MAILAM Engineering College

(

Approved by AICTE, New Delhi, Permanently Affiliated to Anna University Chennai,
Accredited by NBA, NAAC with A Grade and TCS)

DEPARTMENT OF

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CCS335 - CLOUD COMPUTING

[REGULATION-2021] STUDY MATERIAL

NAME OF THE STUDENT	**:_____**
REGISTER NUMBER	**:_____**
YEAR/SEM	**: III/V**
ACADEMIC YEAR	**:2024-2025**

PREPARED BY:

Dr.S.ARTHEESWARI, HOD/AI&DS

MAILAM Engineering College

Approved by AICTE, New Delhi, affiliated to Anna University, Chennai, Accredited by NBA & TCS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CCS335 – CLOUD COMPUTING

III YEAR / V SEM

COURSE OBJECTIVES:

- To understand the principles of cloud architecture, models and infrastructure.
- To understand the concepts of virtualization and virtual machines.
- To gain knowledge about virtualization Infrastructure.
- To explore and experiment with various Cloud deployment environments.
- To learn about the security issues in the cloud environment.

UNIT I CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE 6

Cloud Architecture: System Models for Distributed and Cloud Computing – NIST Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models; Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

UNIT II VIRTUALIZATION BASICS 6

Virtual Machine Basics – Taxonomy of Virtual Machines – Hypervisor – Key Concepts – Virtualization structure – Implementation levels of virtualization – Virtualization Types: Full Virtualization – Para Virtualization – Hardware Virtualization – Virtualization of CPU, Memory and I/O devices.

UNIT III VIRTUALIZATION INFRASTRUCTURE AND DOCKER 7

Desktop Virtualization – Network Virtualization – Storage Virtualization – System-level of Operating Virtualization – Application Virtualization – Virtual clusters and Resource Management –Containers vs. Virtual Machines – Introduction to Docker – Docker Components – Docker Container – Docker Images and Repositories.

UNIT IV CLOUD DEPLOYMENT ENVIRONMENT 6

Google App Engine – Amazon AWS – Microsoft Azure; Cloud Software Environments –

Eucalyptus – OpenStack.

UNIT V CLOUD SECURITY 5

Virtualization System-Specific Attacks: Guest hopping – VM migration attack – hyperjacking. Data Security and Storage; Identity and Access Management (IAM) - IAM Challenges - IAM Architecture and Practice.

30 PERIODS

PRACTICAL EXERCISES: **30 PERIODS**

1. Install Virtualbox/VMware/ Equivalent open source cloud Workstation with different flavours of Linux or Windows OS on top of windows 8 and above.
2. Install a C compiler in the virtual machine created using a virtual box and execute Simple Programs
3. Install Google App Engine. Create a hello world app and other simple web applications using python/java.
4. Use the GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Install Hadoop single node cluster and run simple applications like wordcount.
8. Creating and Executing Your First Container Using Docker.
9. Run a Container from Docker Hub

COURSE OUTCOMES:

CO1: Understand the design challenges in the cloud.

CO2: Apply the concept of virtualization and its types.

CO3: Experiment with virtualization of hardware resources and Docker.

CO4: Develop and deploy services on the cloud and set up a cloud environment.

CO5: Explain security challenges in the cloud environment. **TOTAL:60 PERIODS**

TEXT BOOKS

1. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.

2. James Turnbull, "The Docker Book", O'Reilly Publishers, 2014.

3. Krutz, R. L., Vines, R. D, "Cloud security. A Comprehensive Guide to Secure Cloud Computing", Wiley Publishing, 2010.

REFERENCES

1. James E. Smith, Ravi Nair, "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005.

2. Tim Mather, Subra Kumaraswamy, and Shahed Latif, "Cloud Security and Privacy: an enterprise perspective on risks and compliance", O'Reilly Media, Inc., 2009.

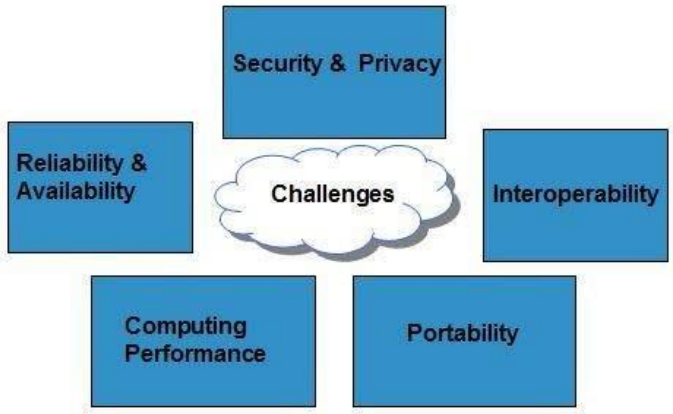
STAFF INCHARGE BC HOD PRINCIPAL

UNIT I
CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS: Cloud Architecture: System Models for Distributed and Cloud Computing – NIST Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models; Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

1. Tabulate the Design Challenges of Cloud Computing. Nov 2023



2. Distinguish between Public and Private Clouds. Nov 2023

- In a private cloud, a single organization controls and maintains the underlying infrastructure to deliver the IT resources.
- In a public cloud, external cloud providers deliver the resources as a fully managed service. For example, applications require computing resources like internal memory, data storage, and CPU.

3. Define Cloud.

- The cloud is an extensive network of remote servers around the world. These servers store and manage data, run applications, and deliver content and services like streaming videos, web mail, and office productivity software over the internet.

4. Define Cloud Computing? Dec 2021

- Cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale.

5. What are the different deployment model of cloud computing?

- Public Cloud
- Private Cloud
- Hybrid Cloud

- Community Cloud

6. List the Characteristics of Cloud computing? Nov 2020

- Cloud computing has some interesting characteristics that bring benefits to both cloud service consumers **(CSCs)** and cloud service providers **(CSPs).**
- These characteristics are
  - On-demand Self Service
  - Broad Network Access
  - Resource Pooling
  - Rapid Elasticity
  - Measured Service

7. What are the different hardware architectures for parallel processing?

The hardware architecture of parallel computing is distributed along the following categories as given below:

1. Single-instruction, single-data (SISD) systems
2. Single-instruction, multiple-data (SIMD) systems
3. Multiple-instruction, single-data (MISD) systems
4. Multiple-instruction, multiple-data (MIMD) systems Refer to learn about the hardware architecture of parallel computing

8. What is distributed computing?

- Distributed computing refers to a system where processing and data storage is distributed across multiple devices or systems, rather than being handled by a single central device.
- In a distributed system, each device or system has its own processing capabilities and may also store and manage its own data.
- These devices or systems work together to perform tasks and share resources, with no single device serving as the central hub.

9. What are the types of Cloud service model?

- Infrastructure as a Service
- Platform as a Service
- Software as a Service

10. What is elasticity in cloud computing?

- In cloud computing, elasticity is defined as "the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner".
- The dynamic adaptation of capacity, e.g., by altering the use of computing resources, to meet a varying workload is called "elastic computing".

11. What are the advantages of cloud services?

- **Cost:** It reduces the huge capital costs of buying hardware and software.

- **Speed:** Resources can be accessed quickly.
- **Scalability:** The requirement of resources can be increased or decreased according to the business requirements.
- **Productivity:** The IT team can be more productive and focus on achieving business goals due to less operational effort.
- **Reliability:** Backup and recovery of data are less expensive and very fast.
- **Security:** Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

12. List the companies who offer cloud service development?

- Amazon
- Google App Engine
- IBM
- Salesforce.com
- Microsoft

13. Why is Cloud Computing important?

- **For developers**,
  - Cloud computing provides increased amounts of storage and processing power to run the applications they develop.
  - Cloud computing also enables new ways to access information, process and analyze data, and connect people and resources from any location anywhere in the world.
- **For users**
  - Documents hosted in the cloud always exist, no matter what happens to the user's machine.
  - Users from around the world can collaborate on the same documents, applications, and projects, in real time. A
  - nd cloud computing does all this at lower costs, because the cloud enables more efficient sharing of resources than does traditional network computing.

14. What are the advantages and disadvantages of Cloud Computing?

Advantages

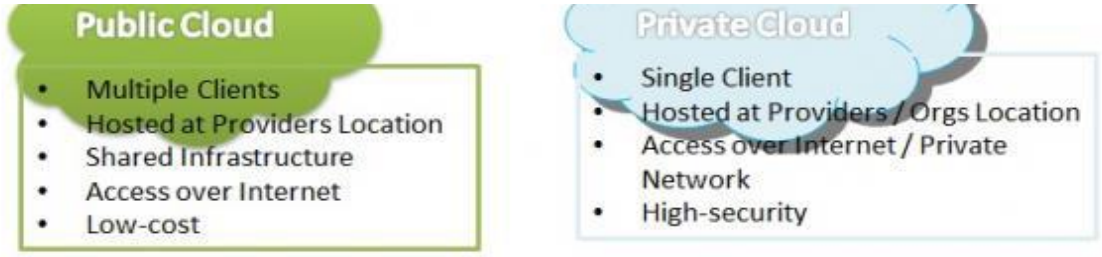
- Lower-Cost Computers for Users
- Improved Performance
- Lower IT Infrastructure Costs
- Fewer Maintenance Issues
- Lower Software Costs
- Instant Software Updates
- Increased Computing Power
- Unlimited Storage Capacity
- Increased Data Safety

- Improved Compatibility Between Operating Systems
- Improved Document Format Compatibility
- Easier Group Collaboration
- Universal Access to Documents
- Latest Version Availability
- Removes the Tether to Specific Devices

Disadvantages

- Requires a Constant Internet Connection
- Doesn't Work Well with Low-Speed Connections
- Can Be Slow
- Features Might Be Limited
- Stored Data Might Not Be Secure

15. Bring out the differences between private cloud and public cloud?



16. What is cloud service management?

- *Cloud Service Management* includes all of the service-related functions that are necessary for the management and operation of those services required by or proposed to cloud consumers.

17. What is on demand of cloud computing

- On-demand (OD) computing is an increasingly popular enterprise model in which computing resources are made available to the user as needed.
- The resources may be maintained within the user's enterprise, or made available by a service provider

18. What is Cloud Scalability?

- Cloud scalability is the ability of the system's infrastructure to handle growing workload requirements while retaining a consistent performance adequately.

19. What is the difference between Cloud Elasticity and Cloud Scalability?

Cloud Elasticity	Cloud Scalability
• Cloud Elasticity is a tactical resource allocation operation. • It provides the necessary resources required for the current task and handles varying loads for short periods.	• Cloud Scalability is a strategic resource allocation operation. • Scalability handles the increase and decrease of resources according to the system's workload demands.

• For example, running a sentiment analysis algorithm, doing database backups or just taking on user traffic surges on a website	

20. Depict the importance of on-demand provisioning in e-commerce applications. DEC 2021

- On-demand computing (ODC) is a delivery model in which computing resources are made available to the user as needed.
- The resources may be maintained within the user's enterprise or made available by a cloud service provider.
- E-commerce stores like Amazon and eBay were the first on-demand apps.

Importance of on-demand provisioning in e-commerce applications.

- Mobile-push for change
- Overwhelming Social Media penetration
- Addressing value as well as a convenience
- Frictionless business process

Part-B

1. **Define cloud and cloud computing? Explain the Architecture of Cloud Computing.**

➢ **Cloud**

- The word "cloud" refers to the datacenter full of servers connected to the Internet performing a service. Cloud servers are located in data centers all over the world.

➢ **Cloud Computing**

- Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction.
- This cloud model is composed of five essential characteristics, three service models, and four deployment models.

➢ **Essential characteristics**

- On-demand self-service
- Ubiquitous network access
- Resource pooling
- Location independence
- Rapid elasticity
- Measured service

➢ **Service models**

- Cloud Software as a Service (SaaS)—Use provider's applications over a network.
- Cloud Platform as a Service (PaaS)—Deploy customer-created applications to a cloud.
- Cloud Infrastructure as a Service (IaaS)—Rent processing, storage, network capacity, and other fundamental computing resources.

➢ **Deployment models**

- Private cloud—Enterprise owned or leased
- Community cloud—Shared infrastructure for specific community
- Public cloud—Sold to the public, mega-scale infrastructure
- Hybrid cloud—Composition of two or more clouds

➢ **Cloud Computing Architecture**

Refer Figure 1.1 for basic architecture of cloud.

- Cloud Deployment Model
- Cloud Service Model
- Essential Characteristics of Cloud Computing

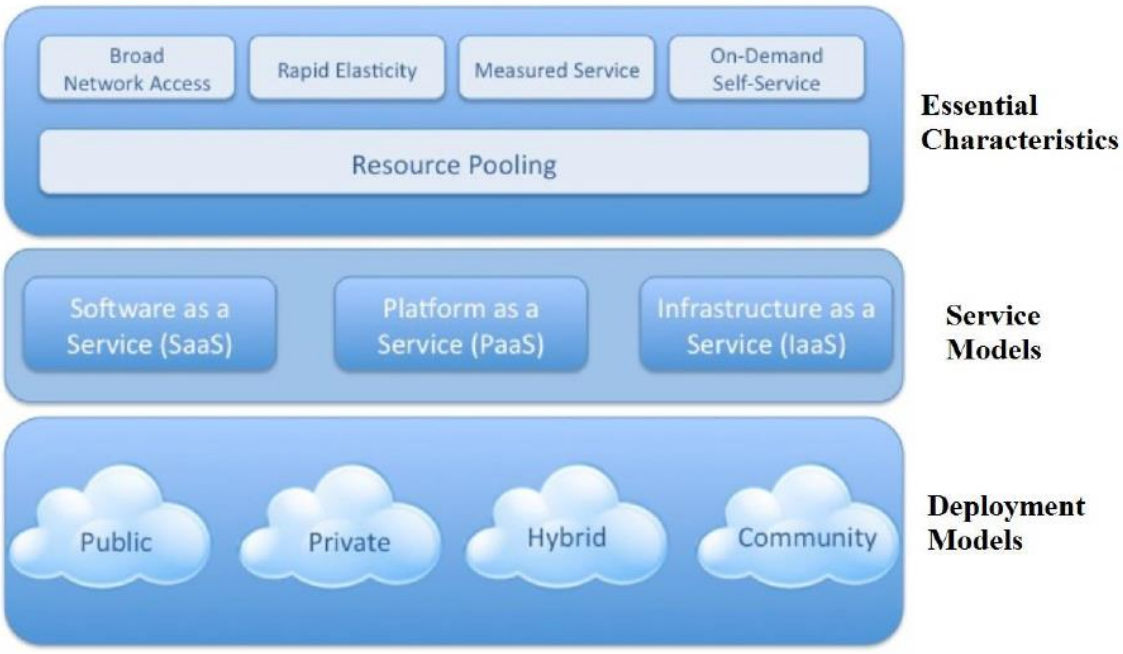


Figure 1.1 – Basic Architecture of Cloud

➢ **Advantages of cloud computing**

- **Cost:** It reduces the huge capital costs of buying hardware and software.
- **Speed:** Resources can be accessed quickly.
- **Scalability:** The requirement of resources can be increased or decreased according to the business requirements.
- **Productivity:** The IT team can be more productive and focus on achieving business goals due to less operational effort.
- **Reliability:** Backup and recovery of data are less expensive and very fast.
- **Security:** Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Disadvantages

- Ongoing operating costs
- Security
- Dependency on Internet connection
- Vendor lock-in

2. List out characteristics of cloud computing?

- **Characteristics of Cloud Computing**
  - Cloud computing has some interesting characteristics that bring benefits to both cloud service consumers **(CSCs)** and cloud service providers **(CSPs).**
  - These characteristics are
    - On-demand Self Service
    - Broad Network Access
    - Resource Pooling
    - Rapid Elasticity
    - Measured Service

1. **On-demand Self Service**
   - Cloud Computing services are available on-demand and do not require much human interaction.
   - The user himself can provision, manage, and monitor the resources as per his requirement.
   - This is done through a web-based self-service management console.
   - The customer can create the service on his own, like creating a new mailbox or adding additional disk space to a virtual machine, etc.
   - **For example**, for booking a ticket on a travel portal, a passenger gets the flexibility to book his ticket by himself without any human interaction. Right from choosing the flight to preference class, the process is entirely automated and does not require any salesperson in between.
2. **Broad Network Access**
   - Cloud computing is accessible from a network, generally over the internet.
   - Similarly, private cloud services can be accessed from anywhere within the enterprise.
   - The services are provided over heterogeneous devices such as mobile phones, laptops, tablets, office computers, etc.
   - The user can access the existing data on a cloud platform or upload new data on the cloud from anywhere using a device and internet connection.
   - In the above example, the passenger can book his ticket via the internet from any device like a smartphone, laptop, tablet, etc., which has access to a network.
3. **Resource Pooling and Multi-tenancy**
   - Computing resources like networks, servers, storage, applications, and service can be pooled to serve multiple consumers by securely separating the resources on a logical level.
   - This is done using a multi-tenant model, which allows multiple customers to share the same application or physical infrastructure while retaining data security and privacy.

- Same example of the travel portal, the flights can carry several passengers in a single trip. These passengers travel to the same destination, board the same flight, and are allotted separate seats as per the demand and requirement.

4. **Rapid Elasticity and Scalability**
   - Resource capabilities can be elastically provisioned and released to meet immediate requirements.
   - Similarly, they can be removed or scaled-down when not required.
   - Scalability adds a cost-effectiveness aspect to cloud technology. When the demand or workload is high, more servers can be added for that particular period.
   - For example, to meet the demand of the increasing number of passengers, an airline can increase the number of flights for a particular time and stop the flights when the demand goes down.
5. **Measured Service**
   - The utilization of resources is tracked, monitored, controlled, and reported for each occupant.
   - This gives transparency to both the service provider and the consumer.
   - The cloud system has a metering capability, which is leveraged to monitor billing, use of resources, and pay only for what has been used.
   - When a passenger is traveling by train, he has to pay only for the distance traveled by him and not for the entire journey that the train takes.

3. Explain the service models of cloud computing in detail.

The service models are categorized into three basic models:

1) Software-as-a-Service (SaaS)
2) Platform-as-a-Service (PaaS)
3) Infrastructure-as-a-Service (IaaS)

The Figure 1.2 represents Cloud Service Model for different service levels.

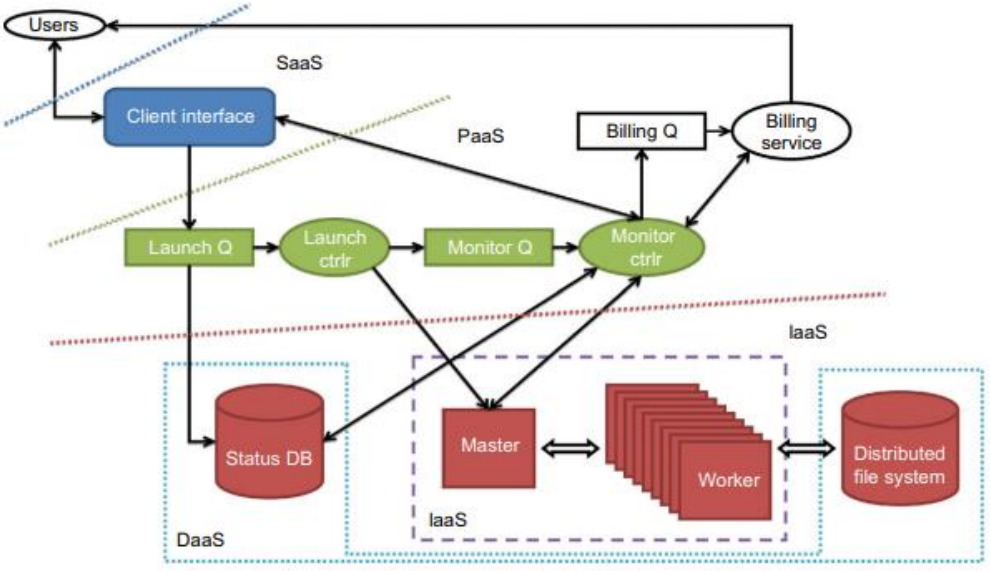


Figure 1.2 – Cloud Service Model for different service levels

1) Software-as-a-Service (SaaS)

- SaaS is known as **'On-Demand Software'.**
- It is a software distribution model. In this model, the applications are hosted by a cloud service provider and publicized to the customers over internet.
- In SaaS, associated data and software are hosted centrally on the cloud server.
- The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure.
- User can access SaaS by using a thin client through a web browser.
- The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, storage, or even individual application'. Refer Figure 1.3 for example representation.
- CRM, Office Suite, Email, games, etc. are the software applications which are provided as a service through Internet.

Figure 1.3 – SaaS Model

SaaS providers

- Google's Gmail, Docs, Talk etc
- Microsoft's Hotmail, Sharepoint
- SalesForce,
- Yahoo, Facebook

Advantages of SaaS

- SaaS is easy to buy because the pricing of SaaS is based on monthly or annual fee.
- SaaS needed less hardware, because the software is hosted remotely.
- Less maintenance cost is required.

Disadvantages of SaaS

- SaaS applications are totally dependent on Internet connection.
- It is difficult to switch amongst the SaaS vendors.

