permissions. It is essential to ensure that users and services have only the necessary permissions to perform their tasks, thereby reducing the risk of security breaches and unauthorized access to their AWS resources.

Key aspects of IAM:

1. Ability to create and manage user accounts, each with unique credentials for accessing AWS services. Users can be grouped together to simplify permissions management.
2. Roles are permissions to grant data access to users for specific services or resources
3. IAM enables defining granular permissions for users and resources. These permissions are defined through policies.
4. Policies are JSON documents that define the permissions and actions that are allowed or denied within AWS. Policies can be attached to users, groups, or roles
5. IAM supports Multi-Factor Authentication (MFA), which adds an extra layer of security to user accounts by requiring a secondary authentication method in addition to a password.

5.1.9.2.Web Application Firewall (WAF)

AWS Web Application Firewall is a managed security service that is used for enhancing the security and resilience of web applications, preventing unauthorized access, and mitigating

various types of web-based attacks. Combining AWS WAF with other AWS security services, can create a robust security strategy for cloud-based applications.

Figure 5.1.9.2.1 Basic working of AWS WAF

Key aspects of AWS WAF:

1. WAF operates at the application layer of the OSI model and can be used to inspect and filter HTTP and HTTPS requests to mitigate web application-specific attacks.
2. Ability to create custom security rules in order to control access to web applications. Common use cases include protection against SQL injection, XSS, and other application-layer attacks.
3. Web ACLs (Access Control Lists): Web ACLs are a set of rules that can be applied to AWS resources, such as Application Load Balancers, or API Gateway. These ACLs help in defining comprehensive security policies.
4. Provides detailed logs of web requests and security events. Based on these logs users can use Amazon CloudWatch to create custom metrics and set up alerts.
5. Ability to create custom rules and conditions tailored to specific application's security requirements.

5.2.Microsoft Azure

5.2.1.Overview

Microsoft Azure stands as a comprehensive and diverse cloud computing platform, encompassing infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).

It offers an extensive suite of services, including scalable compute resources like virtual machines and serverless computing, varied storage options ensuring secure data handling, and networking tools for connecting on-premises infrastructure with the cloud. With an array of database services and comprehensive AI and machine learning solutions, Azure empowers developers to create intelligent applications. Security features encompass identity management, threat detection, and compliance solutions, meeting various regulatory standards.

5.2.1.1.Azure cloud Strategy

Microsoft Azure relies on Virtualization. All the hardware is emulated in software. EMulation layers are used to map software instructions to hardware instructions. These layers allow virtualized hardware to execute in software like the actual hardware itself.Essentially, Azure Cloud is a set of physical servers in one or more data centers that execute virtualized hardware for customers.

This cloud can create, start, stop, and delete millions of instances of virtualized hardware for millions of customers simultaneously.

Inside each datacenter, the server racks or clusters run virtualized hardware instances for the user. Some of them run cloud management software known as a fabric controller which is responsible for allocating servers, monitoring the health of the server and the services running on it.

Each instance of the fabric controller is connected to another set of servers running cloud orchestration software, typically known as the front end. The front end hosts the web services, RESTful APIs, and internal Azure databases, which are used for all functions in the cloud.

The services situated at the front end are responsible for managing customer requests. These requests involve the allocation of various Azure resources and services, including Azure Virtual Machines and Azure Cosmos DB. Initially, the front end conducts validation and authorization checks to ensure that users are permitted to allocate the requested resources. Upon successful validation, the front end then searches a database to identify an appropriate server rack with ample capacity. Subsequently, it communicates with the fabric controller to allocate the necessary resources.

Azure encompasses an extensive array of servers and networking hardware, facilitating the operation of intricate distributed applications. These applications play a pivotal role in coordinating the configuration and functioning of virtualized hardware and software across the server.

5.2.2.Azure Global Data Center Presence

Microsoft Azure spans over 200 data centers worldwide, organized into 80 regions strategically positioned to meet latency parameters and interconnected by 175,000+ miles of terrestrial and subsea fiber-optic networks. Within each region, it consists of 1 to 3 availability zones, totaling 120 operational and 46 under development, reaching 166 in total.

These zones offer data centers equipped with independent power, cooling, and networking, ensuring protection against facility-level disruptions.

5.2.3.Azure Service Offerings

Azure supports a large collection of services, which includes platform as a service (PaaS), infrastructure as a service (IaaS), and managed database service capabilities. These services cater to various functionalities, including computing, storage, databases, networking, artificial intelligence, machine learning, Internet of Things (IoT), developer tools, security, and more. Below we will be discussing a few services in more detail.

Azure Compute Services enable organizations to deploy, manage and scale applications and workloads in the cloud. Azure provides scalable and flexible compute resources to meet diverse computing needs.

Compute Services include:

- Virtual Machines (VMs): Infrastructure as a Service (IaaS) that allows users to create, manage, and run virtualized applications on Windows or Linux operating systems.
- Azure App Service: Platform as a Service (PaaS) for building, deploying, and scaling web, mobile, and API applications without managing underlying infrastructure.
- Azure Kubernetes Service (AKS): Managed Kubernetes service for deploying, managing, and scaling containerized applications using Kubernetes.
- Azure Functions: A serverless compute service that enables event-driven, on-demand functions that automatically scale as required.
- Azure Batch: Cloud-based job scheduling service for large-scale parallel and high-performance computing applications.

These services offer flexibility, scalability, and various features to cater to different application needs, from traditional virtual machines to fully managed, serverless compute options. Azure Compute Services provide the foundation for deploying and managing applications in the cloud, covering a wide range of use cases and requirements.

Azure virtual machines are one of several types of on-demand, scalable computing resources that Azure offers. Azure virtual machines can be used as computers with specific configurations to develop and test applications, to run the applications so they can be scaled up and down based on usage and as an extended data center.