2) Platform-as-a-Service (PaaS)

- PaaS is a programming platform for developers.
- This platform is generated for the programmers to create, test, run and manage the applications.
- A developer can easily write the application and deploy it directly into PaaS layer.
- PaaS gives the runtime environment for application development and deployment tools.
- The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider.
- The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

- The PaaS vendor provides several services for application developers:
  - A virtual development environment
  - Application standards, usually based on the developers' requirements
  - Toolkits configured for the virtual development environment
  - A ready-made distribution channel for public application developers

- Refer Figure 1.4 for PaaS Model Example.

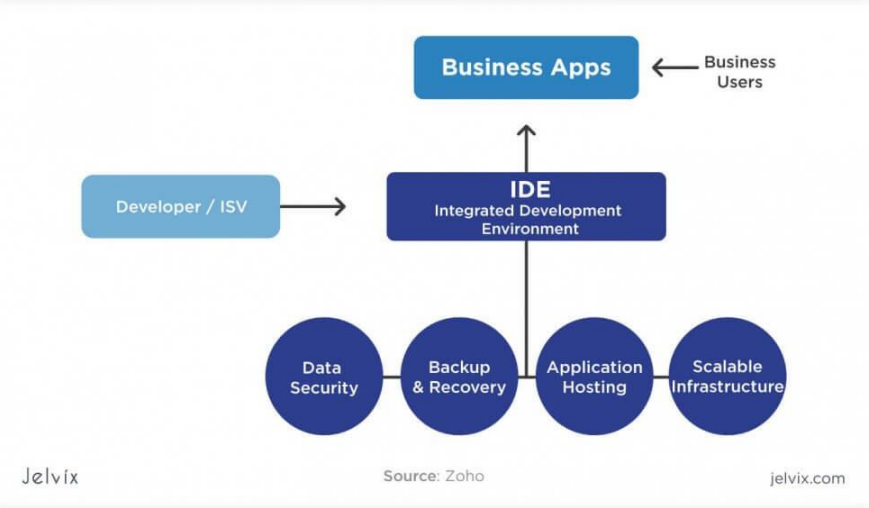


Figure 1.4 – PaaS Model

PaaS providers

- Google App Engine
  - Python, Java, Eclipse
- Microsoft Azure
  - .Net, Visual Studio
- Sales Force
  - Apex, Web wizard
- TIBCO,
- VMware,
- Zoho

Advantages of PaaS

- PaaS is easier to develop.
- In PaaS, developer only requires a PC and an Internet connection to start building applications.

Disadvantages of PaaS

- Moving the application to another PaaS vendor is a problem.

3) Infrastructure-as-a-Service (IaaS)

- The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.
- The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components.
- IaaS is a way to deliver a cloud computing infrastructure like server, storage, network and operating system.

- The customers can access these resources over cloud computing platform i.e Internet as an on-demand service.
- Refer Figure 1.5 for IaaS Model

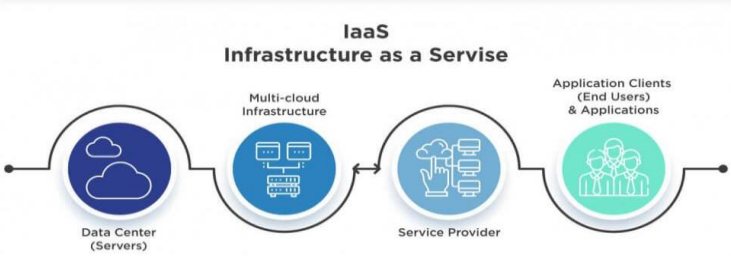


Figure 1.5 – IaaS Model

IaaS providers

- Amazon Elastic Compute Cloud (EC2)
- RackSpace Hosting
- Joyent Cloud
- Go Grid

Advantages of IaaS

- In IaaS, user can dynamically choose a CPU, memory storage configuration according to need.
- Users can easily access the vast computing power available on IaaS Cloud platform.

Disadvantages of IaaS

- IaaS cloud computing platform model is dependent on availability of Internet and virtualization services.

4. Anything as a Service (XAAS)

- ***Storage as a Service (SAAS)***
  - Storage as a Service is a business model in which a large company rents space in their storage infrastructure to a smaller company or individual.
  - Storage as a Service is generally seen as a good alternative for a small or mid-sized business that lacks the capital budget and/or technical personnel to implement and maintain their own storage infrastructure.
- ***Communications as a Service (CAAS)***
  - Communications as a Service (CAAS) is an outsourced enterprise communications solution that can be leased from a single vendor.
  - Such communications can include voice over IP (VoIP or Internet telephony), instant messaging (IM), collaboration and video conference applications using fixed and mobile devices.
  - CAAS allows businesses to selectively deploy communications devices and modes on a pay-as-you-go, as-needed basis.

- ***Network as a Service (NAAS)***
  - NAAS is a new cloud computing model in which the clients have access to additional computing resources collocated with switches and routers.
  - NAAS can include flexible and extended Virtual Private Network (VPN), bandwidth on demand, custom routing, multicast protocols, security firewall, intrusion detection and prevention, Wide Area Network (WAN), content monitoring and filtering, and antivirus.
- ***Monitoring as a Service (MAAS)***
  - Monitoring-as-a-service (MAAS) is a framework that facilitates the deployment of monitoring functionalities for various other services and applications within the cloud.
  - The most common application for MAAS is online state monitoring, which continuously tracks certain states of applications, networks, systems, instances or any element that may be deployable within the cloud.

Refer Figure 1.6 for the comparison of Cloud Service Model

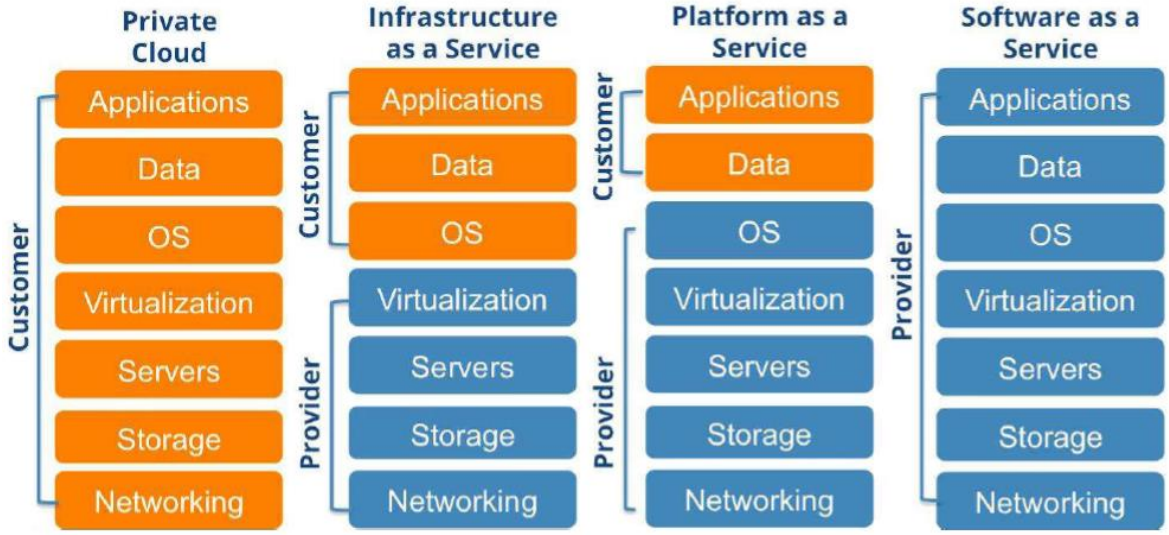


Figure 1.6 – Cloud Service Model Comparison

4. Explain in details about deployment model of cloud computing?
Discuss the various cloud service, deployment models with neat sketch.
Nov 2023

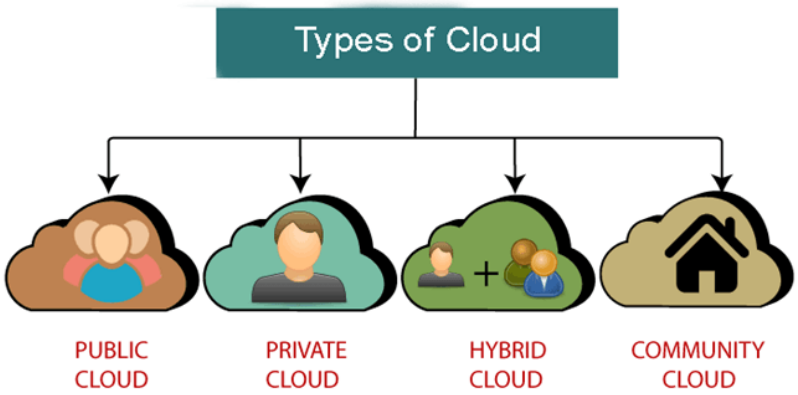


Figure 1.7 – Cloud Deployment Model

Deployment models of Cloud:

Refer Figure 1.7 for the various types of Deployment in cloud.

- **Public Cloud**
  - Public cloud is **open to all** to store and access information via the Internet using the pay-per-usage method.
  - In public cloud, computing resources are managed and operated by the Cloud Service Provider (CSP).
  - A public cloud is built over the Internet and can be accessed by any user who has paid for the service.
  - Public clouds are owned by service providers and are accessible through a subscription. Refer Figure 1.8.
  - **Example:** Amazon elastic compute cloud (EC2), IBM Smart Cloud Enterprise, Microsoft, Google App Engine, Windows Azure Services Platform.

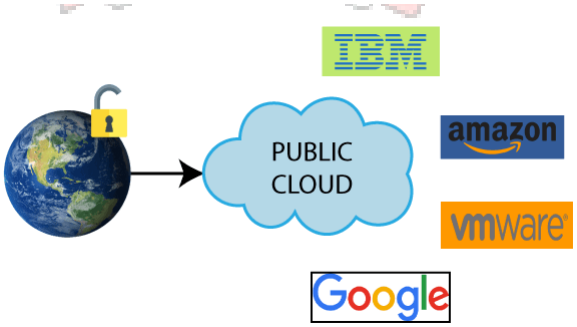


Figure 1.8 – Public Cloud

Advantages of Public Cloud

- Owned at a lower cost than the private and hybrid cloud.
- Maintained by the cloud service provider.
- Easier to integrate.
- Is location independent.
- Highly scalable.
- No limit to the number of users.

Disadvantages of Public Cloud

- It is less secure because resources are shared publicly.
- Performance depends upon the high-speed internet network link to the cloud provider.
- The Client has no control of data.

➢ **Private Cloud**

- Private cloud is also known as an **internal cloud** or **corporate cloud**.
- It is used by organizations to build and manage their own data centers internally or by the third party.
- A private cloud is built within the domain of an intranet owned by a single organization.
- Therefore, it is client owned and managed, and its access is limited to the owning clients and their partners.
- A private cloud is supposed to deliver more efficient and convenient cloud services.
- It can be deployed using Opensource tools such as Openstack and Eucalyptus. Refer Figure 1.9.
- Based on the location and management, National Institute of Standards and Technology (NIST) divide private cloud into the following two parts-
  - On-premise private cloud
  - Outsourced private cloud

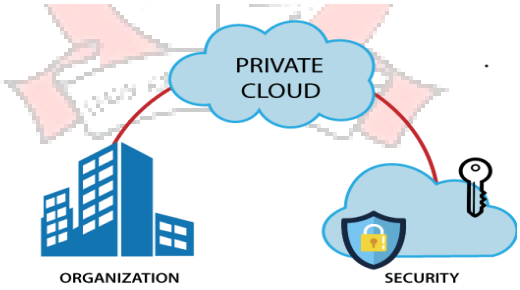


Figure 1.9 – Private Cloud

Advantages of Private Cloud

- Private cloud provides a high level of security and privacy to the users.
- Private cloud offers better performance with improved speed and space capacity.
- It allows the IT team to quickly allocate and deliver on-demand IT resources.
- The organization has full control over the cloud because it is managed by the organization itself.
- It is suitable for organizations that require a separate cloud for their personal use and data security is the first priority.

Disadvantages of Private Cloud

- Skilled people are required to manage and operate cloud services.
- Private cloud is accessible within the organization, so the area of operations is limited.
- Private cloud is not suitable for organizations that have a high user base, and organizations that do not have the prebuilt infrastructure, sufficient manpower to maintain and manage the cloud.

3. Hybrid Cloud

- Hybrid Cloud is a combination of the public cloud and the private cloud.

Hybrid Cloud = Public Cloud + Private Cloud

- Hybrid cloud is partially secure because the services which are running on the public cloud can be accessed by anyone, while the services which are running on a private cloud can be accessed only by the organization's users.
- A hybrid cloud provides access to clients, the partner network, and third parties. Refer Figure 1.10 for Hybrid Cloud
- **Example:** Google Application Suite (Gmail, Google Apps, and Google Drive), Office 365 (MS Office on the Web and One Drive), Amazon Web Services.

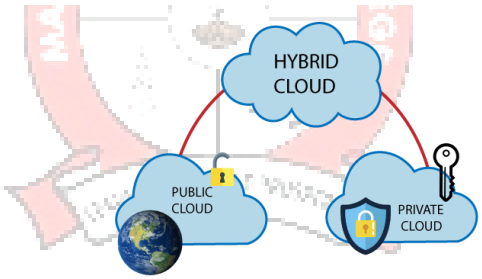


Figure 1.10 – Hybrid Cloud

Advantages of Hybrid Cloud

- It is suitable for organizations that require more security than the public cloud.
- It helps you to deliver new products and services more quickly.
- It provides an excellent way to reduce the risk.
- It offers flexible resources because of the public cloud and secure resources because of the private cloud.

Disadvantages of Hybrid Cloud

- In Hybrid Cloud, security feature is not as good as the private cloud.
- Managing a hybrid cloud is complex because it is difficult to manage more than one type of deployment model.
- In the hybrid cloud, the reliability of the services depends on cloud service providers.

4. Community Cloud

- Community cloud allows systems and services to be accessible by a group of several organizations to share the information between the organization and a specific community.
- It is owned, managed, and operated by one or more organizations in the community, a third party, or a combination of them.
- Refer Figure 1.11 for Community Cloud
- **Example:** Health Care community cloud

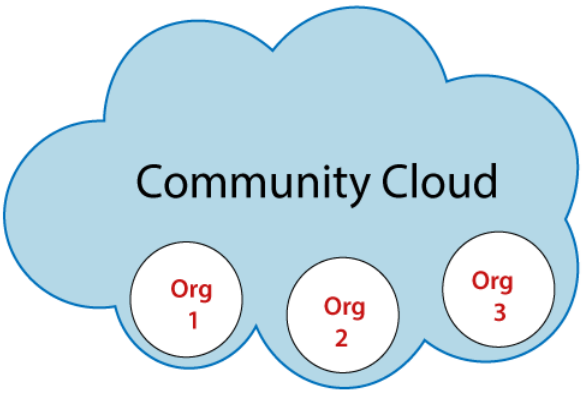


Figure 1.11– Community Cloud

Advantages of Community Cloud

There are the following advantages of Community Cloud -

- Community cloud is cost-effective because the whole cloud is being shared by several organizations or communities.
- Community cloud is suitable for organizations that want to have a collaborative cloud with more security features than the public cloud.
- It provides better security than the public cloud.
- It provides collaborative and distributive environment.
- Community cloud allows us to share cloud resources, infrastructure, and other capabilities among various organizations.

Disadvantages of Community Cloud

- Community cloud is not a good choice for every organization.
- Security features are not as good as the private cloud.
- It is not suitable if there is no collaboration.
- The fixed amount of data storage and bandwidth is shared among all community members.

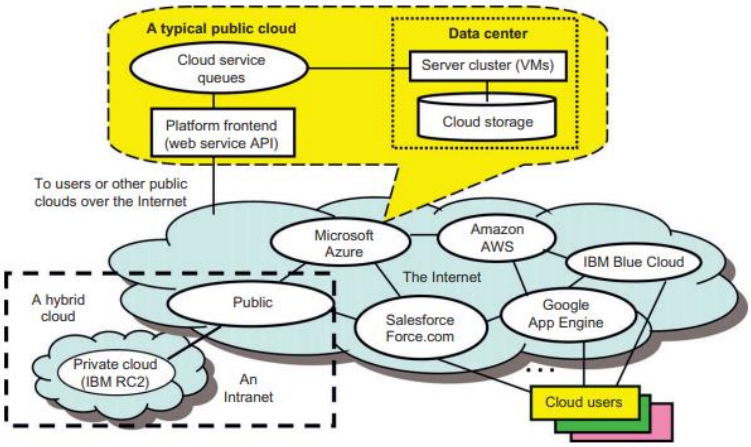


Figure 1.12– Cloud deployment models

Difference between public cloud, private cloud, hybrid cloud, and community cloud – Refer Figure 1.12 and Table 1.1

Table 1.1 - Comparison of cloud deployment models

Parameter	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
Host	Service provider	Enterprise (Third party)	Enterprise (Third party)	Community (Third party)
Users	General public	Selected users	Selected users	Community members
Access	Internet	Internet, VPN	Internet, VPN	Internet, VPN
Owner	Service provider	Enterprise	Enterprise	Community

5. Explain in detail about NIST Cloud Computing reference architecture. (Or) Describe the NIST cloud computing reference architecture with its components. **Nov 2023**

The Conceptual Reference Model

- NIST cloud computing reference architecture, which identifies the major actors, their activities and functions in cloud computing.

- The figure 1.13 depicts a generic high-level architecture and is intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing.

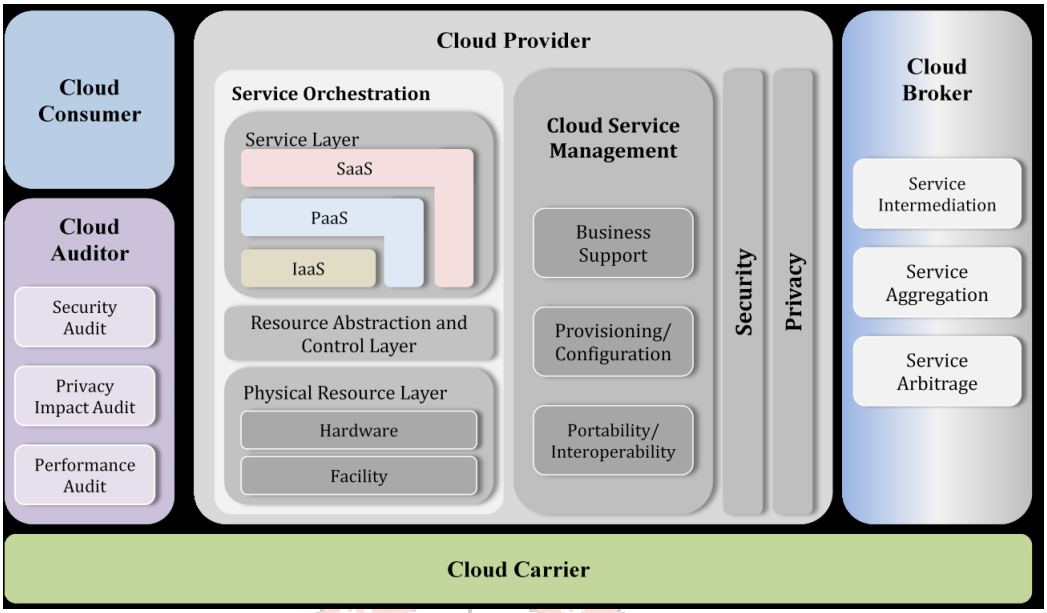


Figure 1.13– NIST Cloud Computing reference architecture

The NIST cloud computing reference architecture defines five major actors:

- *cloud consumer,*
- *cloud provider,*
- *cloud carrier,*
- *cloud auditor* and
- *cloud broker.*

Each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in cloud computing. Figure 1.14 depicts the Interaction between different actors in Cloud

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, *Cloud Providers*.
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between *Cloud Providers* and *Cloud Consumers*.
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from *Cloud Providers* to *Cloud Consumers*.

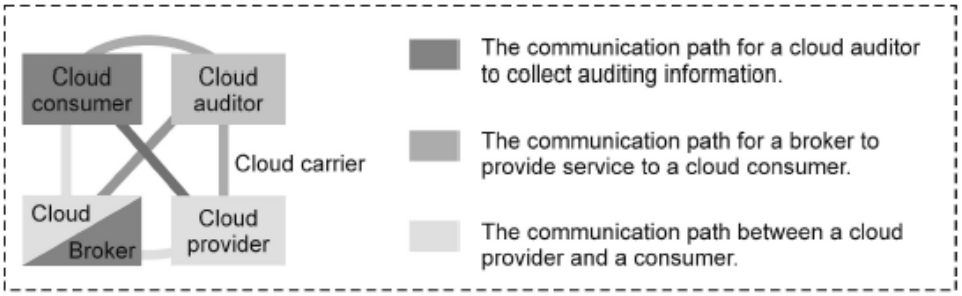


Figure 1.14– Interaction between different actors in Cloud

Cloud Consumer

- The cloud consumer is the principal stakeholder for the cloud computing service.
- A cloud consumer represents a person or organization that maintains a business relationship with, and uses the service from a cloud provider.
- A cloud consumer browses the service catalog from a cloud provider, requests the appropriate service, sets up service contracts with the cloud provider, and uses the service.
- The cloud consumer may be billed for the service provisioned, and needs to arrange payments accordingly.
- A cloud consumer can freely choose a cloud provider with better pricing and more favorable terms.

Cloud Provider

- A cloud provider is a person, an organization; it is the entity responsible for making a service available to interested parties.
- A Cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the Cloud Consumers through network access.
- A cloud provider conducts its activities in the areas of service deployment, service orchestration, cloud service management, security, and privacy.

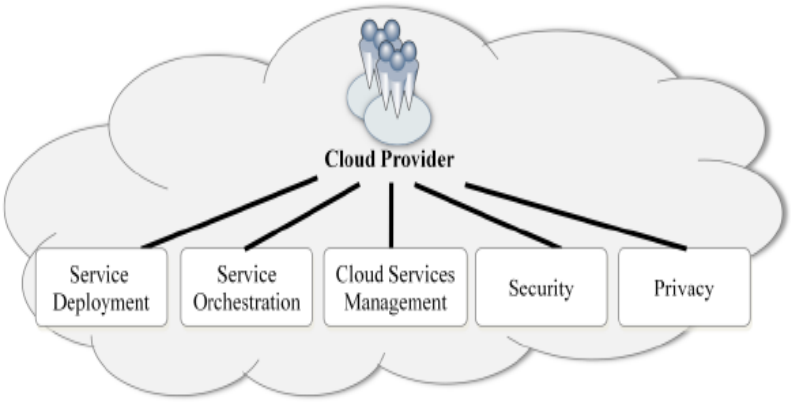


Figure 1.15 – Major activities of a cloud provider

The major activities of a cloud provider include :

- **Service deployment :** Service deployment refers to provisioning private, public, hybrid and community cloud models.
- **Service orchestration :** Service orchestration implies the coordination, management of cloud infrastructure and arrangement to offer optimized capabilities of cloud services. The capabilities must be cost-effective in managing IT resources and must be determined by strategic business needs.
- **Cloud services management :** This activity involves all service-related functions needed to manage and operate the services requested or proposed by cloud consumers.
- **Security :** Security, which is a critical function in cloud computing, spans all layers in the reference architecture. CSPs must take care of security.
- **Privacy :** Privacy in cloud must be ensured at different levels, such as user privacy, data privacy, authorization and authentication and it must also have adequate assurance levels.

Cloud Auditor

- A cloud auditor is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon.
- Audits are performed to verify conformance to standards through review of objective evidence.
- A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance, etc.

Cloud Broker

- As cloud computing evolves, the integration of cloud services can be too complex for cloud consumers to manage.
- A cloud consumer may request cloud services from a cloud broker, instead of contacting a cloud provider directly.
- A cloud broker is an entity that manages the use, performance and delivery of cloud
- services and negotiates relationships between cloud providers and cloud consumers.
- In general, a cloud broker can provide services in three categories:
  - **Service intermediation :** Here the cloud broker will improve some specific capabilities, and provide value added services to cloud consumers.
  - **Service aggregation :** The cloud broker links and integrates different services into one or more new services.
  - **Service Arbitrage :** In service arbitrage, the broker has the liberty to choose services from different agencies.

Cloud Carrier

- A cloud carrier acts as an intermediary that provides connectivity and transport of cloud services between cloud consumers and cloud providers.
- Cloud carriers provide access to consumers through network, telecommunication and other access devices.
- The distribution of cloud services is normally provided by network and telecommunication carriers or a *transport agent*, where a transport agent refers to a business organization that provides physical transport of storage media such as high capacity hard drives.
- A cloud provider will set up SLAs with a cloud carrier to provide services consistent with the level of SLAs offered to cloud consumers, and may require the cloud carrier to provide dedicated and secure connections between cloud consumers and cloud providers.

6. Explain the various design challenges for effective cloud computing environment. NOV 2020
Explain the various architectural design challenges in Cloud.

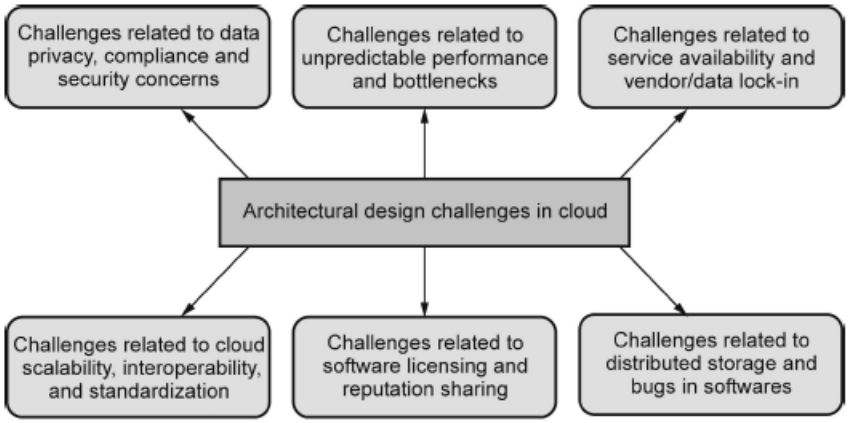


Figure 1.16 – Architecture design challenges

Challenge 1—Service Availability and Data Lock-in Problem

- The management of a cloud service by a single company is often the source of **single points of failure**.
- If a company has multiple data centers located in different geographic regions, it may have common software infrastructure and accounting systems. Therefore, using multiple cloud providers may provide more protection from failures.
- Another availability obstacle is **distributed denial of service (DDoS) attacks.**
- Criminals threaten to cut off the incomes of SaaS providers by making their services unavailable.
- The obvious solution is to standardize the APIs so that a SaaS developer can deploy services and data across multiple cloud providers. This will rescue the loss of all data due to the failure of a single company.
- In addition to mitigating data lock-in concerns, standardization of APIs enables a new usage model in which the **same software infrastructure can be used in both** public and private clouds.
- Such an option could enable "**surge computing**," in which the public cloud is used to capture the extra tasks that cannot be easily run in the data center of a private cloud.

Challenge 2—Data Privacy and Security Concerns

- Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks.
- For example, encrypt data before placing it in a cloud.
- Many nations have laws requiring SaaS providers to keep customer data and copyrighted material within national boundaries.
- Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms.

- In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits.
- Another type of attack is the man-in-the-middle attack for VM migrations.
- Passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

Challenge 3—Unpredictable Performance and Bottlenecks

- Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic.
- One solution is to improve I/O architectures and operating systems to efficiently virtualize interrupts and I/O channels.
- Internet applications continue to become more data-intensive.
- Therefore, data transfer bottlenecks must be removed,

Challenge 4—Distributed Storage and Widespread Software Bugs

- The database is always growing in cloud applications.
- The opportunity is to create a storage system that will not only meet this growth, but also combine it with the cloud advantage of scaling arbitrarily up and down on demand.
- This demands the design of efficient distributed SANs.
- Large-scale distributed bugs cannot be reproduced, so the debugging must occur at a scale in the production data centers. No data center will provide such a convenience.
- The level of virtualization may make it possible to capture valuable information in ways that are impossible without using VMs.
- Debugging over simulators is another approach to attacking the problem, if the simulator is well designed.

Challenge 5—Cloud Scalability, Interoperability, and Standardization

- The pay-as-you-go model applies to storage and network bandwidth; both are counted in terms of the number of bytes used.
- Computation is different depending on virtualization level.
- GAE automatically scales in response to load increases and decreases; users are charged by the cycles used.
- AWS charges by the hour for the number of VM instances used, even if the machine is idle.
- The opportunity here is to scale quickly up and down in response to load variation, in order to save money, but without violating SLAs.

- Open Virtualization Format (OVF) describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of VMs. It also defines a format for distributing software to be deployed in VMs.
- This VM format does not rely on the use of a specific host platform, virtualization platform, or guest operating system.
- The approach is to address virtual platform-agnostic packaging with certification and integrity of packaged software. The package supports virtual appliances to span more than one VM.
- Also need to enable VMs to run on heterogeneous hardware platform hypervisors. This requires hypervisor-agnostic VMs.

Challenge 6—Software Licensing and Reputation Sharing

- Many cloud computing providers originally relied on open source software because the licensing model for commercial software is not ideal for utility computing.
- The primary opportunity is either for open source to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing.
- One can consider using both pay-for-use and bulk-use licensing schemes to widen the business coverage.
- One customer's bad behavior can affect the reputation of the entire cloud.
- Cloud providers want legal liability to remain with the customer, and vice versa. This problem must be solved at the SLA level.

7. Explain in details about Architectural Design of Compute and Storage Clouds.

➢ **A Generic Cloud Architecture Design**

- An Internet cloud is a public cluster of servers provisioned on demand to perform collective web services or distributed applications.

Cloud Platform Design Goals

- Scalability, virtualization, efficiency, and reliability are four major design goals of a cloud computing platform.
- Cloud management receives the user request, finds the correct resources, and then calls the provisioning services which invoke the resources in the cloud.
- The cloud management software needs to support both physical and virtual machines.
- Security in shared resources and shared access of data centers also pose another design challenge.

- The hardware and software systems are combined to make it easy and efficient to operate.
- System scalability is achieved if one service takes a lot of processing power, storage capacity, or network traffic, it is simple to add more servers and bandwidth.
- System reliability is attained if data can be put into multiple locations. For example, user e-mail can be put in three disks which expand to different geographically separate data centers.
- In such a situation, even if one of the data centers crashes, the user data is still accessible.

Enabling Technologies for Clouds

- Cloud users are able to demand more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers can increase system utilization via multiplexing, virtualization, and dynamic resource provisioning.
- Clouds are enabled by the progress in hardware, software, and networking technologies summarized in Table 1.2.

Table 1.2 – Cloud Enabling Technologies in Hardware, Software and Networking

Technology	Requirements and Benefits
Fast platform deployment	Fast, efficient, and flexible deployment of cloud resources to provide dynamic computing environment to users
Virtual clusters on demand	Virtualized cluster of VMs provisioned to satisfy user demand and virtual cluster reconfigured as workload changes
Multitenant techniques	SaaS for distributing software to a large number of users for their simultaneous use and resource sharing if so desired
Massive data processing	Internet search and web services which often require massive data processing, especially to support personalized services
Web-scale communication	Support for e-commerce, distance education, telemedicine, social networking, digital government, and digital entertainment applications
Distributed storage	Large-scale storage of personal records and public archive information which demands distributed storage over the clouds
Licensing and billing services	License management and billing services which greatly benefit all types of cloud services in utility computing

A Generic Cloud Architecture

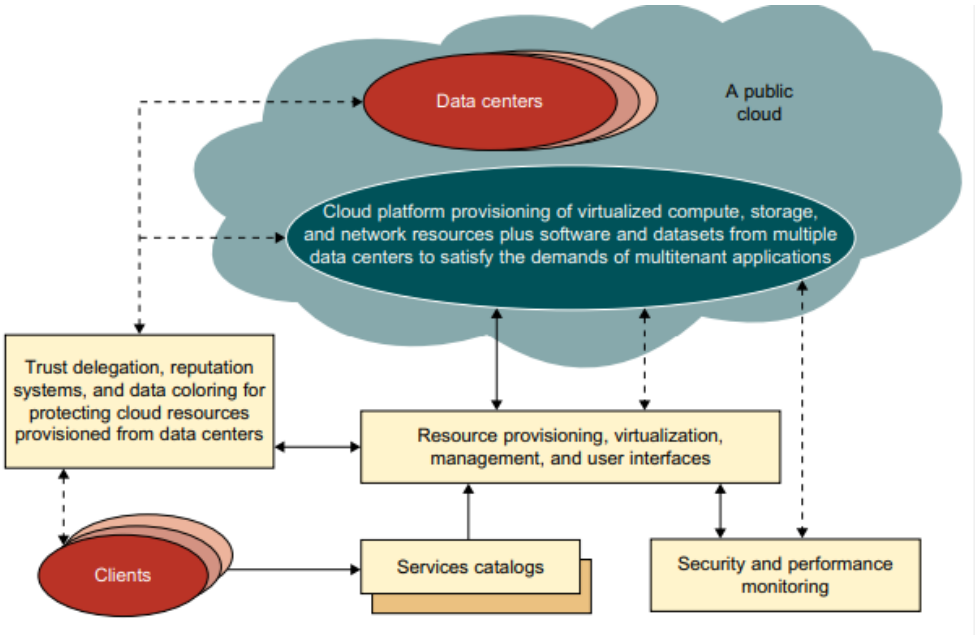


Figure 1.17 – A security-aware cloud platform

- Figure 1.17 shows a security-aware cloud architecture.
- The Internet cloud is envisioned as a massive cluster of servers.
- The cloud platform is formed dynamically by provisioning or deprovisioning servers, software, and database resources.
- The cloud computing resources are built into the data centers, which are typically owned and operated by a third-party provider.
- In a cloud, software becomes a service.
- The cloud demands a high degree of trust of massive amounts of data retrieved from large data centers. Need to build a framework to process large-scale data stored in the storage system.
- This demands a distributed file system over the database system.
- Other cloud resources are added into a cloud platform, including storage area networks (SANs), database systems, firewalls, and security devices.
- Web service providers offer special APIs that enable developers to exploit Internet clouds.
- Monitoring and metering units are used to track the usage and performance of provisioned resources.
- The software infrastructure of a cloud platform must handle all resource management and do most of the maintenance automatically.
- Cloud computing providers, such as Google and Microsoft, have built a large number of data centers all over the world. Each data center may have thousands of servers. The location of the data center is chosen to reduce power and cooling costs. Thus, the data centers are often built around hydroelectric power.

- In general, private clouds are easier to manage, and public clouds are easier to access.

➢ **Layered Cloud Architectural Development**

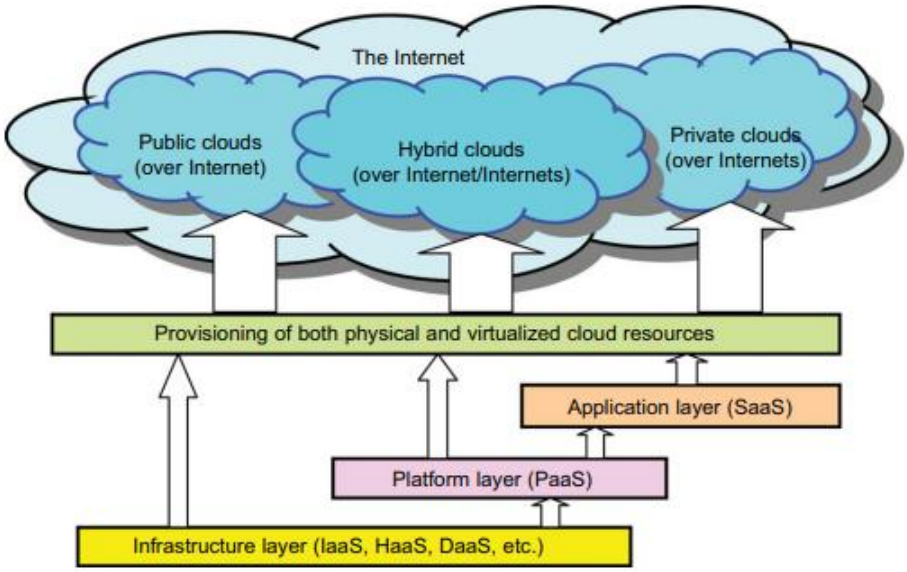


Figure 1.18 – Layered architectural development of the cloud platform

- The architecture of a cloud is developed at three layers: infrastructure, platform, and application, as demonstrated in Figure 1.18.
- These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud.
- The services to public, private, and hybrid clouds are conveyed to users through networking support over the Internet and intranets involved.
- The infrastructure layer is deployed first to support IaaS services. This infrastructure layer serves as the foundation for building the platform layer of the cloud for supporting PaaS services.
- In turn, the platform layer is a foundation for implementing the application layer for SaaS applications.
- The infrastructure layer is built with virtualized compute, storage, and network resources.
- The abstraction of these hardware resources is meant to provide the flexibility demanded by users.
- The platform layer is for general-purpose and repeated usage of the collection of software resources.

- This layer provides users with an environment to develop their applications, to test operation flows, and to monitor execution results and performance.
- The platform should be able to assure users that they have scalability, dependability, and security protection.
- In a way, the virtualized cloud platform serves as a "system middleware" between the infrastructure and application layers of the cloud.

➢ **Architectural Design Challenges**

- Challenge 1—Service Availability and Data Lock-in Problem
- Challenge 2—Data Privacy and Security Concerns
- Challenge 3—Unpredictable Performance and Bottlenecks
- Challenge 4—Distributed Storage and Widespread Software Bugs
- Challenge 5—Cloud Scalability, Interoperability, and Standardization
- Challenge 6—Software Licensing and Reputation Sharing

8. What is distributed computing? Describe about components for distributed computing.

➢ **Distributed Computing**:

- In distributed systems there is no shared memory and computers communicate with each other through message passing.
- In distributed computing a single task is divided among different computers.

➢ **Distributed System Architectures**

- Distributed system architectures are bundled up with components and connectors.
- Components can be individual nodes or important components in the architecture whereas connectors are the ones that connect each of these components.
- **Component:** A modular unit with well-defined interfaces; replaceable; reusable
- **Connector:** A communication link between modules which mediates coordination or cooperation among components
- Figure 1.19 depicts the representation of components and connectors

Figure 1.19 – Components and Connectors

➢ **Architectural Styles**

- Layered Architecture

- Object Based Architecture
- Data-centered Architecture
- Event Based Architecture
- Hybrid Architecture

Layered Architecture

- The layered architecture separates layers of components from each other, giving it a much more modular approach.
- The layers on the bottom provide a service to the layers on the top. The request flows from top to bottom, whereas the response is sent from bottom to top.
- The advantage of using this approach is that, the calls always follow a predefined path, and that each layer can be easily replaced or modified without affecting the entire architecture.
- The figure 1.20 is the basic idea of a layered architecture style.

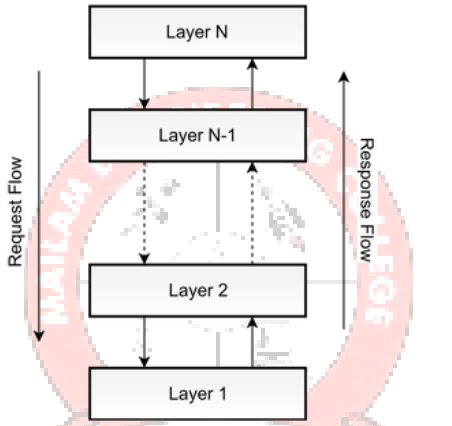


Figure 1.20 – Layered Architecture

Object Based Architecture

- This architecture style is based on loosely coupled arrangement of objects.
- Each of the components are referred to as objects, where each object can interact with other objects through a given connector or interface as in figure 1.21.
- These are much more direct where all the different components can interact directly with other components through a direct method call.

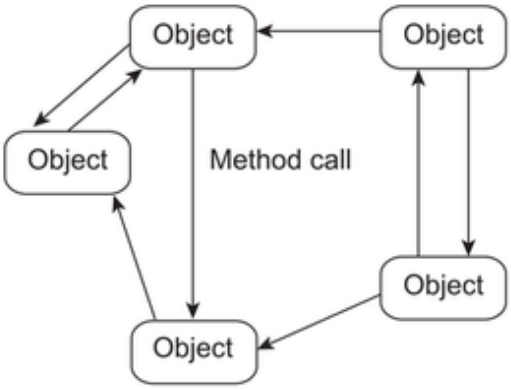


Figure 1.21 – Object Based Architecture

Data Centered Architecture

- This architecture is based on a data center, where the primary communication happens via a central data repository as shown in figure 1.21..
- This common repository can be either active or passive.
- This supports different components (or objects) by providing a persistent storage space for those components (such as a MySQL database).
- All the information related to the nodes in the system are stored in this persistent storage.

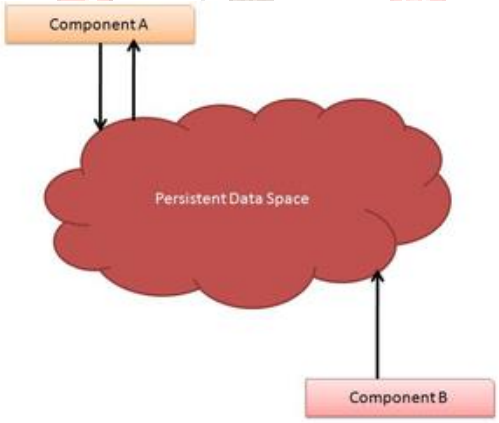


Figure 1.21 – Data Centered Architecture

Event Based Architecture

- The entire communication of a system happens through events.
- When an event is generated, it will be sent to the bus system as shown in figure 1.22..
- If anyone is interested, that node can pull the event from the bus and use it. Sometimes these events could be data, or even URLs to resources.
- So the receiver can access whatever the information is given in the event and process accordingly.

- These events occasionally carry data.

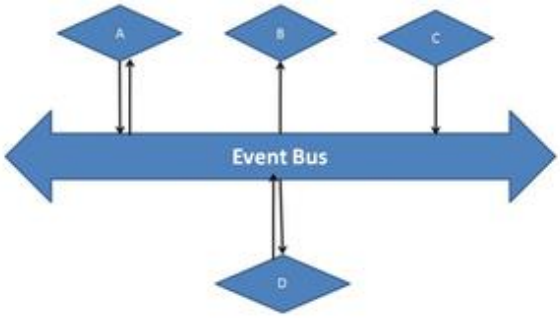


Figure 1.22 – Event Based Architecture

The event based architecture supports, several communication styles.