Azure App Service

Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends.(Figure 5.2.3.1)

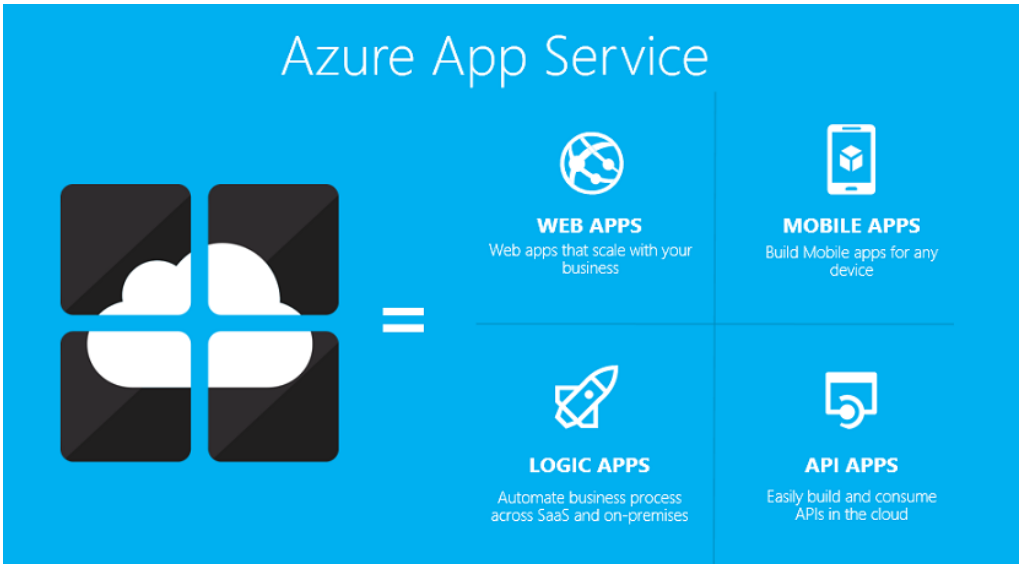

Figure 5.2.3.1 Brief overview of how a Azure App service works

Azure Kubernetes Service (AKS) simplifies deploying a managed Kubernetes cluster in Azure by offloading the operational overhead to Azure. As a hosted Kubernetes service, Azure handles critical tasks, like health monitoring and maintenance. When you create an AKS cluster, a control plane is automatically created and configured.

When you deploy an AKS cluster, you specify the number and size of the nodes, and AKS deploys and configures the Kubernetes control plane and nodes.

A Kubernetes cluster is divided into two components(Figure 5.2.3.2)

- *Control plane*: provides the core Kubernetes services and orchestration of application workloads.
- *Nodes*: run your application workloads.

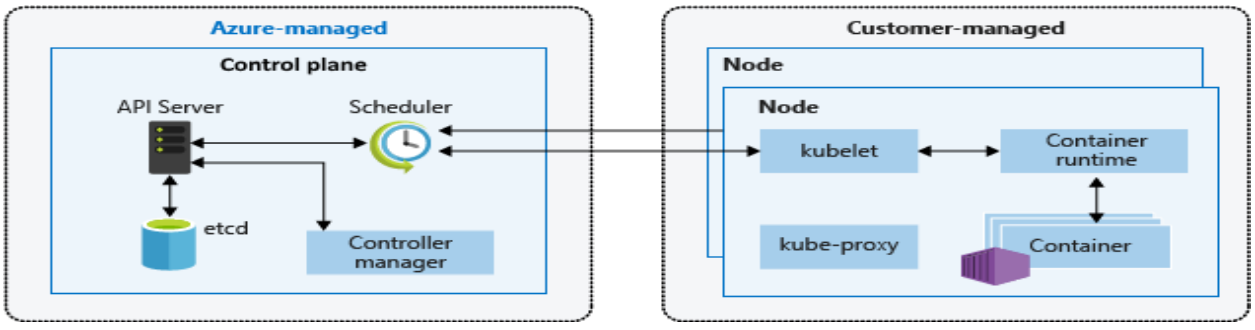

Figure 5.2.3.2 how control plane interacts with nodes

The control plane comprises core Kubernetes components—kube-apiserver, etcd (a distributed key-value store), kube-scheduler, and kube-controller-manager—overseeing various functionalities such as managing Kubernetes APIs, maintaining cluster state, workload assignment to nodes, and supervising controllers for pod and node operations.

An AKS cluster has at least one node, an Azure virtual machine (VM) that runs the Kubernetes node components and container runtime.

The kubelet, a Kubernetes agent, processes orchestration requests from the control plane, managing container scheduling and execution. It also handles node-level virtual networking, facilitating traffic routing and IP management for services and pods. This setup enables containerized applications to interact with resources like the virtual network and storage.(Figure 5.2.3.3)

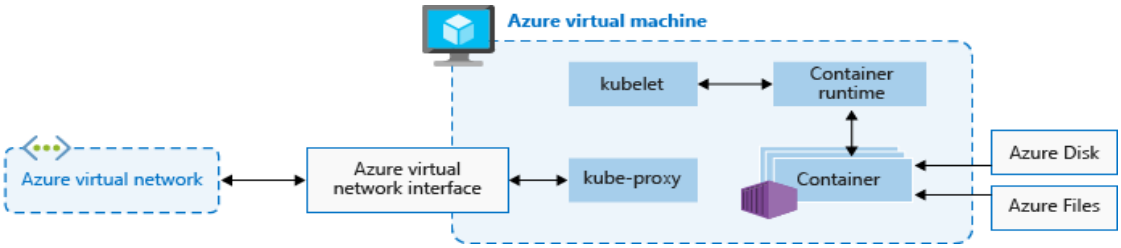

Figure 5.2.3.3 containerized applications interaction with virtual network and storage

Azure Functions is a serverless compute service provided by Microsoft Azure that allows developers to run code in the cloud without the need to manage or provision the infrastructure. It enables the creation of event-driven, on-demand functions that automatically scale based on demand.(Figure 5.2.3.4)

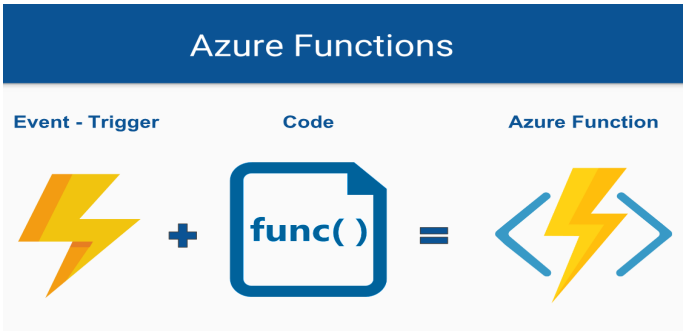

Figure 5.2.3.4 Azure Functions quick glance

Functions provides a comprehensive set of event-driven triggers and bindings that connect your functions to other services without having to write extra code.

Azure function use cases -

- ➔ Azure Functions can be used with Azure messaging service to create advanced event-driven messaging solutions.