- Publisher-subscriber
- Broadcast
- Point-to-Point

System Level Architecture

- **Client Server Architecture**
  - The client server architecture has two major components. The client and the server as shown in figure 1.23..
  - If the node is requesting something, it can be known as a client, and if some node is providing something, it can be known as a server.
  - The Server is where all the processing, computing and data handling is happening, whereas the Client is where the user can access the services and resources given by the Server (Remote Server).
  - The clients can make requests from the Server, and the Server will respond accordingly.
  - Generally, there is only one server that handles the remote side.

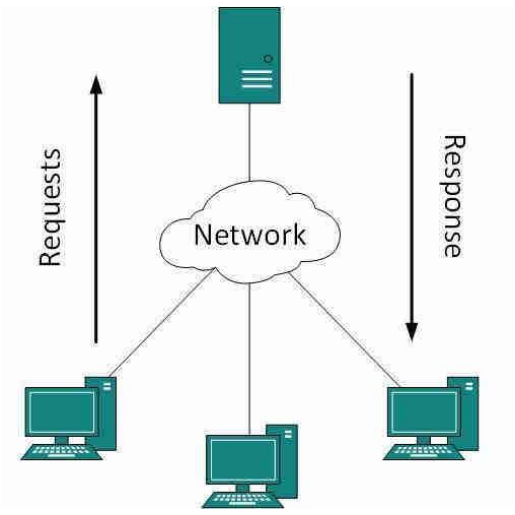


Figure 1.23 – Client Server Architecture

Advantages:

- Easier to Build and Maintain
- Better Security
- Stable

Disadvantages:

- Single point of failure
- Less scalable

What are Peer-to-Peer Network Families? Nov 2023

Peer to Peer (P2P)

- The general idea behind peer to peer is there is no central control in a distributed system.
- The basic idea is that, each node can either be a client or a server at a given time.
- In general, each node is referred to as a Peer as shown in figure 1.24.

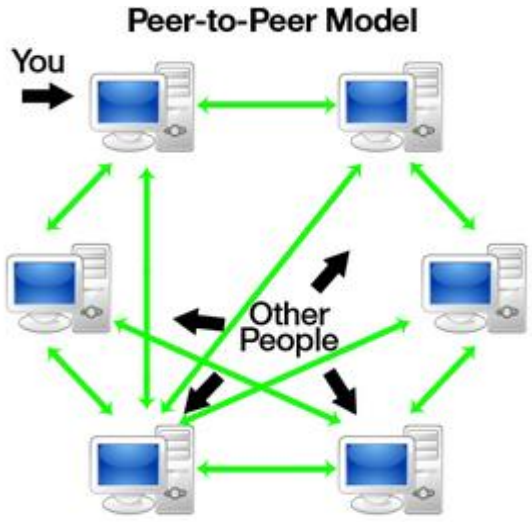


Figure 1.24 – Peer to Peer

9. Explain in details about System Models for Distributed and Cloud Computing.

- Distributed and cloud computing systems are built over a large number of autonomous computer nodes. These node machines are interconnected by SANs, LANs, or WANs in a hierarchical manner.
- Massive systems are classified into four groups:
    - clusters,
    - P2P networks,
    - computing grids,
    - Internet clouds over huge data centers.

➢ **Clusters of Cooperative Computers**

- A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.

Cluster Architecture

- Figure 1.25 defines a cluster of servers interconnected by a high-bandwidth SAN or LAN with shared I/O devices and disk arrays;
- The cluster acts as a single computer attached to the Internet.

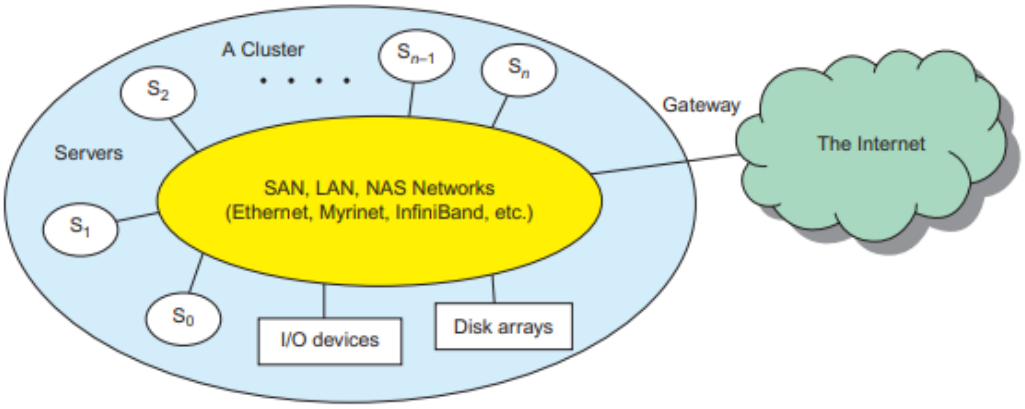


Figure 1.25 – A cluster of servers

- To build a larger cluster with more nodes, the interconnection network can be built with multiple levels of Gigabit Ethernet, Myrinet, or InfiniBand switches.
- The cluster is connected to the Internet via a virtual private network (VPN) gateway. The gateway IP address locates the cluster.
- The system image of a computer is decided by the way the OS manages the shared cluster resources.

Single-System Image

- An SSI is an illusion created by software or hardware that presents a collection of resources as one integrated, powerful resource.
- SSI makes the cluster appear like a single machine to the user.

Hardware, Software, and Middleware Support

- The building blocks are computer nodes (PCs, workstations, servers, or SMP), special communication software and a network interface card in each computer node.
- Special cluster middleware supports are needed to create SSI or high availability (HA).

Major Cluster Design Issues

- A cluster-wide OS for complete resource sharing is not available yet.
- Middleware or OS extensions were developed at the user space to achieve SSI at selected functional levels.
- Without this middleware, cluster nodes cannot work together effectively to achieve cooperative computing.
- The software environments and applications must rely on the middleware to achieve high performance.

➢ **Grid Computing Infrastructures**

- Internet services such as the Telnet command enables a local computer to connect to a remote computer.
- A web service such as HTTP enables remote access of remote web pages.
- Grid computing is envisioned to allow close interaction among applications running on distant computers simultaneously.

Computational Grids

- A computing grid offers an infrastructure that couples computers, software/middleware, special instruments, and people and sensors together.
- The grid is often constructed across LAN, WAN, or Internet backbone networks at a regional, national, or global scale.
- The computers used in a grid are primarily workstations, servers, clusters, and supercomputers. Personal computers, laptops, and PDAs can be used as access devices to a grid system.

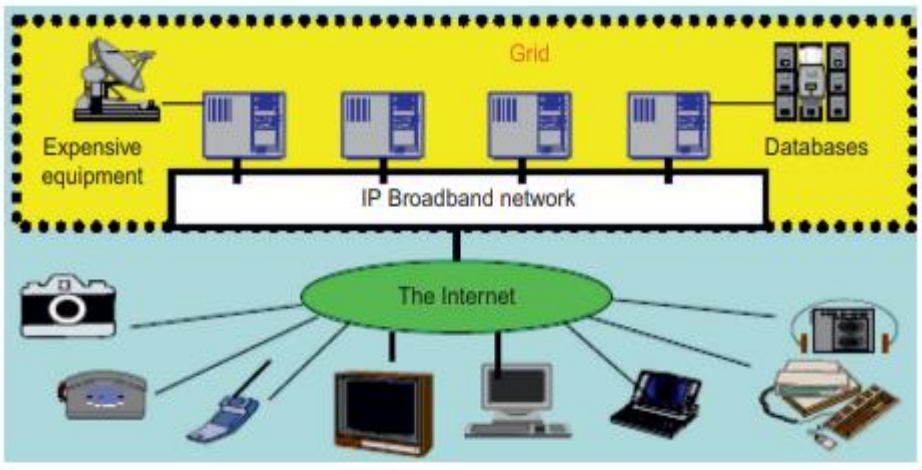


Figure 1.26 – Computational grid or data grid

.

- Figure 1.26 shows Computational grid or data grid providing computing utility, data, and information services through resource

sharing and cooperation among participating organizations.

- The grid is built across various IP broadband networks including LANs and WANs already used by enterprises or organizations over the Internet.
- The grid is presented to users as an integrated resource pool as shown in the upper half of the figure 1.26.
- The grid integrates the computing, communication, contents, and transactions as rented services.

Grid Families

- In Table 1.3, grid systems are classified into two categories: computational or data grids and P2P grids.
- Computing or data grids are built primarily at the national level.

Table 1.3 - Two Grid Computing Infrastructures and Representative Systems

Design Issues	Computational and Data Grids	P2P Grids
Grid Applications Reported	Distributed supercomputing, National Grid initiatives, etc.	Open grid with P2P flexibility, all resources from client machines
Representative Systems	TeraGrid built in US, ChinaGrid in China, and the e-Science grid built in UK	JXTA, FightAid@home, SETI@home
Development Lessons Learned	Restricted user groups, middleware bugs, protocols to acquire resources	Unreliable user-contributed resources, limited to a few apps

➢ **Peer-to-Peer Network Families**

- An example of a well-established distributed system is the client-server architecture where client machines are connected to a central server for compute, e-mail, file access, and database applications.
- The P2P architecture offers a distributed model of networked systems.
- A P2P network is client-oriented instead of server-oriented.
- P2P systems are introduced at the physical level and overlay networks at the logical level.

P2P Systems

- In a P2P system, every node acts as both a client and a server, providing part of the system resources.
- Peer machines are simply client computers connected to the Internet.
- This implies that no master-slave relationship exists among the peers.

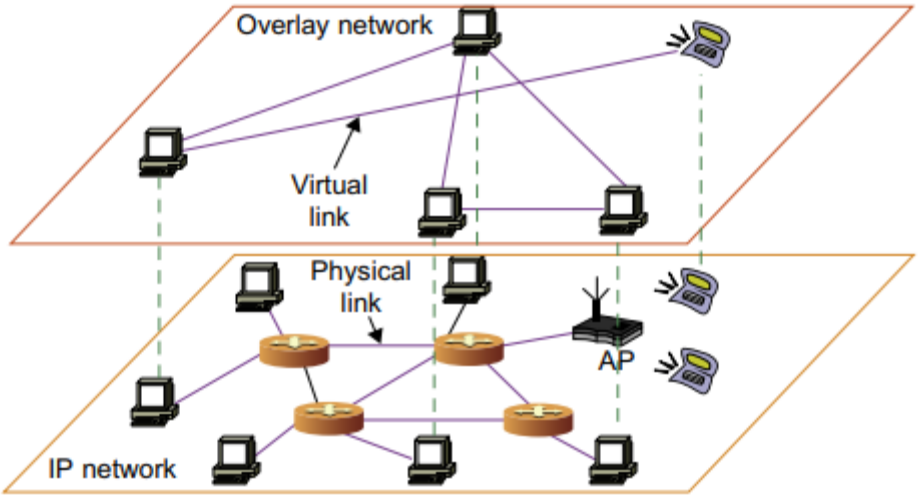


Figure 1.27– Computational grid or data grid

- Figure 1.27 shows the structure of a P2P system by mapping a physical IP network to an overlay network built with virtual links.
- Initially, the peers are totally unrelated.
- Each peer machine joins or leaves the P2P network voluntarily.
- Only the participating peers form the physical network at any time.
- Thus, the physical network varies in size and topology dynamically due to the free membership in the P2P network.

Overlay Networks

- Data items or files are distributed in the participating peers.
- Based on communication or file-sharing needs, the peer IDs form an overlay network at the logical level.
- This overlay is a virtual network formed by mapping each physical machine with its ID, logically, through a virtual mapping.
- When a new peer joins the system, its peer ID is added as a node in the overlay network.
- When an existing peer leaves the system, its peer ID is removed from the overlay network automatically.
- There are two types of overlay networks: unstructured and structured.
- An unstructured overlay network is characterized by a random graph. There is no fixed route to send messages or files among the nodes so flooding is used.
- Structured overlay networks follow certain connectivity topology and rules for inserting and removing nodes (peer IDs) from the overlay graph. Routing mechanisms are used to send messages or files among the nodes.

P2P Application Families

- Based on application, P2P networks are classified into four groups,

1. Distributed file sharing of digital contents on the P2P network.
2. Collaboration P2P networks.
3. Distributed P2P computing
4. Other P2P platforms,

P2P Computing Challenges

- P2P computing faces three types of heterogeneity problems in hardware, software, and network requirements.
  - There are too many hardware models and architectures to select from;
  - incompatibility exists between software and the OS;
  - different network connections and protocols make it too complex to apply in real applications.

➢ Cloud Computing over the Internet

- A cloud is a pool of virtualized computer resources.

Internet Clouds

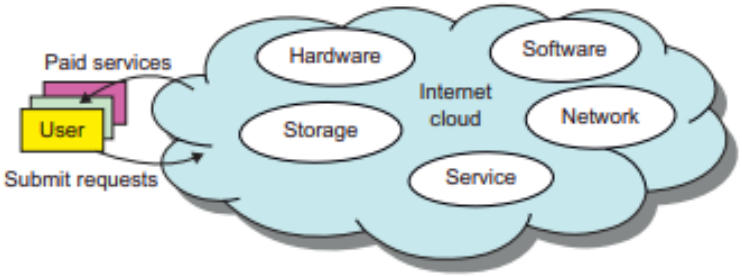


Figure 1.28 – Virtualized resources from data centers to form an Internet cloud

- Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically as in figure 1.28.
- Cloud computing leverages its low cost and simplicity to benefit both users and providers.
- The cloud ecosystem must be designed to be secure, trustworthy, and dependable.

Deployment models

- Public Cloud
- Private Cloud
- Hybrid Cloud
- Community Cloud

Characteristics of Cloud computing

- On-demand Self Service
- Broad Network Access
- Resource Pooling

- Rapid Elasticity
- Measured Service

Cloud service model

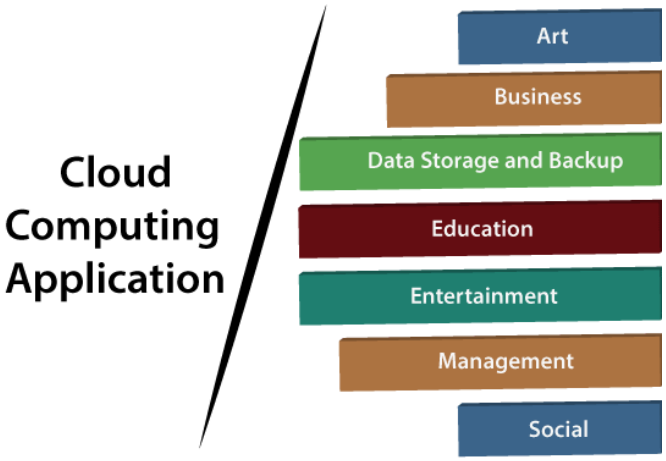
- Infrastructure as a Service - This model puts together infrastructures demanded by users—namely servers, storage, networks, and the data center fabric
- Platform as a Service - This model enables the user to deploy user-built applications onto a virtualized cloud platform.
- Software as a Service - This refers to browser-initiated application software over thousands of paid cloud customers.

Eight reasons to adapt the cloud for upgraded Internet applications and web services:

1. Desired location in areas with protected space and higher energy efficiency
2. Sharing of peak-load capacity among a large pool of users, improving overall utilization
3. Separation of infrastructure maintenance duties from domain-specific application development
4. Significant reduction in cloud computing cost, compared with traditional computing paradigms
5. Cloud computing programming and application development
6. Service and data discovery and content/service distribution
7. Privacy, security, copyright, and reliability issues
8. Service agreements, business models, and pricing policies

10. List some Cloud Computing Applications

- Cloud service providers provide various applications in the field of art, business, data storage and backup services, education, entertainment, management, social networking, etc.



1. Art Applications

Moo

- Moo is one of the best cloud art applications. It is used for designing and printing business cards, postcards, and mini cards.

Vistaprint

- Vistaprint allows us to easily design various printed marketing products such as business cards, Postcards, Booklets, and wedding invitations cards.

Adobe Creative Cloud

- Adobe creative cloud is made for designers, artists, filmmakers, and other creative professionals.

2. Business Applications

MailChimp

- MailChimp is an **email publishing platform** which provides various options to **design, send,** and **save** templates for emails.

Salesforce

- Salesforce platform provides tools for sales, service, marketing, e-commerce, and more. It also provides a cloud development platform.

Chatter

- Chatter helps us to **share important information** about the organization in real time.

Paypal

- Paypal offers the simplest and easiest **online payment** mode using a secure internet account.

3. Data Storage and Backup Applications

Box.com

- Box provides an online environment for **secure content management, workflow,** and **collaboration**. It allows us to store different files such as Excel, Word, PDF, and images on the cloud.

Mozy

- Mozy provides powerful **online backup solutions** for our personal and business data. It schedules automatically back up for each day at a specific time.

Google G Suite

- Google G Suite is one of the best **cloud storage** and **backup** application. It includes Google Calendar, Docs, Forms, Google+, Hangouts, as well as cloud storage and tools for managing cloud apps. The most popular app in the Google G Suite is Gmail. Gmail offers free email services to users.

4. Education Applications

Google Apps for Education

- Google Apps for Education is the most widely used platform for free web-based email, calendar, documents, and collaborative study.

Chrome books for Education

- Chrome book for Education is one of the most important Google's projects. It is designed for the purpose that it enhances education innovation.

Tablets with Google Play for Education

- It allows educators to quickly implement the latest technology solutions into the classroom and make it available to their students.

AWS in Education

- AWS cloud provides an education-friendly environment to universities, community colleges, and schools.

5. Entertainment Applications

Online game

- It offers various online games that run remotely from the cloud. The best cloud gaming services are Shaow, GeForce Now, Vortex, Project xCloud, and PlayStation Now.

Video Conferencing Apps

- Video conferencing apps provides a simple and instant connected experience. It allows us to communicate with our business partners, friends, and relatives using a cloud-based video conferencing.

6. Management Applications

Toggl

- Toggl helps users to track allocated time period for a particular project.

Evernote

- Evernote allows to sync and save your recorded notes, typed notes, and other notes in one convenient place. It is available for both free as well as a paid version.

GoToMeeting

- GoToMeeting provides **Video Conferencing** and **online meeting apps**, which allows to start a meeting with business partners from anytime, anywhere using mobile phones or tablets.

7. Social Applications

Facebook

- Facebook is a **social networking website** which allows active users to share files, photos, videos, status, more to their friends, relatives, and business partners using the cloud storage system.

Twitter

- Twitter is a **social networking** site.It allows users to follow high profile celebrities, friends, relatives, and receive news. It sends and receives short posts called tweets.

LinkedIn

- LinkedIn is a **social network** for students, freshers, and professionals.

MAILAM Engineering College

Approved by AICTE, New Delhi, Permanently Affiliated to Anna University
Chennai,Accredited by NBA, NAAC with A Grade and TCS)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

III YEAR / V SEM

CCS335 CLOUD COMPUTING

UNIT 2

VIRTUALIZATION BASICS

SYLLABUS: Virtual Machine Basics – Taxonomy of Virtual Machines – Hypervisor – Key Concepts – Virtualization structure – Implementation levels of virtualization – Virtualization Types: Full Virtualization – Para Virtualization – Hardware Virtualization – Virtualization of CPU, Memory and I/O devices.

PART A

1. What is the Hypervisor? Nov 2023

- The hypervisor is also known as the VMM (Virtual Machine Monitor).
- The hypervisor supports hardware-level virtualization on bare metal devices like CPU, memory, disk and network interfaces.
- The hypervisor software sits directly between the physical hardware and its OS.
- This virtualization layer is referred to as either the VMM or the hypervisor.
- The hypervisor provides hypercalls for the guest OSes and applications.

2. Write the Role of CPU Virtualization.

- Central processing unit (CPU) virtualization is the fundamental technology that makes hypervisors, virtual machines, and operating systems possible.
- It allows a single CPU to be divided into multiple virtual CPUs for use by multiple VMs.

3. What is meant by Virtual Machine?

- A virtual machine (VM) is a virtual representation, or emulation, of a physical computer. Virtual machine software can run programs and operating systems, store data, connect to networks, and do other

computing functions, and requires maintenance such as updates and system monitoring.

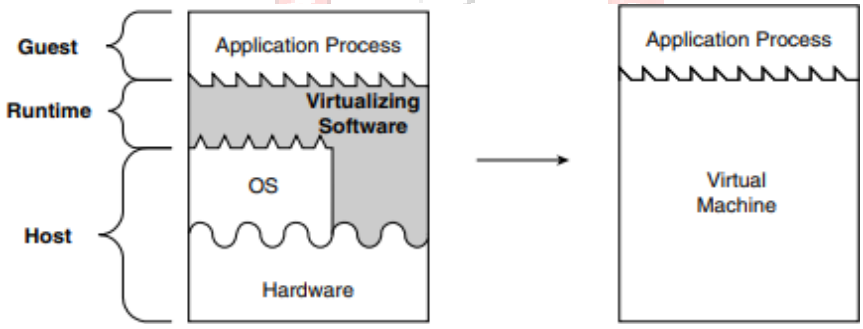
4. State the benefits of Virtual Machine.

- Cost savings
- Agility and speed
- Lowered downtime
- Scalability
- Security benefits

5. State Process-level virtual machine with a neat diagram.

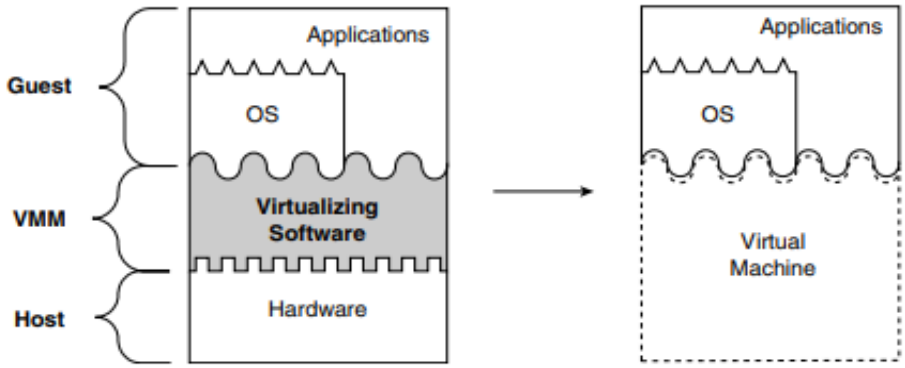
Process virtual machine

Process virtual machine is designed to run a single program, which means that it supports a single process.



6. State System-level virtual machine with a neat diagram.

- A system virtual machine provides a complete system platform which supports the execution of a complete operating system (OS)



7. What is meant by Hypervisor?

- The hypervisor is also known as the VMM (Virtual Machine Monitor).
- The hypervisor supports hardware-level virtualization on bare metal devices like CPU, memory, disk and network interfaces.

8. State the functionality of hypervisor.

- The hypervisor software sits directly between the physical hardware and its OS. The hypervisor provides hypercalls for the guest OSes and applications

9. State the classification of hypervisor architecture based on functionality.

- Micro-Kernel Architecture
- Monolithic Hypervisor Architecture

10. What is meant by Xen Architecture?

- Xen is an open source hypervisor program developed by Cambridge University. Xen is a microkernel hypervisor, which separates the policy from the mechanism.

11. What are the various implementation levels of Virtualization?

- Instruction Set Architecture Level
- Hardware Abstraction Level
- Operating System Level
- Library Support Level
- User-Application Level

12. What is meant by Hardware Virtualization?

- Hardware virtualization is the method used to create virtual versions of physical desktops and operating systems

13. What are the various types of hardware virtualization?

- Full virtualization
- Para virtualization
- Hardware-assisted virtualization

14. What are three ways to implement I/O virtualization?

- Full device emulation
- Para-virtualization
- Direct i/o

15. What are the different critical instruction in CPU Virtualization?

- Privileged instructions
- Control sensitive instructions
- Behavior-sensitive instructions

16. Define Full Virtualization.

- Fully simulates the hardware to enable a guest OS to run in an isolated instance. In a fully virtualized instance, an application would run on top of a guest OS, which would operate on top of the hypervisor and finally the host OS and hardware.

17. What is meant by Para virtualization?

- Runs a modified and recompiled version of the guest OSes in a VM. The hardware isn't necessarily simulated in para virtualization but uses an application program interface (API) that can modify guest OSes.

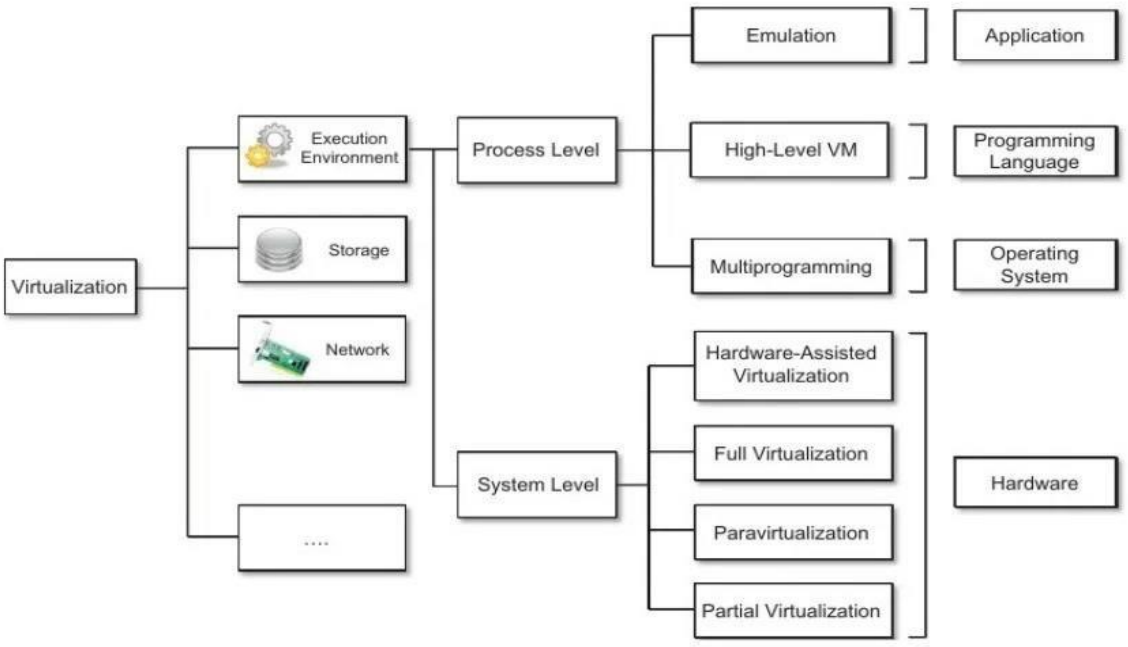
18. What is meant by Hardware-Assisted CPU Virtualization?

- Uses a computer's hardware as architectural support to build and manage a fully virtualized VM.

19. What are the uses of Virtual Machines?

- Testing
- Running software designed for other OSes
- Running outdated software
- Browser isolation

20. Illustrate the taxonomy of Virtual Machines.



21. What is meant by memory virtualization?

- Memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.
- A two-stage mapping process should be maintained by the guest OS and the VMM: virtual memory to physical memory and physical memory to machine memory

22. What is meant by Virtualization?

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.
- The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.

23. What is meant by Instruction Set Architecture Level virtualization?

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine

24. What is meant by Hardware Abstraction Level virtualization?

- Hardware-level virtualization is performed right on top of the bare hardware. The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices.

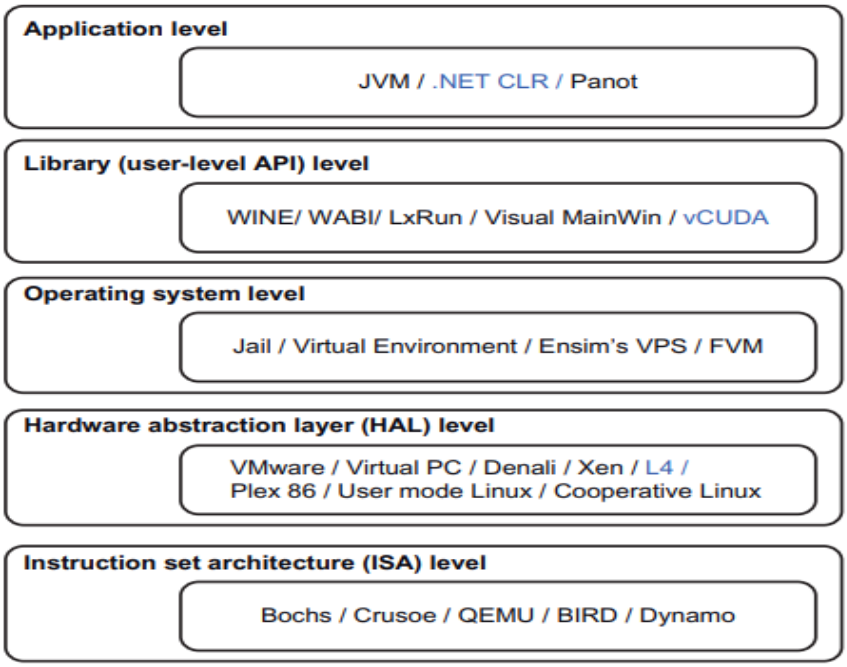
25. What are the advantages of host based virtualization?

- First, the user can install this VM architecture without modifying the host OS.
- Second, the host-based approach appeals to many host machine configurations.

26. What is meant by KVM (Kernel-Based VM)?

- KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants.

27. Draw the implementation levels of Virtualization with example.



Part-B

1. Explain in detail about Virtual Machine Basics.

- Virtual machines
- Uses of Virtual machines
- Benefits of Virtual machines
- Virtualization
- Types of Virtual Machine
- Virtual Machine Applications

➢ **Virtual machines**

- A virtual machine (VM) is a virtual representation, or emulation, of a physical computer.
- Virtual machine software can run programs and operating systems, store data, connect to networks, and do other computing functions, and requires maintenance such as updates and system monitoring.
- They are often referred to as a guest while the physical machine they run on is referred to as the host.

➢ **Uses of Virtual machines**

- **Testing** - Software developers often want to test their applications in different environments. They can use virtual machines to run their applications in various OSes on one computer.
- **Running software designed for other OSes** - A VM can run software designed for a different OS.
- **Running outdated software** - Users who want to run these applications can run an old OS on a virtual machine.
- **Browser isolation** - Browser isolation is the practice of 'isolating' web browser activity away from the rest of a computer's operating system to keep malware from affecting the computer's other files and programs.

➢ **Benefits of Virtual machines**

- Cost savings
- Agility and speed
- Lowered downtime
- Scalability
- Security benefits

- **Virtualization**
  - Virtualization is the process of creating a software-based, or "virtual" version of a computer, with dedicated amounts of CPU, memory, and storage that are "borrowed" from a physical host computer—such as personal computer— and/or a remote server—such as a server in a cloud provider's datacenter.
  - A VM cannot interact directly with a physical computer.
  - It needs a lightweight software layer called a hypervisor to coordinate between it and the underlying physical hardware.
  - The hypervisor allocates physical computing resources—such as processors, memory, and storage—to each VM.
  - It keeps each VM separate from others so they don't interfere with each other.
  - There are two primary types of hypervisors.
    - Type 1 hypervisors run directly on the physical hardware, taking the place of the OS.
    - Type 2 hypervisors run as an application within a host.
  - The process of virtualization consists of two parts:
    (1) the mapping of virtual resources or state, e.g., registers, memory, or files, to real resources in the underlying machine
    (2) the use of real machine instructions and/or system calls to carry out the actions specified by virtual machine instructions and/or system calls
- **Types of Virtual Machine**
  - Process-level virtual machines
  - System-level virtual machines

Process-level virtual machines / A Process Virtual Machine.

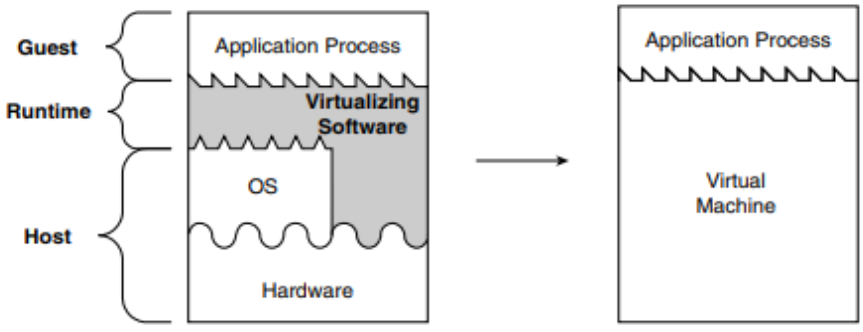


Figure 2.1 – Process level virtual machine

- Virtualizing software translates a set of OS and user-level instructions composing one platform to another, forming a process virtual machine capable of executing programs developed for a different OS and a different ISA. Refer figure 2.1.

System-level virtual machines / A System Virtual Machine

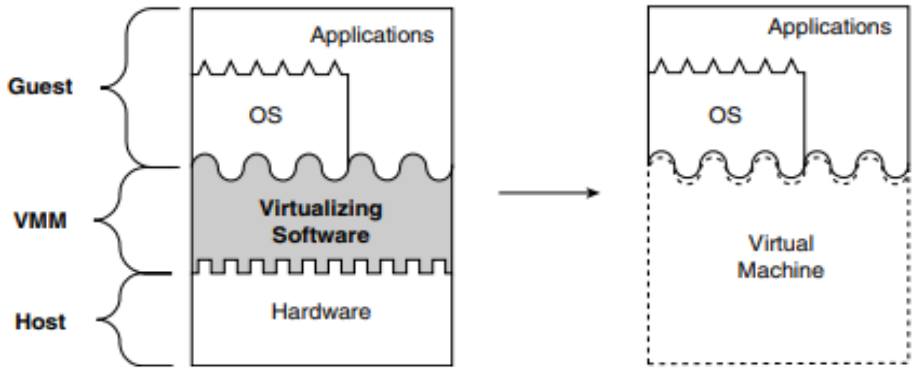
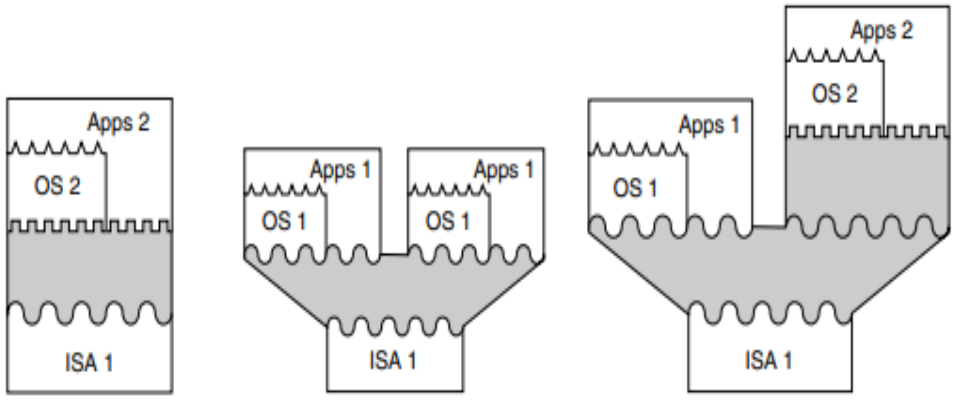


Figure 2.2 – System level virtual machine

Virtualizing software translates the ISA used by one hardware platform to another, forming a system virtual machine, capable of executing a system software environment developed for a different set of hardware. Refer Figure 2.2

➢ **Virtual Machine Applications**



A *B* *C*

Figure 2.3 – Examples of Virtual Machine Applications

(A) Emulating one instruction set with another;

(B) replicating a virtual machine so that multiple operating systems can be supported simultaneously;

(C) composing virtual machine software to form a more complex, flexible system.

Emulating one instruction set with another;

- Replicating a virtual machine so that multiple operating systems can be supported simultaneously;
- Composing virtual machine software to form a more complex, flexible system.

2. Explain in detail about Taxonomy of Virtual Machines.

Depict the Taxonomy of Virtual Machines and narrate steps for launching a server in AWS cloud platform. Nov 2023

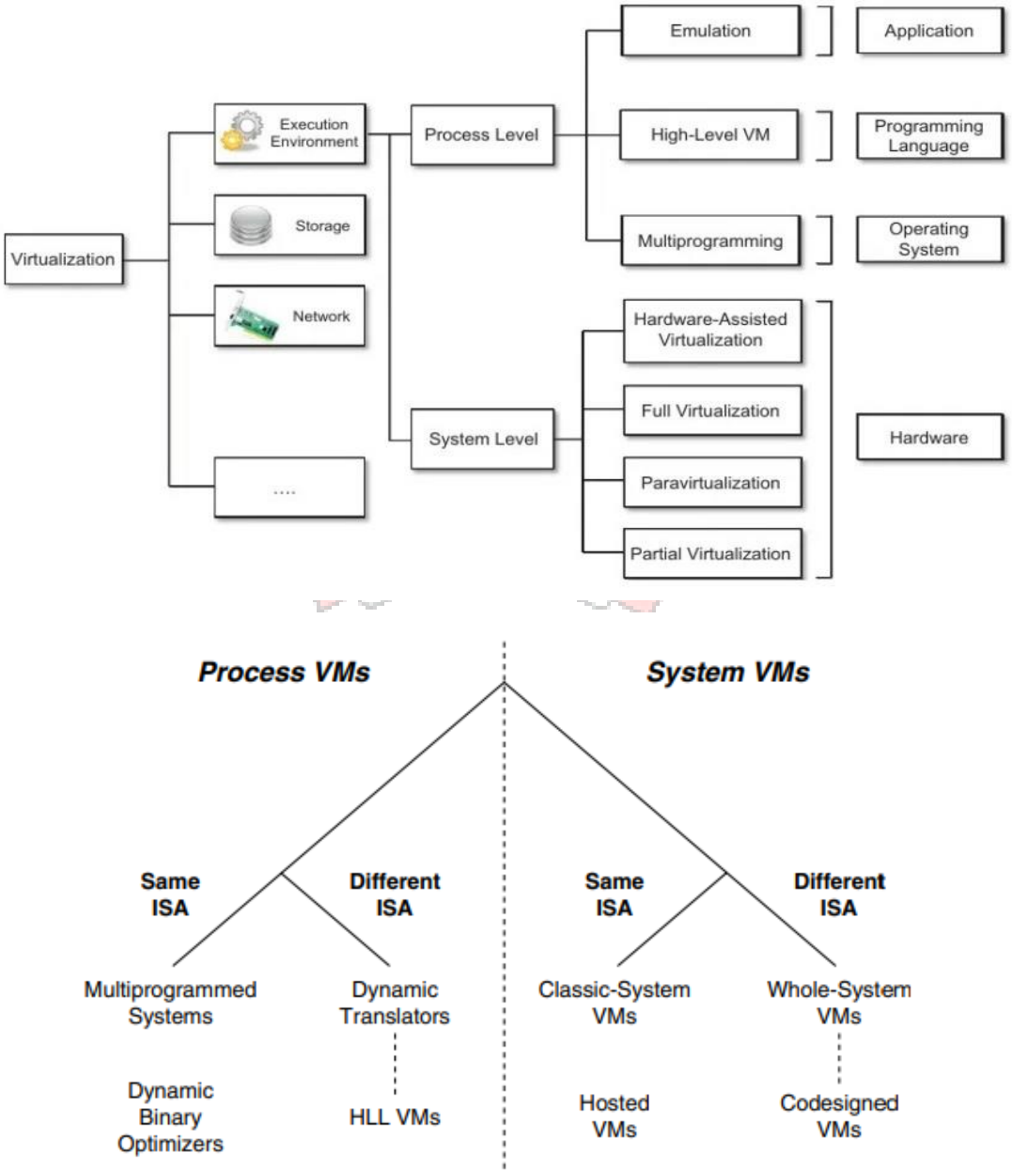


Figure 2.4 – Taxonomy of Virtual Machines

- First, VMs are divided into the two major types: process VMs and system VMs.
- In process, the VM supports an ABI — user instructions plus system calls; in the system, the VM supports a complete ISA — both user and system instructions.
- Finer divisions in the taxonomy are based on whether the guest and host use the same ISA.
- On the left-hand side of Figure 2.4 are process VMs.
  - These include VMs where the host and guest instruction sets are the same.
  - Examples.
    - Multi programmed systems,
    - dynamic binary optimizers,
  - For process VMs where the guest and host ISAs are different, are two examples.
    - dynamic translators
    - HLL VMs.
- On the right-hand side of the figure are system VMs.
- If the guest and host use the same ISA, examples include
  - "classic" system VMs
  - hosted VMs.
- Examples of system VMs where the guest and host ISAs are different includes
  - whole-system VMs
  - codesigned VMs.
- Codesigned VMs are connected using dotted lines because their interface is typically at a lower level than other system VMs.
- HLL VM's are connected to the VM taxonomy via a "dotted line" because their process-level interface is at a different, higher level than the other process VMs.

3. Explain in detail about Hypervisor - Xen Architecture and Virtualization Structure.

- Introduction
- Hypervisor
- The Xen Architecture

➢ **Introduction**

- Before virtualization, the operating system manages the hardware.
- After virtualization, a virtualization layer is inserted between the hardware and the operating system. The virtualization layer is responsible for converting portions of the real hardware into virtual hardware.
- Therefore, different operating systems such as Linux and Windows can run on the same physical machine, simultaneously.
- Depending on the position of the virtualization layer, there are several classes of VM architectures, namely the
  - hypervisor architecture,
  - para virtualization,
  - host-based virtualization.

➢ **Hypervisor**

- The hypervisor is also known as the VMM (Virtual Machine Monitor).
- The hypervisor supports hardware-level virtualization on bare metal devices like CPU, memory, disk and network interfaces.
- The hypervisor software sits directly between the physical hardware and its OS.
- This virtualization layer is referred to as either the VMM or the hypervisor.
- The hypervisor provides hypercalls for the guest OSes and applications.
- Depending on the functionality, a hypervisor can be
  - a micro-kernel architecture
  - a monolithic hypervisor architecture

- A micro-kernel hypervisor includes only the basic and unchanging functions (such as physical memory management and processor scheduling).
- A monolithic hypervisor implements the device drivers and other changeable components.
- Therefore, the size of the hypervisor code of a micro-kernel hypervisor is smaller than that of a monolithic hypervisor.
- Essentially, a hypervisor must be able to convert physical devices into virtual resources dedicated for the deployed VM to use.