Triggers can be set up on Azure Storage queues as a way to chain together a series of function executions. Or use service bus queues and triggers for an online ordering system.(Figure 5.2.3.5)

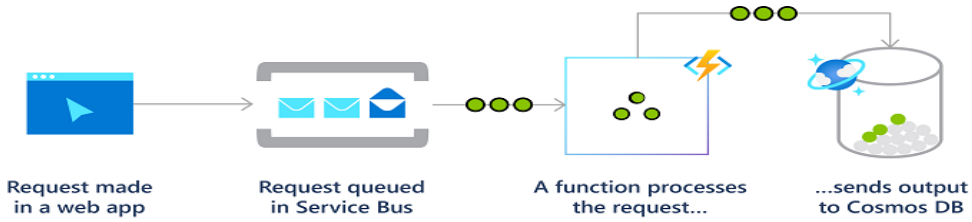

Figure 5.2.3.5 Chaining together a series of function executions

➔ Functions can be used to create triggers to get notified of database changes to initiate log, audit operations. (Figure 5.2.3.6)

Figure 5.2.3.6 Notification workflow

➔ Building a serverless workflow

Functions are often the compute component in a serverless workflow topology, such as a Logic Apps workflow. Long-running orchestrations using the Durable Functions extension can also be created.(Figure 5.2.3.7)

Figure 5.2.3.7 Serverless workflow topology

5.2.4.Storage Services

The Azure Storage platform includes the following data services:

- Azure Blob Storage, Microsoft's cloud-based object storage solution, is optimized for managing extensive volumes of unstructured data, including text and binary files.

  It is well-suited for various purposes such as serving documents directly to web browsers, distributing files for access, streaming audio/video, data backup, disaster recovery, archiving, and enabling analysis by services hosted on-premises or within Azure.
  Accessible globally via HTTP or HTTPS, blobs are accessible through various methods like URLs, Azure Storage REST API, PowerShell, CLI, and client libraries available for multiple programming languages. Additionally, secure connections can be established through SFTP, and containers can be mounted via the NFS 3.0 protocol.

- Azure Files facilitates the creation of network file shares accessible through SMB, NFS, and the Azure Files REST API. This enables multiple VMs to share files with read and write permissions.
  Unlike typical corporate file shares, Azure Files allows global access to files via URLs and shared access signatures (SAS) for specified periods.

- Azure Elastic SAN(Storage Area Network) is Microsoft's solution for optimizing workloads and integrating large-scale databases and performance-intensive applications.

  It simplifies deployment, scaling, and management of SAN while ensuring high availability and cloud-integrated capabilities. It caters to IO-intensive workloads and top-tier databases like SQL, MariaDB, compatible with VMs or containers, including Azure Kubernetes Service.

  Elastic SAN offers simplified deployment, centralized management for multiple compute resources, and cost optimization, making it a robust solution for varied storage needs.

5.2.5.Networking Services

The networking services in Azure provide various networking capabilities including connectivity services, application protection services, application delivery services, network monitoring.

5.2.5.1.Azure Virtual Network (VNet)

Vnet serves as the core infrastructure for constructing private networks within Azure. It enables the deployment of various Azure resources, including virtual machines, Azure App Service

Environments, Azure Kubernetes Service (AKS), and Azure Virtual Machine Scale Sets within the network.

VNets can connect with each other, facilitating communication across resources in different regions through methods such as virtual network peering and Azure Virtual Network Manager. Additionally, resources within a VNet possess outbound communication to the internet by default, with the option to enable inbound communication via public IP addresses or Load Balancers. Moreover, VNets allow connectivity between on-premises networks and computers through features like VPN Gateway or ExpressRoute.

5.2.5.2.Azure Load Balancer

Load balancer functioning at the OSI model's layer 4, acts as the point of contact for clients. It distributes incoming flows to backend pool instances based on configured load-balancing rules and health probes. Backend instances, which can be Azure Virtual Machines or within a Virtual Machine Scale Set, receive these flows.

Public Load Balancers manage outbound connections by translating private IP addresses to public IPs, handling internet traffic to VMs. In contrast, Internal Load Balancers balance traffic within a virtual network and exclusively utilize private IPs at the frontend. This internal load balancer allows access from on-premises networks in hybrid setups (Figure 5.2.5.2.1).

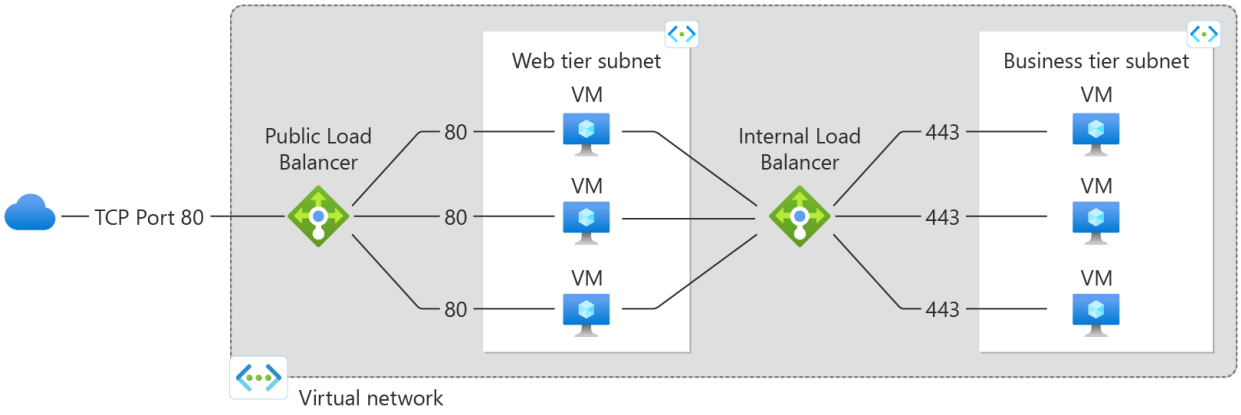

Figure 5.2.5.2.1 Balancing multi-tier applications by using both public and internal Load Balancer

5.2.6.Data Base Services

Azure has different types of databases catering to specific application needs

- Single database has its own set of resources and is managed via a server. Each database is isolated, using a dedicated database engine. It is suitable for modern cloud applications and microservices requiring a dependable single data source and usage patterns are relatively predictable.

- An elastic pool is a grouping of single databases that share resources like CPU or memory. It allows the movement of single databases in and out of the pool as needed. They are suitable for applications which have unpredictable usage patterns.

- Azure Cosmos DB - Modern applications necessitate high responsiveness and constant availability. To achieve this, deploying instances of these applications in nearby data centers is essential for low latency and high availability. The challenge lies in real-time responses during usage spikes, managing vast data volumes, and integrating various databases, adding complexity to scaling applications.