➢ **The Xen Architecture**

- Xen is an open source hypervisor program developed by Cambridge University.
- Xen is a microkernel hypervisor, which separates the policy from the mechanism.

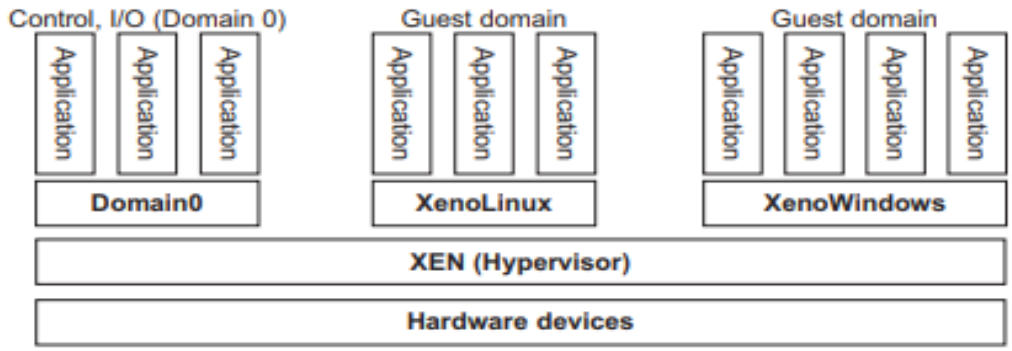


Figure 2.5 – Xen Hypervisor

- The Xen hypervisor implements all the mechanisms, leaving the policy to be handled by Domain 0, as shown in Figure 2.5.
- Xen does not include any device drivers, It just provides a mechanism by which a guest OS can have direct access to the physical devices.
- As a result, the size of the Xen hypervisor is kept rather small.
- Xen provides a virtual environment located between the hardware and the OS.
- A number of vendors are in the process of developing commercial Xen hypervisors, among them are Citrix XenServer and Oracle VM.

- The core components of a Xen system are the hypervisor, kernel, and applications.
- The guest OS, which has control ability, is called Domain 0, and the others are called Domain U.
- Domain 0 is a privileged guest OS of Xen. It is first loaded when Xen boots without any file system drivers being available.
- Domain 0 is designed to access hardware directly and manage devices.
- Therefore, one of the responsibilities of Domain 0 is to allocate and map hardware resources for the guest domains (the Domain U domains).
- For example, Xen is based on Linux. Its management VM is named Domain 0, which has the privilege to manage other VMs implemented on the same host.
- If Domain 0 is compromised, the hacker can control the entire system. So, in the VM system, security policies are needed to improve the security of Domain 0.

4. Explain in detail about Implementation Levels of Virtualization.

Nov 2023

- Virtualization
- Implementation Levels of Virtualization
  - Instruction Set Architecture Level
  - Hardware Abstraction Level
  - Operating System Level
  - Library Support Level
  - User-Application Level

➢ **Virtualization**

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.
- The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.
- Hardware resources (CPU, memory, I/O devices, etc.) or software resources (operating system and software libraries) can be virtualized in various functional layers.

- A traditional computer runs with a host operating system specially tailored for its hardware architecture as shown in figure 2.6a.
- After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS.
- This is often done by adding additional software, called a virtualization layer as shown in Figure 2.6(b).

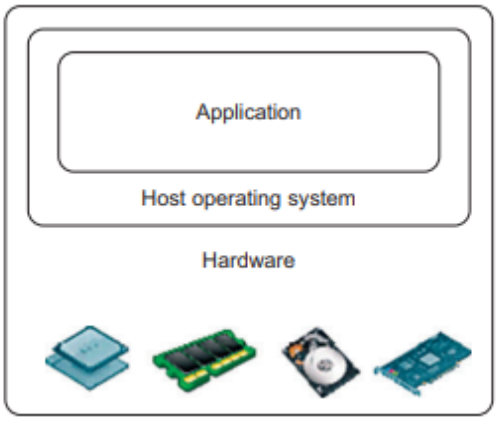


Figure 2.6 (a) – Before Virtualization

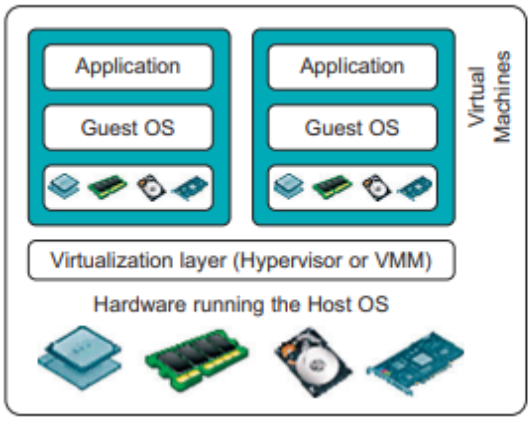


Figure 2.6 (b) – Before Virtualization

- This virtualization layer is known as hypervisor or virtual machine monitor (VMM).
- The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs, exclusively.
- The virtualization software creates the abstraction of VMs by interposing a virtualization layer at various levels of a computer system.
- As shown in Figure 2.7, Common virtualization layers include the instruction set architecture (ISA) level, hardware level, operating system level, library support level, and application level.

➢ **Implementation Levels of Virtualization**

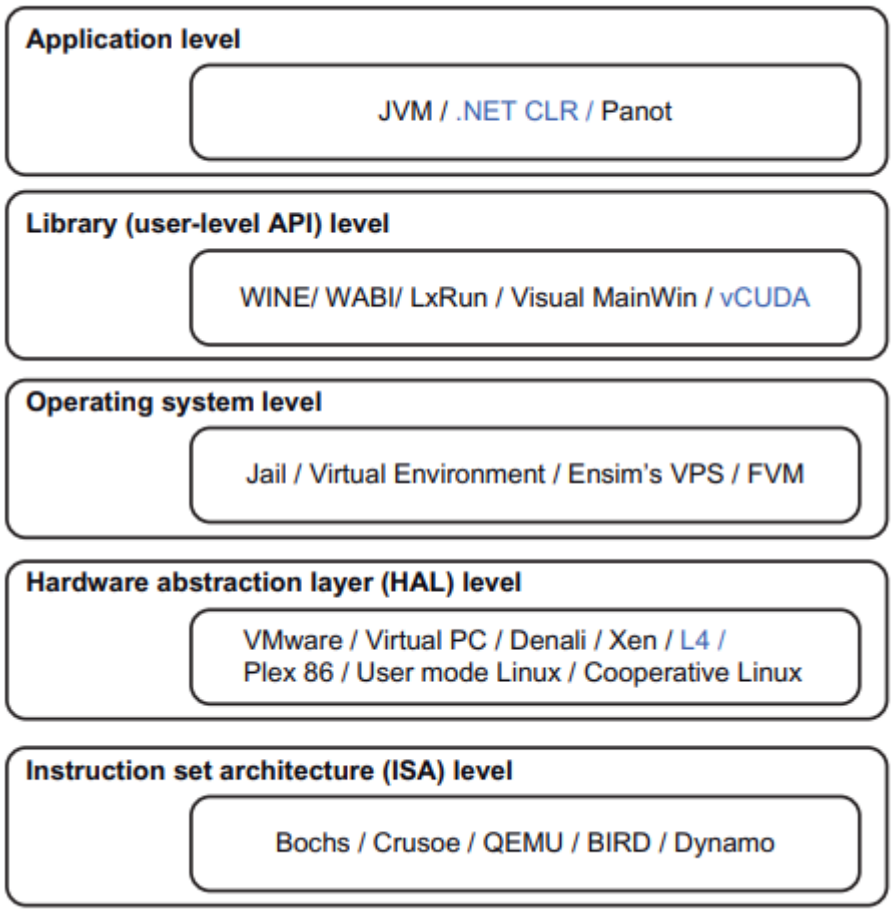


Figure 2.7 – Implementation Levels of Virtualization

Instruction Set Architecture Level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine.
- For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.
- With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.
- The basic emulation method is through code interpretation.
- An interpreter program interprets the source instructions to target instructions one by one.
- This process is relatively slow.
- For better performance, dynamic binary translation is desired.

- This approach translates basic blocks of dynamic source instructions to target instructions.
- The basic blocks can also be extended to program traces or super blocks to increase translation efficiency.
- Instruction set emulation requires binary translation and optimization.

Hardware Abstraction Level

- Hardware-level virtualization is performed right on top of the bare hardware.
- On the one hand, this approach generates a virtual hardware environment for a VM.
- On the other hand, the process manages the underlying hardware through virtualization.
- The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices.
- The intention is to upgrade the hardware utilization rate by multiple users concurrently.

Operating System Level

- This refers to an abstraction layer between traditional OS and user applications.
- OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers. The containers behave like real servers.
- OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users.

Library Support Level

- Most applications use APIs exported by user-level libraries.
- Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks.
- The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts.

User-Application Level

- Virtualization at the application level virtualizes an application as a VM.

- Therefore, application-level virtualization is also known as process-level virtualization.
- The most popular approach is to deploy high level language (HLL) VMs.
- The virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.
- Any program written in the HLL and compiled for this VM will be able to run on it.
- The Microsoft .NET CLR and Java Virtual Machine (JVM) are two good examples of this class of VM.

5. Discuss about Virtualization types, full virtualization and binary translation with full virtualization in detail.

Hardware virtualization can be classified into two categories:

- full virtualization
- host-based virtualization.

➢ **Full virtualization**

- Full virtualization does not need to modify the host OS.
- Full virtualization allows multiple guest operating systems to execute on a host operating system independently
- It relies on binary translation to virtualize the execution of certain instructions.
- With full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software.
- Both the hypervisor and VMM approaches are considered full virtualization.
- Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do.
- Therefore, running noncritical instructions on hardware not only can promote efficiency, but also can ensure system security.

Binary Translation of Guest OS Requests Using a VMM

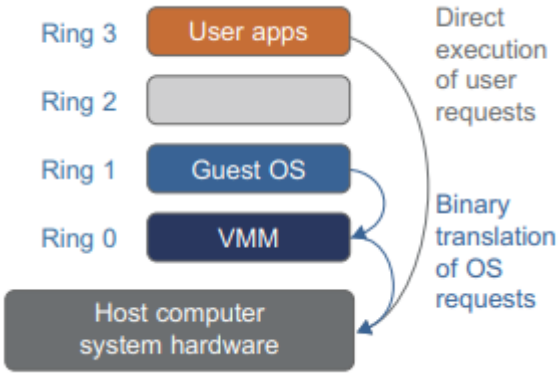


Figure 2.8 – Binary translation of Guest OS

- As shown in Figure 2.8, VMware puts the VMM at Ring 0 and the guest OS at Ring 1.
- The VMM scans the instruction stream and identifies the privileged, control-and behavior-sensitive instructions.
- When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions.
- The method used in this emulation is called binary translation. Therefore, full virtualization combines binary translation and direct execution.
- The guest OS is completely decoupled from the underlying hardware.
- The performance of full virtualization may not be ideal, because it involves binary translation which is rather time-consuming.
- Binary translation employs a code cache to store translated hot instructions to improve performance, but it increases the cost of memory usage.
- ESXi, VMWare, and Microsoft virtual servers are the technologies that provide full virtualization capabilities.

➢ **Host-Based Virtualization**

- An alternative VM architecture is to install a virtualization layer on top of the host OS.
- This host OS is still responsible for managing the hardware.
- The guest OSes are installed and run on top of the virtualization layer.

- Dedicated applications may run on the VMs. Some other applications can also run with the host OS directly.
- This host based architecture has advantages,
  - First, the user can install this VM architecture without modifying the host OS.
  - Second, the host-based approach appeals to many host machine configurations.
- Compared to the hypervisor/VMM architecture, the performance of the host-based architecture may also be low.

6. Discuss about in detail about para virtualization.

- Para Virtualization
- Para-Virtualization Architecture
- KVM (Kernel-Based VM)
- Para-Virtualization with Compiler Support

Para Virtualization

- Para-virtualization modifies the guest operating systems.
- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications.
- Performance degradation is a critical issue of a virtualized system.
- Para-virtualization attempts to reduce the virtualization overhead, and thus improve performance by modifying only the guest OS kernel.

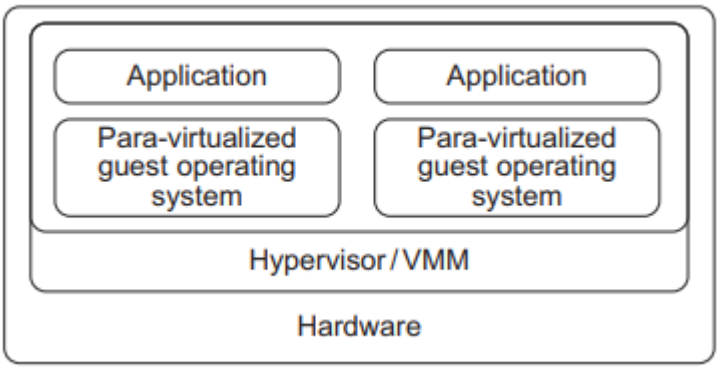


Figure 2.9 – Para Virtualized VM Architecture

- Figure 2.9 illustrates the concept of a para-virtualized VM architecture.

- The guest operating systems are para-virtualized. They are assisted by an intelligent compiler to replace the non virtualizable OS instructions by hypercalls as illustrated in Figure 2.10

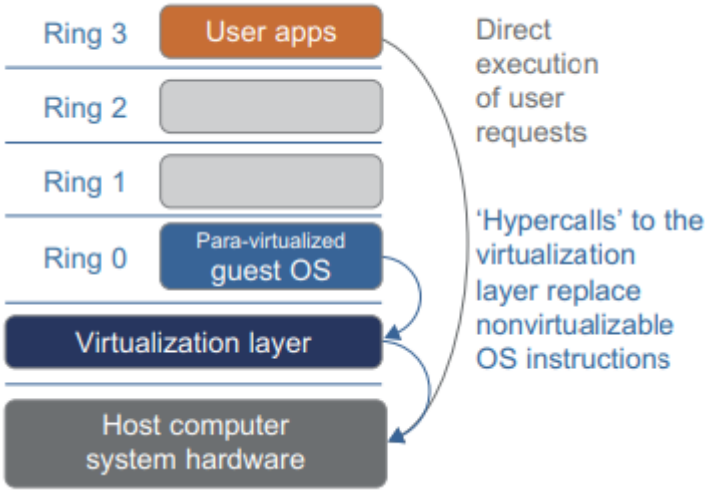


Figure 2.10 – Para Virtualized Guest OS

- The traditional x86 processor offers four instruction execution rings: Rings 0, 1, 2, and 3.
- The lower the ring number, the higher the privilege of instruction being executed.
- The OS is responsible for managing the hardware and the privileged instructions to execute at Ring 0, while user-level applications run at Ring 3.

Para-Virtualization Architecture

- When the x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS.
- According to the x86 ring definition, the virtualization layer should also be installed at Ring 0.
- However, when the guest OS kernel is modified for virtualization, it can no longer run on the hardware directly.
- Although para-virtualization reduces the overhead, it has incurred other problems.
    - Compatibility and portability issue
    - the cost of maintaining para-virtualized OSes is high
- Compared with full virtualization, para-virtualization is relatively easy and more practical.

- The popular Xen, KVM, and VMware ESX are good examples.

KVM (Kernel-Based VM)

- This is a Linux para-virtualization system—a part of the Linux version 2.6.20 kernel.
- Memory management and scheduling activities are carried out by the existing Linux kernel.
- KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants.

Para-Virtualization with Compiler Support

- Unlike the full virtualization architecture which intercepts and emulates privileged and sensitive instructions at runtime, para-virtualization handles these instructions at compile time.
- The guest OS kernel is modified to replace the privileged and sensitive instructions with hypercalls to the hypervisor or VMM.
- Xen assumes such a para-virtualization architecture.
- The guest OS running in a guest domain may run at Ring 1 instead of at Ring 0.
- This implies that the guest OS may not be able to execute some privileged and sensitive instructions.
- The privileged instructions are implemented by hypercalls to the hypervisor.
- After replacing the instructions with hypercalls, the modified guest OS emulates the behavior of the original guest OS.
- On an UNIX system, a system call involves an interrupt or service routine. The hypercalls apply a dedicated service routine in Xen.

7. Discuss in detail about hardware virtualization and virtualization of CPU, memory, and i/o devices.

- Hardware virtualization
- Types of hardware virtualization
  - full virtualization,
  - para virtualization
  - hardware-assisted virtualization.
- Virtualization of CPU, memory, and I/O devices
  - CPU Virtualization
    - Hardware-Assisted CPU Virtualization
  - Memory Virtualization
  - I/O Virtualization
    - full device emulation,
    - para-virtualization
    - direct I/O.

➢ **Hardware virtualization**

- Hardware virtualization is the method used to create virtual versions of physical desktops and operating systems.
- It uses a virtual machine manager (VMM) called a hypervisor to provide abstracted hardware to multiple guest operating systems, which can then share the physical hardware resources more efficiently.
- Hardware virtualization installs a hypervisor or virtual machine manager (VMM), which creates an abstraction layer between the software and the underlying hardware.
- Once a hypervisor is in place, software relies on virtual representations of the computing components, such as virtual processors rather than physical processors.
- Popular hypervisors include VMware's vSphere, based on ESXi, and Microsoft's Hyper-V.
- Since a hypervisor or VMM is installed directly on computing hardware and other OSes and applications are installed later, hardware virtualization is often referred to as bare-metal virtualization.

- **Types of hardware virtualization**
    - full virtualization,
    - para virtualization
    - hardware-assisted virtualization.
- **Full virtualization:** Fully simulates the hardware to enable a guest OS to run in an isolated instance. In a fully virtualized instance, an application would run on top of a guest OS, which would operate on top of the hypervisor and finally the host OS and hardware.
- **Para virtualization:** Runs a modified and recompiled version of the guest OSes in a VM. The hardware isn't necessarily simulated in para virtualization but uses an application program interface (API) that can modify guest OSes.
- **Hardware-assisted virtualization**: Uses a computer's hardware as architectural support to build and manage a fully virtualized VM.

- **Virtualization of CPU, memory, and I/O devices**
- **CPU Virtualization**
    - The critical instructions are divided into three categories:
        - privileged instructions,
        - control sensitive instructions,
        - behavior-sensitive instructions.
    - Privileged instructions execute in a privileged mode and will be trapped if executed outside this mode.
    - Control-sensitive instructions attempt to change the configuration of resources used.
    - Behavior-sensitive instructions have different behaviors depending on the configuration of resources.
    - A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode.
    - When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM.
    - In this case, the VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system.

- However, not all CPU architectures are virtualizable.
- RISC CPU architectures can be naturally virtualized because all control- and behavior-sensitive instructions are privileged instructions.
- On the contrary, x86 CPU architectures are not primarily designed to support virtualization. This is because about 10 sensitive instructions, are not privileged instructions. When these instructions execute in virtualization, they cannot be trapped in the VMM.
- Although para virtualization of a CPU lets unmodified applications run in the VM, it causes a small performance penalty.

Hardware-Assisted CPU Virtualization

- This technique attempts to simplify virtualization because full or para virtualization is complicated as shown in figure 2.11.
- Intel and AMD add an additional mode called privilege mode level to x86 processors.
- All the privileged and sensitive instructions are trapped in the hypervisor automatically.
- This technique removes the difficulty of implementing binary translation of full virtualization.

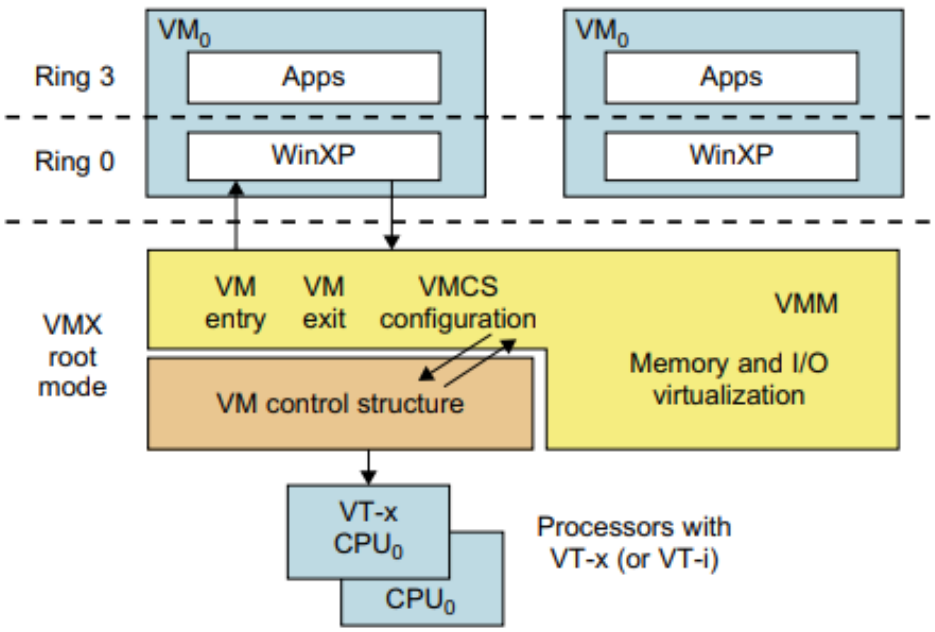


Figure 2.11 – Hardware-Assisted CPU Virtualization

➢ **Memory Virtualization**

- In a virtual execution environment, memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.
- A two-stage mapping process should be maintained by the guest OS and the VMM: virtual memory to physical memory and physical memory to machine memory.
- The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs.
- But the guest OS cannot directly access the actual machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory.
- Figure 2.12 shows the two-level memory mapping procedure.