  The surge in AI-powered apps further complicates matters. **Azure Cosmos DB** provides a solution by serving as a unified AI database, accommodating various data models like relational, document, vector, key-value, graph, and table. It's a fully managed NoSQL and relational database designed for diverse app development, ensuring single-digit millisecond response times, instantaneous scalability, SLA-backed availability, and robust security.

5.2.7.Identity and Access Management

Azure's identity management and access control practices are essential for robust security. These practices emphasize treating identity as the primary security perimeter, centralizing identity management, and enabling single sign-on to enhance user productivity. By turning on conditional access, organizations can make informed, automated access control decisions.

Routine security improvements and password management, including multi factor verification, are integral components. Role-based access control allows precise permission assignment, reducing security risks. Lowering exposure of privileged accounts involves monitoring, implementing just-in-time access, and having a robust break-glass process. Enforcing password-less sign-ins or multi factor authentication for critical admin accounts is crucial, along with promptly deprovisioning admin accounts of departing employees. Regularly testing admin accounts using attack simulators is vital to identifying vulnerabilities and bolstering security measures against potential threats.

5.3.Google Cloud Platform

5.3.1.Introduction to GCP

Despite the frequent confusion between Google Cloud and Google Cloud Platform, experts acknowledge that there is a slight distinction between the two.

All of Google's cloud computing services are collectively referred to by the combined name "Google Cloud." This comprises GCP and those related to globally accessible, ready-to-use Google products like Google Workspace and Google Maps Platform.

However, in the industry, the term "Google Cloud Platform" is primarily used to refer to the collection of cloud computing services that are offered to consumers and businesses who wish to construct their own digital cloud infrastructures. It's fascinating to note that GCP is also used by apps like Gmail, YouTube, and the generative AI Google chatbot Bard.

Google's cloud platform GCP, offers a full range of cloud services to its clients. GCP's infrastructure is based on a strong base of both virtual and physical resources, and it powers Google's own services such as YouTube and Gmail. Hard disks and PCs are examples of physical resources, whereas virtual machines (VMs) are an essential part of GCP's virtual resource offerings. These resources are dispersed throughout an international network of data centers, each of which is assigned to a certain area and further subdivided into separate zones.The distinct names of these zones (such as Asia-east1-c) correspond to both their geography and zone designation.
In addition to providing redundancy for data backup in the event of a system failure and lowering communication latency for increased efficiency, resource distribution also fulfills other goals. Strict guidelines and security procedures control access to these dispersed resources, guaranteeing the confidentiality and integrity of data handled and stored inside the GCP environment.

5.3.2.History and evolution of Google Cloud Platform:

In 2008, Google released a trial version of App Engine, a technology that allows users to run their web applications on Google's servers. It was initially made available to a select group of developers. It wasn't released in its entirety until 2011.

Since then, Google Cloud Platform has created or purchased a wide range of services and goods that let businesses utilize every aspect of its enormous infrastructure. Along with other cloud providers like Amazon, Microsoft, and IBM, Google Cloud Platform swiftly rose to the top of the international cloud vendor rankings.

5.3.3.Google cloud’s Data centers

Alphabet Inc.'s cloud computing service, Google Cloud Platform (GCP), offers networking, storage, and processing capabilities via its data centers spread over 25 nations and almost 40 locations globally. The company's need for additional data center capacity is being driven by the expansion of Google Cloud regions as well as the company's main products and services, including YouTube, Gmail, Google Drive, Google Maps, Google Photos, Google Play, and Search.

With 10 more regions under development and 39 now operational, Google Cloud will have 49 total regions available by the end of 2024. Three to four deployment zones correspond to groups of data centers with different physical infrastructures (power, cooling, and networking) inside each Google Cloud region.known by Google as zones, and commonly referred to by other cloud service providers as availability zones.

Figure 5.3.3.1 Google Cloud Datacenters

5.3.4.Google Cloud Service Offerings

Compute Services: Compute Engine, which offers highly configurable virtual machines (VMs) for executing applications, is one of the compute services offered by Google Cloud. Additionally, you may use Cloud Functions for serverless, event-driven computing, App Engine for Platform-as-a-Service (PaaS) for web apps, and Google Kubernetes Engine (GKE) for managed Kubernetes orchestration.

Storage Services: Cloud Storage, an object storage service for data archiving and retrieval, is one of the storage options provided by Google Cloud. While Cloud Bigtable offers a NoSQL database for large-scale applications, Cloud SQL is a managed relational database service. Cloud Firestore is a serverless NoSQL document database, while Cloud Spanner provides a globally distributed, horizontally scalable database. Managed file storage is offered by Cloud Filestore.

Networking Services: Google Cloud provides Virtual Private Cloud (VPC) for setting up and maintaining private networks within the networking space. Global traffic distribution is

guaranteed by cloud load balancing, and content delivery is improved globally by cloud CDN. For domain management, Cloud DNS is a managed Domain Name System service.

Security and identity services: Strong security and identity services are offered by Google Cloud, including Identity and Access Management (IAM) for identity management and access control. The management of device and user identities is the purpose of Cloud Identity. Cryptographic key management is made possible by Google Cloud Key Management Service (KMS), and security and risk management are provided by Cloud Security Command Center.

Services for Data Analytics and Machine Learning: Google Cloud provides BigQuery, a fully managed serverless data warehouse, for data analytics and machine learning. While Dataflow manages batch and stream data processing, Dataprep offers data preparation and cleaning services. Vision AI and Natural Language API provide machine learning-based text and picture analysis, while AI Platform is a managed machine learning platform. TensorFlow is an open-source machine learning framework, and autoML offers automated machine learning.

Storage Services: Transfer Appliance and Transfer Service are two services offered by Google Cloud that provide hardware-based and online data transfers to the cloud. Data transfers from on-premises systems or other cloud providers to Cloud Storage are automated with Storage Transfer Service.

Development and DevOps Services: Cloud Build for continuous integration and continuous delivery (CI/CD), Cloud Source Repositories for version control, Cloud Functions for serverless computing, Cloud Run for managed containers, and Cloud Endpoints for API management are just a few of the development and DevOps tools that Google Cloud provides. Performance analysis is provided by Cloud Profiler, while debugging and monitoring are aided by Cloud Debugger and Cloud Trace.

5.3.4.1.GCP Compute Services

Compute Services: Compute Services are required to carry out specific tasks in order to establish and manage a Virtual Machine on the Google Cloud Platform. Depending on the demands of the user, Google Cloud Platform's Compute Engine offers a range of computing alternatives. Some of them are discussed below,

1. Cloud Dataproc: Using Apache Hadoop, Cloud Dataproc is a managed service for handling big data workloads. It is perfect for jobs like data processing and analysis since it enables users to build virtual machine clusters for analytics.

2. Cloud Functions: Without having to worry about maintaining infrastructure, developers may run code in response to events using Cloud Functions, a serverless computing service. It can be used to create event-driven applications and carry out operations like image processing, database searches, and notification sending.

3. Cloud Storage API: With the help of this RESTful API, developers may store and retrieve data from Google Cloud Storage programmatically. It's a scalable and highly available storage solution that lets you query metadata about items that are stored.

4. Cloud Pub/Sub: This publish/subscribe messaging system facilitates cloud-based real-time messaging applications. For real-time communication, it enables developers to establish topics, publish messages, and subscribe to topics.

5. Cloud Bigtable: A multi-model database that is distributed globally is provided by Google Cloud Bigtable. It enables developers to efficiently manage data, write and read data, and implement administrative tasks.

6. Cloud Spanner: Through RESTful calls, Cloud Spanner offers a globally distributed relational database service that is horizontally scalable. It is intended for use in applications that require horizontal scaling and high availability without managing infrastructure.

7. Cloud Dataflow: This fully managed solution handles batch and streaming data processing. Because of its high availability and scalability, it is appropriate for processing massive amounts of data.

8. Cloud Tasks: This managed task execution solution allows us to run quick code in response to user inputs. It can be used by developers to schedule jobs for particular days or times, or to perform them on demand.

9. Cloud Storage: Programmatic access to Google's object storage service is possible using the Google Cloud Storage APIs. Google Cloud Storage buckets allow developers to store, retrieve, and query objects, making it a highly available and scalable storage solution.

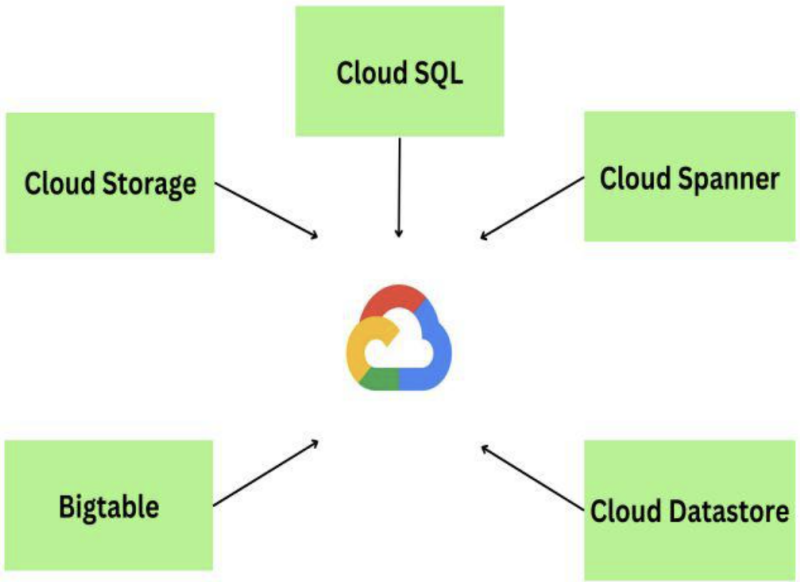

Figure:5.3.4.1.1 Google Clouds compute services

5.3.4.2.Google Cloud Storage Services

1. Google Cloud Storage - Object Storage

Google Cloud Storage (GCS) is a powerful object storage system made to act as a single location for a variety of data, including pictures, videos, documents, and other types of files. Data in GCS is arranged into objects, which are the basic building blocks of storage. These items are arranged into groups known as "buckets," which are essential for organizing and limiting data access. Because GCS offers flexible storage options, it may be used for a wide range of purposes, such as disaster recovery, data archiving, website content delivery, and the smooth transfer of big data files to end users.

2.Google File System

The Google File System (GFS) was carefully designed to meet Google's unique needs for storing large datasets on a variety of low-cost commodity hardware. It can be succinctly characterized by its unique architecture, robust functionality, and its optimal suitability for handling large-scale workloads of various kinds.

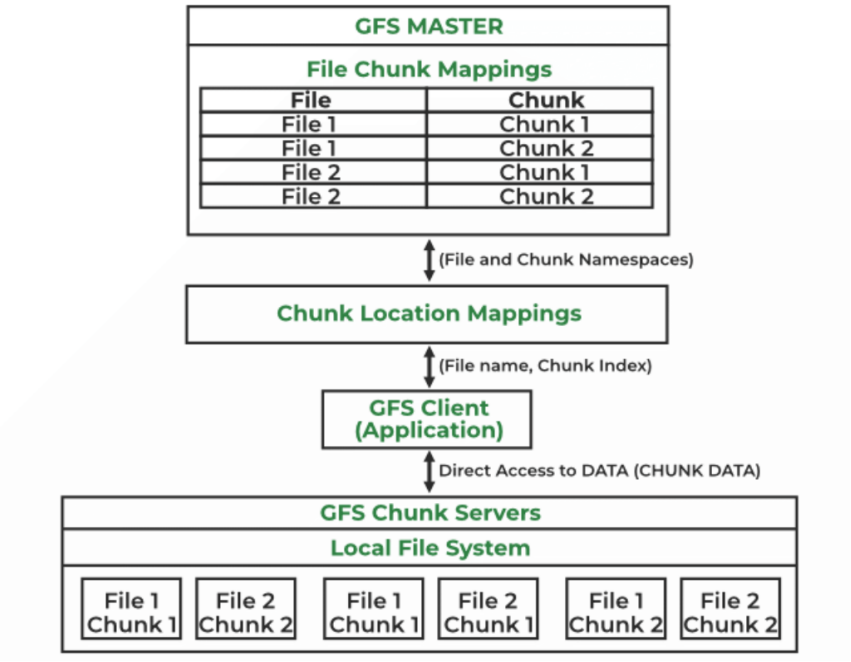

Figure 5.3.4.2.1. Architectural diagram of the Google File System

GFS Components:

GFS Clients: These are the interfaces that programs use to communicate with the file system. They start reads, writes, and appends among other file system operations.

GFS Master Server: The file system's control information is stored on this server. It keeps track of all file metadata, including namespace details, access control details, and chunk locations as of right now.