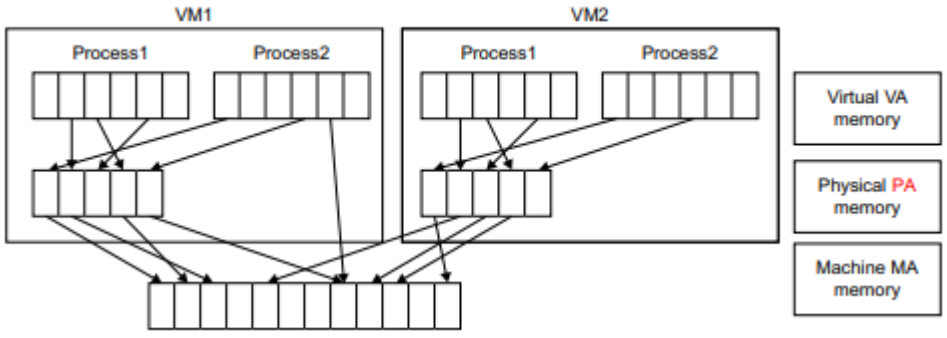


Figure 2.12 – Two-level memory mapping procedure.

- Each page table of the guest OSes has a separate page table in the VMM called as the shadow page table as shown in figure 2.13.
- VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation.
- When the guest OS changes the virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup.

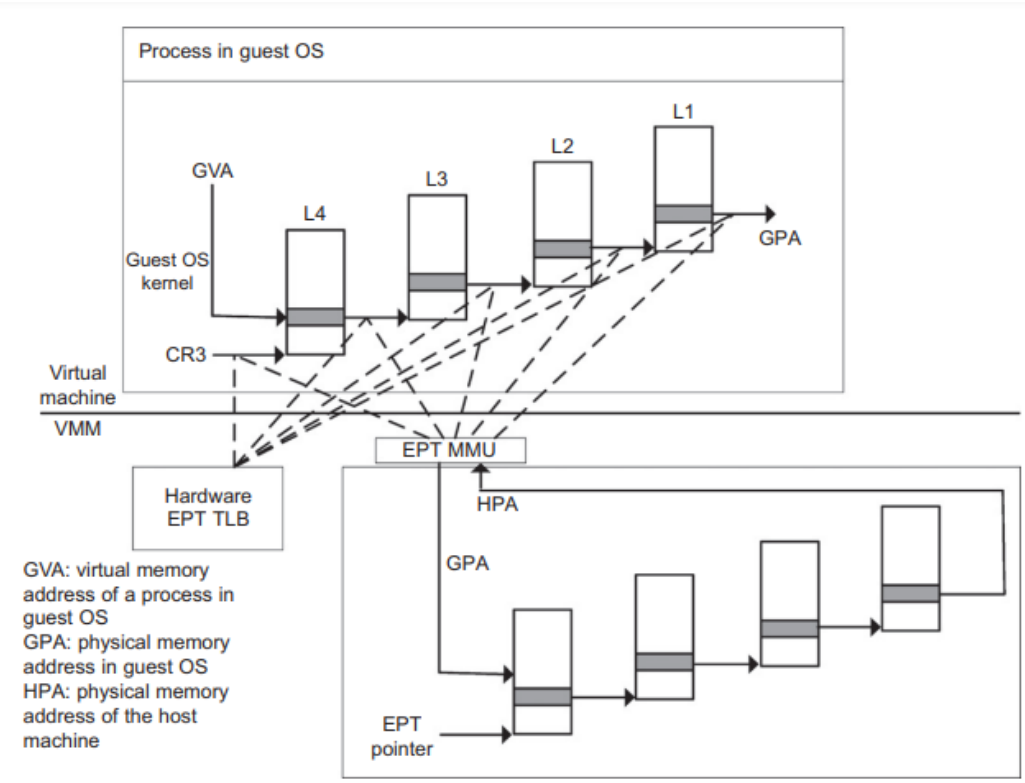


Figure 2.13 – Memory Virtualization using Shadow Page Table.

- **I/O Virtualization**
  - I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.
  - There are three ways to implement I/O virtualization:
    - full device emulation,
    - para-virtualization,
    - direct I/O.
  - **Full device emulation**
    - All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software.
    - This software is located in the VMM and acts as a virtual device.
    - The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.
    - The full device emulation approach is shown in Figure 2.13.

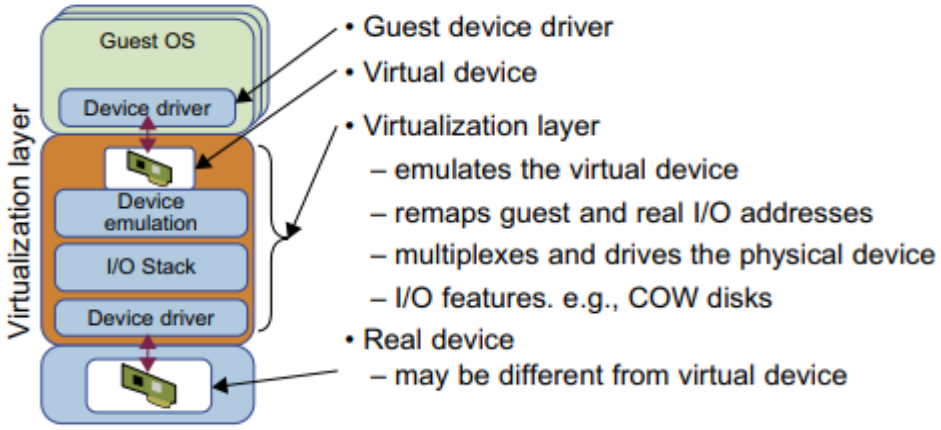


Figure 2.13 – Memory Virtualization using Shadow Page Table.

The para-virtualization method

- It is also known as the split driver model consisting of a frontend driver and a backend driver.
- The frontend driver is running in Domain U and the backend driver is running in Domain 0.
- They interact with each other via a block of shared memory.
- The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs.
- Although para-I/O-virtualization achieves better device performance than full device emulation, it comes with a higher CPU overhead.

Direct I/O virtualization

- It lets the VM access devices directly.
- It can achieve close-to-native performance without high CPU costs.
- Current direct I/O virtualization implementations focus on networking for mainframes.

8. Compare the terms Full Virtualization and Para Virtualization and depict the process of virtualizing CPU, memory, I/O devices. Nov 2023

Full Virtualization: Full Virtualization was introduced by IBM in the year 1966. It is the first software solution for server virtualization and uses binary translation and direct approach techniques. In full virtualization, guest OS is completely

isolated by the virtual machine from the virtualization layer and hardware. Microsoft and Parallels systems are examples of full virtualization.

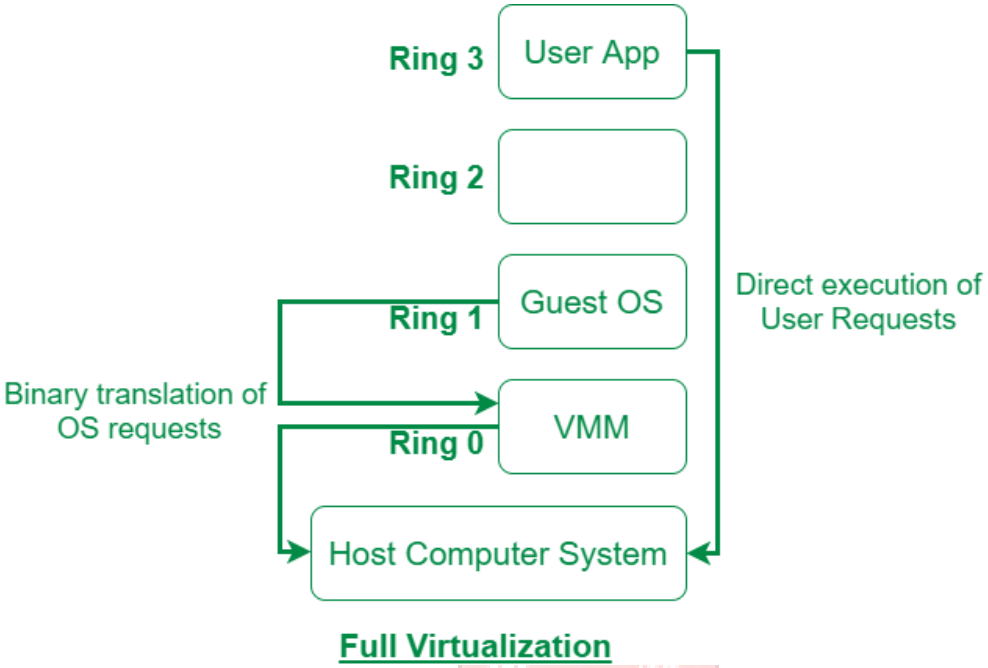


Fig. 2.14 Full Virtualization

Paravirtualization: Paravirtualization is the category of CPU virtualization which uses hypercalls for operations to handle instructions at compile time. In paravirtualization, guest OS is not completely isolated but it is partially isolated by the virtual machine from the virtualization layer and hardware. VMware and Xen are some examples of paravirtualization.

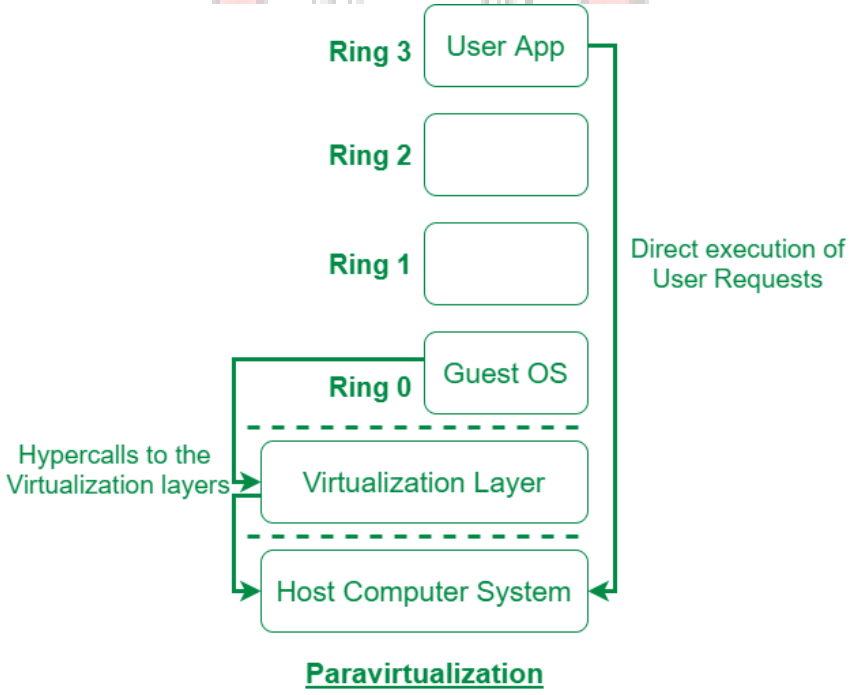


Fig.2.15 Para Virtualization

MAILAM Engineering College

Approved by AICTE, New Delhi, Permanently Affiliated to Anna University Chennai,Accredited by NBA, NAAC with A Grade and TCS)

DEPARTMENT OF

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

III YEAR / V SEM

CCS335 CLOUD COMPUTING

UNIT 3

VIRTUALIZATION INFRASTRUCTURE AND DOCKER

SYLLABUS: Desktop Virtualization – Network Virtualization – Storage Virtualization – System-level of Operating Virtualization – Application Virtualization – Virtual clusters and Resource Management – Containers vs. Virtual Machines – Introduction to Docker – Docker Components – Docker Container – Docker Images and Repositories.

PART A

1. What are the benefits of Network Virtualization? Nov 2023

- Reduce network provisioning time from weeks to minutes.
- Achieve greater operational efficiency by automating manual processes.
- Place and move workloads independently of physical topology.
- Improve network security within the data center.

2. What is a Docker in Cloud Computing? Nov 2023

- Docker is a software platform that allows you to build, test, and deploy applications quickly.
- Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.

3. What is meant by Desktop Virtualization?

- The desktop virtualization is a technology that separates an individual's PC applications from their desktop and provides a way for users to maintain their individual desktops on a single, central server that can be connected to from a LAN, WAN, or simply over the Internet.

4. What are the types of desktop Virtualization?

- Virtual Desktop Infrastructure(VDI)

- Remote Desktop Services
- Desktop-as-a-Service (DaaS)

5. State the benefits of Desktop Virtualization.

- Better security and control
- Ease of maintenance
- Remote work
- Resource management
- Reduced costs
- Increased employee productivity
- Improved flexibility

6. What is meant by network virtualization?

- Network Virtualization is a process of logically grouping physical networks and making them operate as single or multiple independent networks called Virtual Networks.

7. What are the functions of network virtualization?

- It enables the functional grouping of nodes in a virtual network.
- It enables the virtual network to share network resources.
- It allows communication between nodes in a virtual network without routing of frames.

8. State the Applications of Network Virtualization.

- Network virtualization may be used in the development of application testing to mimic real-world hardware and system software
- Network virtualization allows the simulation of connections between applications, services, dependencies, and end-users for software testing.

9. What is meant by storage virtualization?

- Storage virtualization in Cloud Computing is the sharing of physical storage into multiple storage devices which appears to be a single storage device.

10. What are the various types of Storage Virtualization?

- File Level Virtual Storage
- Object Level Virtual Storage
- Block Level Virtual Storage

11. State the pros and cons of Storage Virtualization.

Advantages/Pros

- Better Storage Utilization
- Easier Data Management

- Enables Addition of Advanced Features

Disadvantages/Cons

- Increased Traffic in the Storage Area Network
- Shared Environment may Lead to Data Compromise

12. What are the examples of Storage Virtualization?

- Logical Unit Number (LUN). LUN is a unique number used to identify a logical unit for computer storage.
- RAID groups. RAID (Redundant Array of Independent Disks) is a storage technology that features a configuration of multiple hard drives to work as a single computer system.
- Logical Volume (LV). A Logical Volume is a practice of combining multiple disk partitions or hard drives into a single volume group (VG).

13. What are the benefits of Application virtualization?

- Simplified management
- Increased Scalability
- Enhanced Security
- Allows the running of legacy apps.
- Enables cross-platform operations

14. What are the various steps in live migration of a VM?

- Step1 : Start migration
- Step2 : Transfer memory
- Step 3 : Suspend the VM and copy the last portion of the data
- Step4,5: Commit and activate the new host

15. Differentiate between Virtual machines and Containers.

Virtual Machines(VM)	Containers
VM is piece of software that allows to install other software inside of it.	While a container is a software that allows different functionalities of an application independently.
Examples: KVM, Xen, VMware.	Example: RancherOS, PhotonOS, Containers by Docker.

16. What are the various Docker Components?

- The Docker client and server
- Docker Images
- Registries
- Docker Containers

17. State the Docker Components.

- The Docker client and server
- Docker Images
- Registries
- Docker Containers

18. What are the two ways to create a Docker Image?

- docker commit command
- docker build command with a Dockerfile

19. State the execution instructions of Docker.

- Docker runs a container from the image.
- An instruction executes and makes a change to the container.
- Docker runs the equivalent of docker commit to commit a new layer.
- Docker then runs a new container from this new image.
- The next instruction in the file is executed, and the process repeats until all instructions have been executed

20. How to choose right desktop virtualization solution?

- Identify the costs associated with setting up the infrastructure and deployment of virtual desktops
- Determine the required resources and expertise to adopt these solutions
- Determine the infrastructure control capabilities of the virtualization providers
- Determine the level of elasticity and agility in desktop virtualization solution

21. State the tools for Network Virtualization.

- Physical switch OS
    - The OS must have the functionality of network virtualization.
- Hypervisor
    - The hypervisor is used to create a virtual switch and configuring virtual networks on it.
    - The third-party software is installed onto the hypervisor and it replaces the native networking functionality of the hypervisor.
    - A hypervisor allows us to have various VMs all working optimally on a single piece of computer hardware.

22. State the advantages and disadvantages of Network Virtualization.

Advantages of Network Virtualization:

- Improves manageability
- Reduces CAPEX (Capital Expenditure)
- Improves utilization
- Enhances performance
- Enhances security

Disadvantages of Network Virtualization:

- It needs to manage IT in the abstract.
- Increased complexity.
- Upfront cost- an amount of money paid before a particular service is done

23. What is the need to implement storage virtualization?

- Will improve the management of the storage.
- Less downtime as the storage availability is better.
- Will provide better storage utilization

24. What are the methods of Storage Virtualization?

- File-based Storage Virtualization
- Block-based Virtual Storage

25. What is meant by Address Space Remapping?

- Storage virtualization helps to achieve location independence by utilizing the physical location of the data.
- This system provides the space to the customer to store their data and handles the process of mapping.

26. Justify the importance of Storage Virtualization.

- Performs Tasks
- WAN Management
- Disaster Recovery
- Storage Tiering

27. What are the advantages and disadvantages of OS Extensions?

Advantages

- VMs at the operating system level have minimal startup/shutdown costs, low resource requirements, and high scalability
- All OS-level VMs on the same physical machine share a single operating system kernel

Disadvantages

- All the VMs at operating system level on a single container must have the same kind of guest operating system

28. What are the benefits of Application virtualization?

- Simplified management
- Increased Scalability
- Enhanced Security
- Allows the running of legacy apps.
- Enables cross-platform operations.
- Prevents conflicts with other virtualized apps.
- Permits users to run multiple app instances.

29. How to build image with a Docker file?

- Creating a sample repository
- First Dockerfile

```
# Version: 0.0.1
FROM ubuntu:14.04
MAINTAINER James Turnbull "james@example.com"
RUN apt-get update
RUN apt-get install -y nginx
RUN echo 'Hi, I am in your container' \
    >/usr/share/nginx/html/index.html
EXPOSE 80
```

- Create a directory called static_web to hold Dockerfile. Docker will upload the build context, as well as any files and directories contained in it, to the Docker daemon when the build is run. This provides the Docker daemon with direct access to any code, files or other data.

Part-B

1. Explain in detail about Desktop Virtualization.

- Desktop Virtualization
- Types of desktop virtualization
- Virtual Desktop Infrastructure
- Remote Desktop Services
- Desktop-as-a-Service (DaaS)
- Choosing the right desktop virtualization solution
- Benefits of Desktop Virtualization

➢ **Desktop Virtualization**

- Desktop virtualization is the process of separating the desktop environment and associated application software from the physical client device that is used to access it.
- Desktop virtualization is a method of simulating a user workstation so it can be accessed from a remotely connected device.
- By abstracting the user desktop in this way, organizations can allow users to work from virtually anywhere with a network connecting, using any desktop laptop, tablet, or smartphone.
- User data and programs exist in the desktop virtualization server, not on client devices as in figure 3.1.

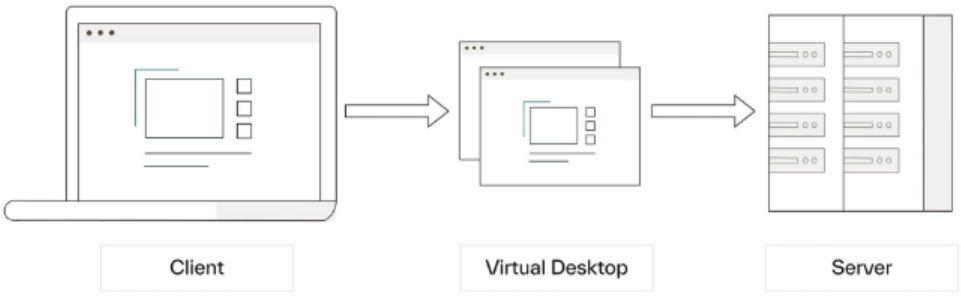


Figure 3.1 – Virtualized Desktop

- Desktop virtualization is also known as client virtualization because the client-server computing model is used in virtualizing desktops.
- The desktop virtualization is a technology that separates an individual's PC applications from their desktop and provides a way for users to maintain their individual desktops on a single, central server that can be connected to from a LAN, WAN, or simply over the Internet.

- Desktop virtualization includes the replacement of traditional physical desktop computing environments with virtual computing environments.
- It helps create and store multiple user desktop environments on a single host, residing in the cloud or a data center.