GFS Chunk Servers: The data chunks, which are 64 MB in size by default, are stored on these servers. Because they manage data serving, chunk servers are crucial for data archiving and retrieval.

3.Google Block storage - Persistent Disk

Google Persistent Disk is a powerful block storage solution offered by Google Cloud Platform (GCP) that is intended to handle storage volumes connected to virtual machine instances effectively. It provides a number of features with high performance, dependability, and adaptability as top priorities. To combine performance and affordability, you can select between HDD and SSD storage solutions, making sure your storage satisfies your unique needs. To guarantee availability and data integrity, data is redundantly stored within the same zone. In the event of hardware failures, data is rapidly copied for durability. Google Persistent Disk is a flexible and adaptive storage option inside GCP's infrastructure since it allows you to dynamically resize disks and scale storage capacity up or down as needed.

An essential part of GCP's infrastructure, persistent disks offer a safe and flexible storage option that can be tailored to meet a range of performance and budgetary requirements. Large databases, enterprise apps, and containerized workloads may all be efficiently supported by Google Persistent Disk, which provides the scalability and dependability required to meet your storage needs.

5.3.4.3.Google Cloud Networking Services

1.**Google Cloud's Virtual Private Cloud (VPC)**
Google Cloud Platform's Virtual Private Cloud (VPC) offers a way to control access and isolate resources inside the cloud. You can specify who is authorized to use the resources deployed in your VPC by specifying IP addresses. Resource isolation requires VPCs in order to safely divide various workloads. Additionally, they provide fine-grained control over network traffic, both entering and leaving the system—a crucial component of a strong security posture. Furthermore, because VPCs are naturally scalable, applications can grow and expand without sacrificing performance or security.

2.Load Balancing in Google Cloud
In Google Cloud, load balancing is used to divide network traffic among several servers in order to prevent any one server from becoming a bottleneck. When there is an increase in network

traffic, like when multiple customers make requests at once, this service is essential. Different forms of load balancing are available from Google Cloud to meet various needs:

Network load balancers(NLBs) : NLBs, are appropriate in situations when hashing individual IP addresses is not necessary. Rather, NLBs distribute requests equally by directing traffic according to the destination server's IP address. They have a limit on the quantity of requests they can handle in a second and mainly support the HTTP and HTTPS protocols.

Distributed Network Load Balancers (DNLB): These sophisticated load balancers support a variety of protocols, including TCP, HTTP, HTTPS, SMTP, and POP3, and use hashing algorithms like SHA-3 to distribute traffic more effectively among servers. DNLBs are useful for many different applications and function especially well with intricate network architectures.

You can run your whole network infrastructure under a single instance that manages load balancing and DNS/DHCP services when you implement load balancing services. This gives all of these services a single IP address, which makes network management easier.

5.3.4.4.GCP Identity and Access Management

In Google Cloud, Cloud Identity and Access Management (IAM) is essential for defining and controlling user behavior across different resources. IAM ensures the proper people have access to the resources they need to do their jobs by centralizing the management of user rights. In contrast to direct permission assignments, IAM uses a role-based paradigm in which users, groups, or service accounts are assigned roles with different permissions. By following the concept of least privilege, this architecture improves security and makes it easier to manage access privileges and link organizational responsibilities to job functions.

The management of Google Cloud resources is made possible by the smooth integration of IAM with GSuite, which uses well-known Google interfaces. Incorporating an audit trail facilitates compliance standards adherence and offers a consolidated perspective of security rules, complementing this harmonization. Within Identity and Access Management (IAM), roles are primarily divided into two categories: custom roles, which are created specifically for a user's needs and are maintained by the user, and predefined roles, which are managed by Google Cloud and updated over time. Custom roles give enterprises additional flexibility and control over how their cloud environments are accessed, while predefined roles adapt easily to new capabilities.

5.3.4.5.GCP DevOps and CI/CD Tools

1. Cloud Build
A serverless build solution is provided by the Google Cloud service called Cloud Build, which makes it easier to import source code from storage or repositories, carry out build processes,

and produce artifacts like Java archives or Docker containers. This service provides necessary resources to build your software supply chain's security, in line with Supply Chain Levels for Software Artifacts (SLSA) at level 3. YAML or JSON files are used to define build settings, which include tasks like dependency retrieval, testing, and code compilation. Cloud Build executes each build action step in a series of Docker containers. Users can use Cloud Build's basic build stages, pre-defined community contributions, or custom steps made to meet specific needs.

Cloud Build offers many methods for initiating and managing builds. The Cloud Build API or the Google Cloud CLI can be used to manually start builds. Build triggers for automated continuous integration and continuous deployment (CI/CD) workflows are set up to initiate fresh builds automatically when there are changes to the code. Popular code repositories like GitHub, Bitbucket, and Cloud Source Repositories are easily integrated with these triggers. Through the Cloud Build API, the gcloud CLI, or the Build History page in the Google Cloud dashboard, users can view build results, which contain thorough logs and information of each build process.

Usually within its default pools, Cloud Build runs in safe and isolated build environments. There is some flexibility available with these default pools regarding machine and disk capacity. On the other hand, Cloud Build offers private pools with additional control, including the ability to create private network access, for those seeking more sophisticated customization choices. Improved control over the build environment is provided by these dedicated worker pools, while Cloud Build manages the default and private pools, guaranteeing scalability and lowering the need for infrastructure maintenance.

2.Cloud Scheduler

Cloud Scheduler is Google Cloud's fully managed cron job service, enabling the scheduling of work units, commonly known as cron jobs, at regular intervals or specific times. Here's a brief overview under three aspects:

Cloud Scheduler allows for the automation of various time-based tasks. These tasks range from daily emails, refreshing cache every few minutes, to hourly summary updates. By leveraging this service, developers can schedule any task that can be triggered through HTTP/S endpoints, making it a versatile tool for maintaining consistent and timely operations within applications.

Cron jobs within Cloud Scheduler can be set up through the Cloud Console or the gcloud CLI, providing flexibility in management. The service's design for "at least once" delivery offers robust reliability, minimizing the chances of missed executions. It's crucial for the targets of these cron jobs to be idempotent, meaning they produce the same outcome regardless of the number of times they are executed to prevent unintended effects of any duplicate runs.
A few other DevOp Tools include Cloud SQL, Cloud Deploy, Cloud SDK, Cloud source repositories, Cloud code, etc

PART 6: Future Developments and Conclusion

6.1.Edge Computing

6.1.1.Background

Edge computing finds its roots in the late 1990s when content delivery networks (CDNs), introduced by Akamai, optimized web performance. These CDNs utilized edge nodes to prefetch and cache web content, especially beneficial for bandwidth savings in video content delivery.

Building on the CDN concept, edge computing extends the idea by integrating cloud computing infrastructure. Unlike CDNs limited to caching web content, edge computing enables cloudlets to execute arbitrary code, encapsulated in virtual machines or lightweight containers for isolation and resource management.

Early insights into edge computing's potential arose in the late '90s, highlighting how offloading computation to nearby servers could improve performance and conserve mobile device battery life. As cloud computing emerged, services like Apple's Siri and Google's speech recognition offloaded computation to the cloud, albeit with high average separation and latency.
These observations led to the conceptual foundation for edge computing in a 2009 article, advocating a two-level architecture involving unmodified cloud infrastructure and dispersed cloudlets. Cloudlets employ persistent caching to simplify management despite their dispersed nature.

	Applicable situation	Network Bandwidth Pressure	Real-Time	Calculation Mode
Cloud Computing	Global	More	High	Large Scale Centralized processing
Edge Computing	Local	Less	Low	Small scale intelligent analysis

Table 6.1.1.1 - Main differences between Edge Computing and Cloud Computing

6.1.2.Edge Computing and Cloud

Before the emergence of edge computing, traditional cloud computing transfers all data to the cloud computing center through the network, and solves the computing and storage problems in a centralized way. With the development of search engines represented by Google, cloud computing starts to show strong vitality. Nowadays, cloud computing has gradually developed. It

is a very powerful network service platform including distributed computing, load balancing, parallel computing, network storage, virtualization and other technologies. However, nowadays, with the popularization and development of the Internet of Things in people's life, the number of devices connected to the Internet of Things is gradually increasing, and a large amount of data is generated. The network bandwidth of cloud computing has been unable to meet the needs of time-sensitive systems and real-time performance. Therefore, cloud computing models have great defects in load, real-time, transmission bandwidth, energy consumption and data security and privacy protection .

Edge computing is a distributed computing paradigm that involves bringing computation and data storage closer to the location where it is needed, typically at the "edge" of the network. It aims to reduce latency and improve efficiency by processing data nearer to the source and reducing the need to transmit information to centralized cloud data centers for processing.

6.1.3 Key aspects of edge computing

Edge computing is fundamentally positioned at the periphery of the network, in close proximity to where data originates, such as IoT devices, sensors, and local networks. It specializes in real-time processing, enabling swift decision-making and significantly reducing latency. Its decentralized nature diversifies the processing landscape, moving away from the centralized model of traditional cloud computing.

With its emphasis on immediate and localized data processing tasks, edge computing bolsters reliability and redundancy. By dispersing computing capabilities, it mitigates reliance on a single centralized data center. Moreover, edge computing optimizes bandwidth usage by minimizing the necessity to transfer extensive volumes of data to remote cloud servers.

Conversely, cloud computing operates from distant data centers, centralizing storage and processing. It excels in managing resource-intensive, data-heavy, and complex computational tasks, offering extensive storage capacities and robust analytics capabilities. Cloud computing and edge computing complement each other by functioning as part of a hybrid ecosystem.

Cloud computing serves as a repository for data storage, complex analytics, and non-immediate processing. Meanwhile, edge computing handles the real-time or near-real-time processing of data at or near its source. Together, they create a hybrid model that amalgamates the strengths of centralized cloud services with the agility and speed of distributed edge computing. This hybrid approach offers a comprehensive, efficient, and flexible solution to cater to a broad spectrum of computing needs in our data-driven and interconnected world.

6.1.4 General Architecture of Edge Computing

Edge Computing typically extends cloud services to the edge of the network by placing edge devices between terminal devices and cloud computing. The structure of cloud-edge collaboration consists of a terminal layer, edge layer and cloud computing layer.

The following is a brief introduction to the composition and functions of each layer in the edge computing architecture.(Figure 6.1.4.1).

The ***terminal layer*** encompasses a variety of devices like mobile terminals and IoT devices such as sensors, smartphones, smart cars, and cameras. Here, devices act as both data consumers and providers, focusing primarily on data perception rather than heavy computing. This layer involves hundreds of millions of devices collecting raw data, which is then transmitted to the upper layer for storage and processing.

The ***edge layer,*** positioned at the network periphery, comprises distributed edge nodes between terminal devices and clouds. It includes base stations, access points, routers, switches, and gateways. This layer facilitates terminal device access, stores, and processes uploaded data, connecting with the cloud to upload processed data. Due to its proximity to users, the edge layer excels in real-time data analysis and intelligent processing, offering more efficient and secure operations compared to cloud computing.

In the cloud-edge computing federated services, cloud computing serves as the most robust data processing center. It comprises high-performance servers and storage devices capable of extensive data analysis, particularly for tasks like regular maintenance and business decision support. The ***cloud layer*** permanently stores edge computing data, handles analysis beyond the edge layer's capacity, and processes global information integration. Additionally, it dynamically adjusts edge computing deployment and algorithms based on control policies.

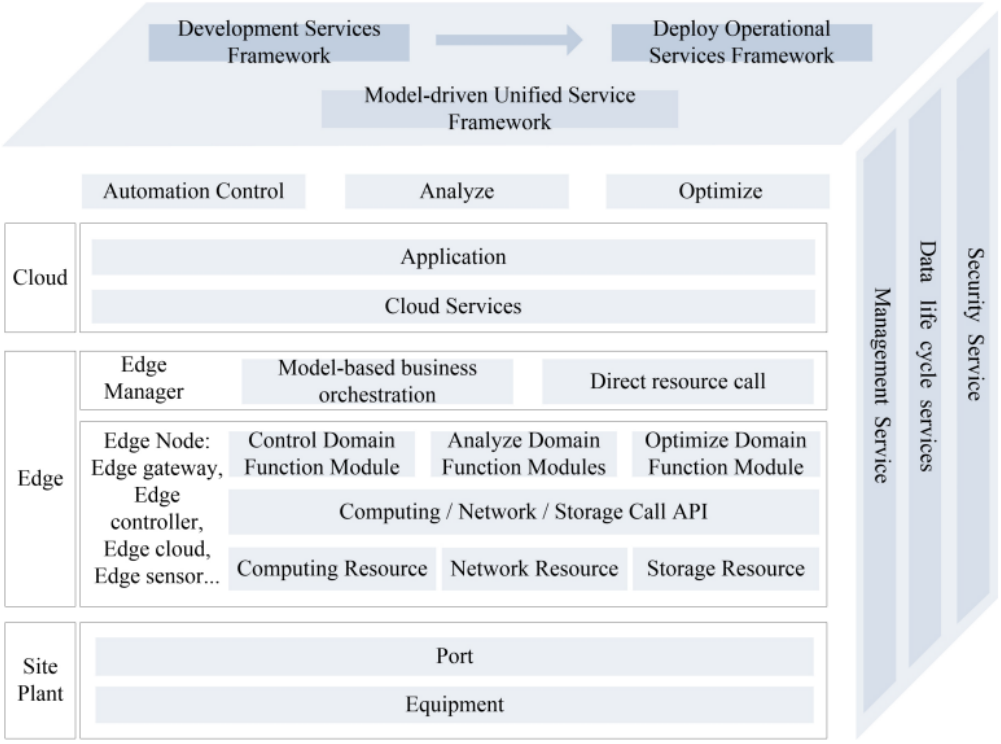

Figure 6.1.4.1 Edge Computing reference architecture 3.0

6.2.Artificial Intelligence

Extending Utilizing cloud computing to access AI

The popularization of AI technology is becoming more and more dependent on cloud infrastructure. Big data and computational resources are needed for training large-scale AI models, such as the ones that power ChatGPT, and these resources are outside the reach of many businesses. These businesses may now access AI as-a-service through cloud platforms, creating new avenues for innovation and digital transformation. Cloud services give even small firms access to state-of-the-art AI capabilities, allowing them to innovate faster and compete on a bigger scale. The capacity of cloud computing to house these AI models offers a strong basis for their broad use, guaranteeing a number of societal and economic benefits from their adoption.

Investing in AI to Improve Cloud Functionality

Cloud service companies are increasing the amount of money they invest in AI and ML, which is expanding the potential of cloud computing. This investment infusion aims to provide advanced features that open the door for more intelligent and durable cloud services, in addition to augmenting current capabilities. Future improvements are probably going to include

self-correcting infrastructures that are able to do preventive maintenance and repair as well as sophisticated, scalable systems that can react dynamically to workload needs. By taking such proactive measures, cloud technology will continue to lead digital infrastructure services and meet complicated business needs more accurately and efficiently.

The Interesting Development of Cloud-AI Coordination

The development of cloud computing is clearly following the same trend as AI. The motivation to combine AI tools with cloud computing and storage services is growing as these algorithms get better at interpreting complicated data patterns. In addition to deepening data analysis, cloud computing and AI cooperation enable these intelligent systems to gradually improve their capabilities. As a result, it is projected that demand for cloud-integrated AI services will increase gradually due to its ability to provide data-driven insights and enhance decision-making processes for both individuals and enterprises. The cloud is poised to revolutionize how we manage our digital information with every breakthrough in AI.

6.3 Conclusion

In conclusion, it is clear that cloud computing has evolved from a mere technology to a driving force for innovation and digital transformation. The cloud has become a powerful tool for making things better, faster, and more innovative in the digital world. It can provide resources instantly and adjust to your needs, revolutionizing various industries. However, using the cloud comes with a responsibility to keep your data safe, use resources wisely, and choose the right cloud services.

This document serves as a foundational resource for understanding cloud computing concepts and introduces renowned cloud computing providers such as AWS, Azure, and GCP. It offers valuable insights into this dynamic and rapidly growing domain, paving the way for future developments in cloud computing. We hope it helps you make smart decisions and explore the world of cloud computing effectively.

References:

1. "Cloud Computing: Concepts, Technology & Architecture" by *Thomas Erl, Ricardo Puttini, and Zaigham Mahmood*.
2. https://cloudsecurityalliance.org/blog/2020/08/26/shared-responsibility-model-explained/
3. https://www.techtarget.com/searchsecurity/definition/cloud-security
4. https://www.mega.com/blog/what-is-scalability-in-cloud-computing
5. https://www.techrepublic.com/article/multi-cloud-architecture/
6. https://www.techrepublic.com/article/google-cloud-platform-the-smart-persons-guide/
7. https://www.geeksforgeeks.org/introduction-to-google-cloud-platform/
8. https://dgtlinfra.com/google-cloud-data-center-locations/

9. https://www.geeksforgeeks.org/key-gcp-compute-services/?ref=lbp
10. https://cloud.google.com/products/tools/?hl=en
11. https://www.androidpolice.com/google-cloud-platform-guide/
12. https://aws.amazon.com/what-is-cloud-computing/
13. https://www.geeksforgeeks.org/cloud-deployment-models/#
14. https://itchronicles.com/cloud/the-evolution-of-cloud-computing-wheres-it-going-next/
15. https://www.aquasec.com/cloud-native-academy/cspm/top-7-risks-of-cloud-computing/
16. https://www.synopsys.com/cloud/insights/essential-cloud-computing-characteristics.html
17. https://www.finra.org/rules-guidance/key-topics/fintech/report/cloud-computing/service-models
18. https://www.inap.com/blog/iaas-paas-saas-differences/
19. https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-private-public-hybrid-clouds/#overview
20. https://scaleyourapp.com/a-super-helpful-guide-to-understanding-workload-its-types-in-cloud/
21. https://aws.amazon.com/what-is/cloud-native/
22. https://aws.amazon.com/devops/what-is-devops/?trk=faq_card
23. https://docs.aws.amazon.com/
24. https://www.knowledgehut.com/blog/cloud-computing/cloud-computing-future#the-future-of-cloud-computing-2025-2030
25. https://www.researchberg.com/index.php/rrst/article/view/18
26. M. Satyanarayanan, "The Emergence of Edge Computing," in Computer, vol. 50, no. 1, pp. 30-39, Jan. 2017, doi: 10.1109/MC.2017.9. https://ieeexplore.ieee.org/document/7807196
27. https://azure.microsoft.com/en-us/explore/global-infrastructure
28. https://en.wikipedia.org/wiki/Elasticity_(system_resource)
29. Z. Wang and A. Basu, "Resource allocation for elastic traffic: architecture and mechanisms," NOMS 2000. 2000 IEEE/IFIP Network Operations and Management Symposium 'The Networked Planet: Management Beyond 2000' (Cat. No.00CB37074), Honolulu, HI, USA, 2000, pp. 157-170, doi: 10.1109/NOMS.2000.830382.
30. K. Cao, Y. Liu, G. Meng and Q. Sun, "An Overview on Edge Computing Research," in IEEE Access, vol. 8, pp. 85714-85728, 2020, doi: 10.1109/ACCESS.2020.2991734.