➢ **Types of desktop virtualization**

The three most popular deployment models of desktop virtualization are:

Virtual Desktop Infrastructure

- A virtual desktop interface (VDI) uses host-based virtual machines (VMs) to run the operating system.
- VDI technology leverages a hypervisor to split a server into different desktop images that users can remotely access via their endpoint devices.
- VDI gives each user their virtual machine and supports only one user per operating system.
- It delivers non-persistent and persistent virtual desktops to all connected devices.
- With a non-persistent virtual desktop, employees can access a virtual desktop from a shared pool, whereas in a persistent virtual desktop, each user gets a unique desktop image that can be customized with data and applications.

Remote Desktop Services

- Remote desktop services (RDS) or remote desktop session host (RDSH) are beneficial where only limited applications require virtualization.
- They allow users to remotely access Windows applications and desktops using the Microsoft Windows Server operating system.
- RDS is a more cost-effective solution, since one Windows server can support multiple users.

Desktop-as-a-Service (DaaS)

- Desktop-as-a-service (DaaS) is a flexible desktop virtualization solution that uses cloud-based virtual machines backed by a third-party provider.

- **Choosing the right desktop virtualization solution**
  - Identify the costs associated with setting up the infrastructure and deployment of virtual desktops
  - Determine the required resources and expertise to adopt these solutions
  - Determine the infrastructure control capabilities of the virtualization providers
  - Determine the level of elasticity and agility in desktop virtualization solution
- **Benefits of Desktop Virtualization**
  - Better security and control
  - Ease of maintenance
  - Remote work
  - Resource management
  - Reduced costs
  - Increased employee productivity
  - Improved flexibility

2. Explain in detail about Network Virtualization.

- Network Virtualization
- Tools for Network Virtualization :
- Functions of Network Virtualization
- Network Virtualization in Virtual Data Center :
- Advantages of Network Virtualization:
- Disadvantages of Network Virtualization :
- Examples of Network Virtualization :
- Applications of Network Virtualization :

- **Network Virtualization**
  - Network Virtualization is a process of logically grouping physical networks and making them operate as single or multiple independent networks called Virtual Networks.

- **Tools for Network Virtualization :**
  - Physical switch OS
    - The OS must have the functionality of network virtualization.
  - Hypervisor
    - The hypervisor is used to create a virtual switch and configuring virtual networks on it.
    - The third-party software is installed onto the hypervisor and it replaces the native networking functionality of the hypervisor.
    - A hypervisor allows us to have various VMs all working optimally on a single piece of computer hardware.
- **Functions of Network Virtualization :**
  - It enables the functional grouping of nodes in a virtual network.
  - It enables the virtual network to share network resources.
  - It allows communication between nodes in a virtual network without routing of frames.
  - It restricts management traffic.
  - It enforces routing for communication between virtual networks.
- **Network Virtualization in Virtual Data Center :**

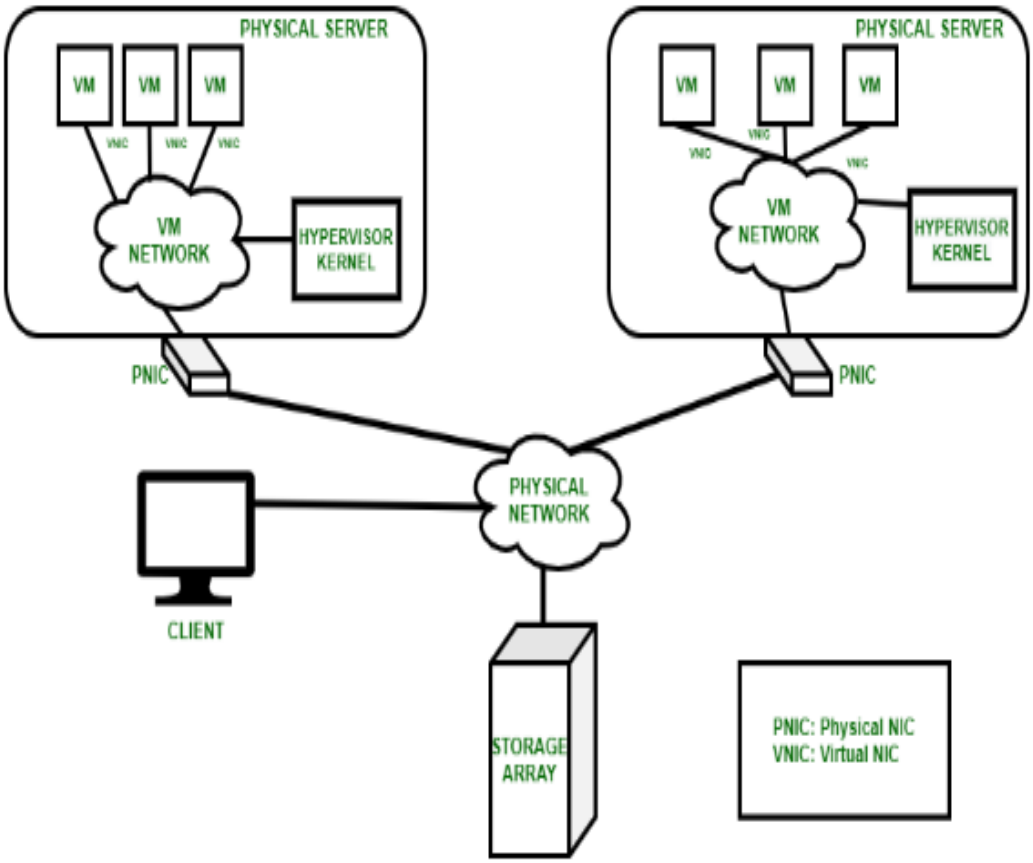


Figure 3.2 – Network Virtualization in VDC

1. Physical Network

- Physical components: Network adapters, switches, bridges, repeaters, routers and hubs.
- Grants connectivity among physical servers running a hypervisor, between physical servers and storage systems and between physical servers and clients.

2. VM Network

- Consists of virtual switches.
- Provides connectivity to hypervisor kernel.
- Connects to the physical network.
- Resides inside the physical server.

➢ **Advantages of Network Virtualization:**

1. Improves manageability – Grouping and regrouping of nodes are eased.

2. Reduces CAPEX (Capital Expenditure) – The requirement to set up separate physical networks for different node groups is reduced.
3. Improves utilization – Multiple VMs are enabled to share the same physical network which enhances the utilization of network resource.
4. Enhances performance – Network broadcast is restricted and VM performance is improved.
5. Enhances security – Sensitive data is isolated from one VM to another VM. Access to nodes is restricted in a VM from another VM.

➢ **Disadvantages of Network Virtualization :**

- It needs to manage IT in the abstract.
- Increased complexity.
- Upfront cost- an amount of money paid before a particular service is done.

➢ **Examples of Network Virtualization :**

Virtual LAN (VLAN)

- The performance and speed of busy networks can be improved by VLAN.
- VLAN can simplify additions or any changes to the network.

Network Overlays

- A framework is provided by an encapsulation protocol called VXLAN for overlaying virtualized layer 2 networks over layer 3 networks.

Network Virtualization Platform: VMware NSX

- VMware NSX Data Center transports the components of networking and security such as switching, firewalling and routing that are defined and consumed in software.
- It transports the operational model of a virtual machine (VM) for the network.

➢ **Applications of Network Virtualization :**

- Network virtualization may be used in the development of application testing to mimic real-world hardware and system software.
- It helps us to integrate several physical networks into a single network or separate single physical networks into multiple analytical networks.
- Network virtualization allows the simulation of connections between applications, services, dependencies, and end-users for software testing.

- It helps us to deploy applications in a quicker time frame, thereby supporting a faster go-to-market.
- Network virtualization helps the software testing teams to derive actual results with expected instances and congestion issues in a networked environment.

3. Discuss in detail about storage virtualization.

- Storage virtualization
- Need to implement Storage Virtualization
- Types of Storage Virtualization
- Methods of Storage Virtualization
- Address Space Remapping
- Importance of Storage Virtualization
- Ways to apply storage to a virtualized environment:
- Advantages of Storage Virtualization
- Disadvantages of Storage Virtualization
- Examples of Storage Virtualization

➢ **Storage virtualization**

- Storage virtualization in Cloud Computing is the sharing of physical storage into multiple storage devices which appears to be a single storage device.
- It can be also called as a group of an available storage device which simply manages from a central console.
- This virtualization provides numerous benefits such as easy backup, achieving, and recovery of the data.

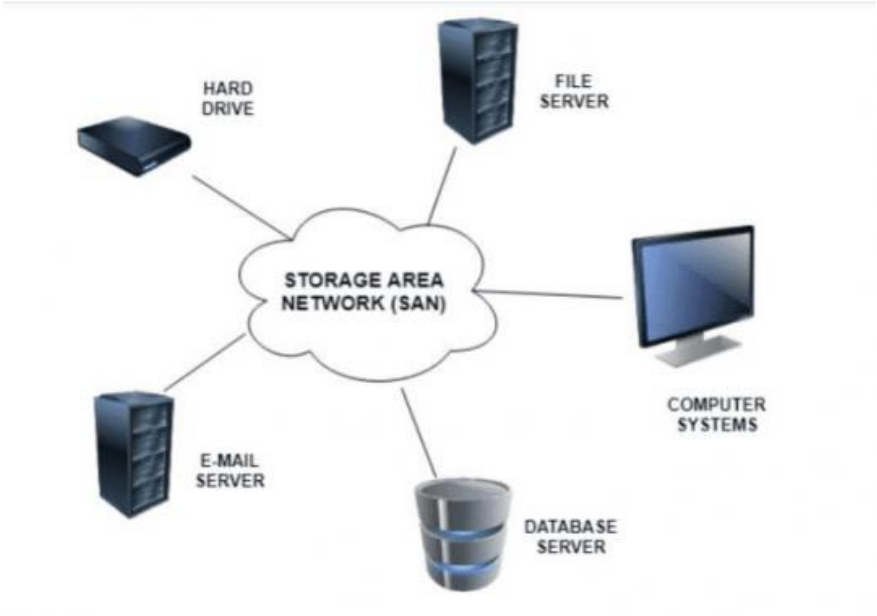


Figure 3.3 – Storage Virtualization

- **Need to implement Storage Virtualization**
  - Will improve the management of the storage.
  - Less downtime as the storage availability is better.
  - Will provide better storage utilization.

- **Types of Storage Virtualization**

  There are three different types of virtual storages:
  - **File Level Virtual Storage**
    - There is no need to manage disk space, it allows multiple people to share a storage device.
  - **Object Level Virtual Storage**
    - In object level virtual storage, the data is abstracted into data buckets rather than being stored in disks.
  - **Block Level Virtual Storage**
    - The server acts as a desktop computer. It can access virtual disks that function like normal hard drives.
      - It provides numerous functionalities such as booting and increased scalability and performance.

- **Methods of Storage Virtualization**

  i. **File-based Storage Virtualization**

- File-based storage virtualization utilizes server message block or network file system protocols.
- This is done between the data being accessed and the location of the physical memory

ii. Block-based Virtual Storage

- The block-based virtual storage system uses logical storage such as drive partition from the physical memory in a storage device.
- It also abstracts the logical storage such as a hard disk drive or any solid state memory device.

➢ **Address Space Remapping**

- Storage virtualization helps to achieve location independence by utilizing the physical location of the data.
- This system provides the space to the customer to store their data and handles the process of mapping.

➢ **Importance of Storage Virtualization**

- Performs Tasks
- WAN Management
- Disaster Recovery
- Storage Tiering

➢ **Ways to apply storage to a virtualized environment:**

- **Network Based**
  - Most organizations use network based virtualization. I
  - A network device, i.e., a purpose built server or smart switch, connects all devices in an iSCSI SAN.
  - It then presents the storage in the network as a virtual pool.
- **Host Based**
  - Host based storage virtualization involves a host that presents virtual drives to guest machines.
  - These guest machines can be cloud based virtual machines, file shares, or physical servers.
- **Array Based**
  - The storage array acts as the primary storage controller.
  - The storage array presents different types of storage tiers.
  - A storage tier can have HDDs or SSDs on multiple storage arrays.

➢ **Advantages of Storage Virtualization**

- Better Storage Utilization

- Easier Data Management
- Enables Addition of Advanced Features
- Improved Data Security
- Scalability
- Easier Data Retrieval

➢ **Disadvantages of Storage Virtualization**

- Increased Traffic in the Storage Area Network
- Shared Environment May Lead to Data Compromise

➢ **Examples of Storage Virtualization**

- **Logical Unit Number (LUN).** LUN is a unique number used to identify a logical unit for computer storage.
- **RAID groups.** RAID (Redundant Array of Independent Disks) is a storage technology that features a configuration of multiple hard drives to work as a single computer system.
- **Logical Volume (LV).** A Logical Volume is a practice of combining multiple disk partitions or hard drives into a single volume group (VG).

4. Discuss in detail about system level of operating virtualization.

- System level of operating virtualization
- Advantages of OS Extensions
- Disadvantages of OS Extensions
- Virtualization on Linux or Windows Platforms

➢ **System level of operating virtualization**

- Operating-system-level virtualization is a server-virtualization which inserts a virtualization layer inside an operating system to partition a machine's physical resources.
- It enables multiple isolated VMs within a single operating system kernel.
- This kind of VM is often called a virtual execution environment (VE), Virtual Private System (VPS), or container.
- VEs like real servers have its own set of processes, file system, user accounts, network interfaces with IP addresses, routing tables, firewall rules, and other personal settings.
- Therefore, OS-level virtualization is also called single-OS image virtualization.

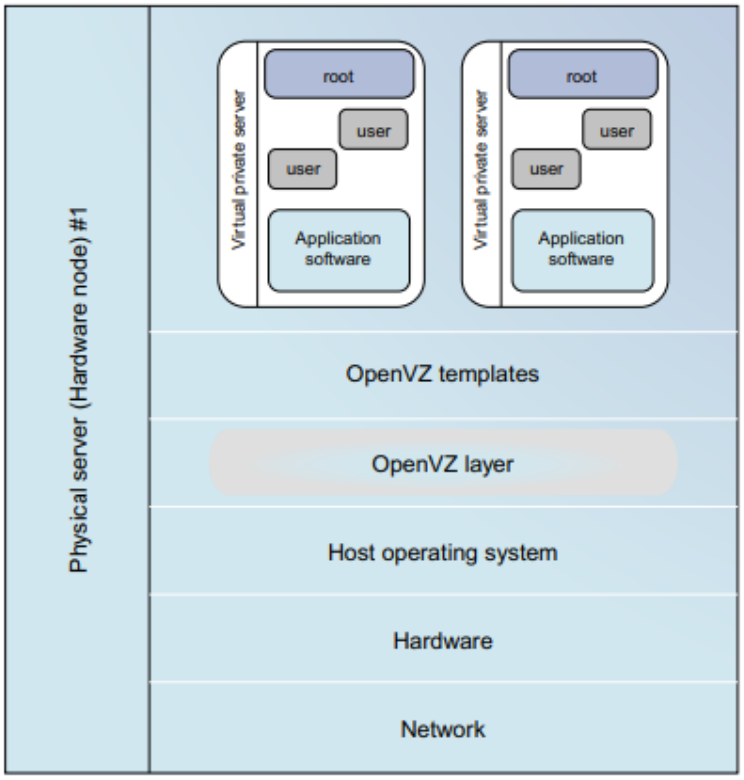


Figure 3.4 – System level of operating virtualization

- Figure 3.4 shows the OpenVZ virtualization layer inside the host OS, which provides some OS images to create VMs quickly.
- The virtualization layer is inserted inside the OS to partition the hardware resources for multiple VMs to run their applications in multiple virtual environments.
- To implement OS-level virtualization, isolated execution environments (VMs) should be created based on a single OS kernel.
- The chroot command in a UNIX system can create several virtual root directories within a host OS.
- These virtual root directories are the root directories of all VMs created.
- There are two ways to implement virtual root directories:
  - duplicating common resources to each VM partition;
  - sharing most resources with the host environment and only creating private resource copies on the VM on demand.

Advantages of OS Extensions

- VMs at the operating system level have minimal startup/shutdown costs, low resource requirements, and high scalability;

- it is possible for a VM and its host environment to synchronize state changes when necessary.
- All OS-level VMs on the same physical machine share a single operating system kernel;
- The virtualization layer can be designed in a way that allows processes in VMs to access as many resources of the host machine as possible, but never to modify them.

Disadvantages of OS Extensions

- All the VMs at operating system level on a single container must have the same kind of guest operating system.

Virtualization on Linux or Windows Platforms

- Most OS-level virtualization systems are Linux-based.
- The Linux kernel offers an abstraction layer to allow software processes to work with and operate on resources without knowing the hardware details.
- New hardware may need a new Linux kernel to support.
- Therefore, different Linux platforms use patched kernels to provide special support for extended functionality.
- Two OS tools (Linux vServer and OpenVZ) support Linux platforms to run other platform-based applications through virtualization.

5. Discuss in detail about Application Virtualization.

- Application Virtualization
- Working of Application Virtualization
- Reason for Application Virtualization
- Application virtualization benefits

➢ **Application Virtualization**

- Application virtualization helps a user to have remote access to an application from a server.
- Application virtualization software allows users to access and use an application from a separate computer than the one on which the application is installed.

- Using application virtualization software, IT admins can set up remote applications on a server and deliver the apps to an end user's computer as shown in Figure 3.5.

Figure 3.5 – Application virtualization

➢ **Working of Application Virtualization**

- The most common way to virtualize applications is the server-based approach.
- This means an IT administrator implements remote applications on a server inside an organization's datacenter or via a hosting service.
- The IT admin then uses application virtualization software to deliver the applications to a user's desktop or other connected device.
- The user can then access and use the application as though it were locally installed on their machine, and the user's actions are conveyed back to the server to be executed.

➢ **Reason for Application Virtualization**

- Limitation of expenditures
- Remote-safe approach
- Implementation of in-house applications

➢ **Application virtualization benefits**

- Simplified management
- Increased Scalability
- Enhanced Security
- Allows the running of legacy apps.
- Enables cross-platform operations.
- Prevents conflicts with other virtualized apps.
- Permits users to run multiple app instances.

6. Discuss in detail about Virtual clusters and Resource Management.

Compare and contrast the Physical Clusters and the Virtual Clusters and depict how resource management could be carried out in virtual machines. **Nov 2023**

- Physical versus Virtual Clusters
- Fast Deployment
- Live VM Migration Steps
- Live migration of a VM consists of the following steps:
- Memory Migration
- File Migration
- Network Migration

Physical versus Virtual Clusters

- A physical cluster is a collection of servers (physical machines) interconnected by a physical network such as a LAN.
- Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters.
- The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks.

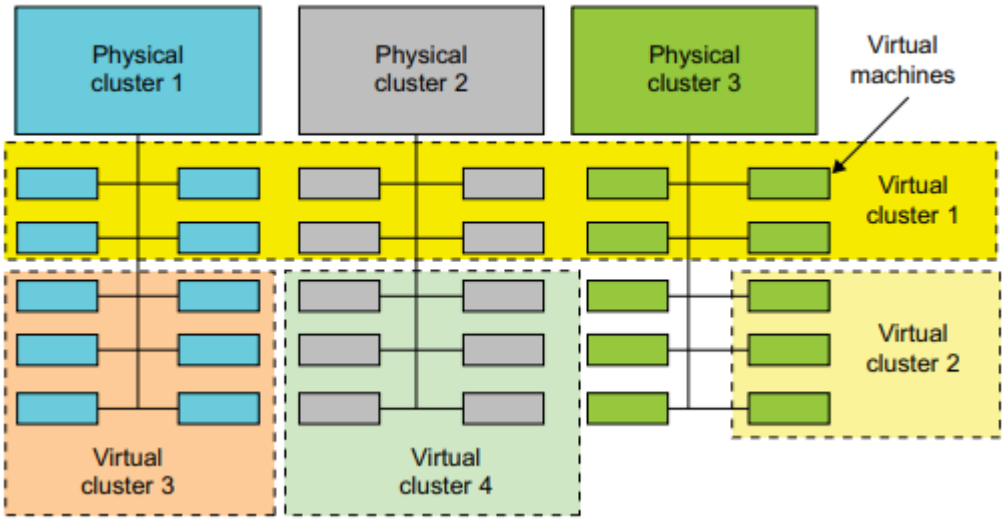


Figure 3.6 – virtual clusters and physical clusters

- Figure 3.6 illustrates the concepts of virtual clusters and physical clusters.
- Three physical clusters are shown on the left side of Figure 3.6.
- Four virtual clusters are created on the right, over the physical clusters.
- Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters.

- The physical machines are also called host systems. In contrast, the VMs are guest systems.
- The host and guest systems may run with different operating systems.
- Each VM can be installed on a remote server or replicated on multiple servers belonging to the same or different physical clusters.
- The boundary of a virtual cluster can change as VM nodes are added, removed, or migrated dynamically over time.
- The provisioning of VMs to a virtual cluster is done dynamically with the following properties:
  - The virtual cluster nodes can be either physical or virtual machines.
  - A VM runs with a guest OS
  - The purpose of using VMs is to consolidate multiple functionalities on the same server.

Fast Deployment

- To construct and distribute software stacks (OS, libraries, applications) to a physical node inside clusters as fast as possible, and to quickly switch runtime environments from one user's virtual cluster to another user's virtual cluster.
- If one user finishes using his system, the corresponding virtual cluster should shut down or suspend quickly to save the resources to run other VMs for other users.
- The live migration of VMs allows workloads of one node to transfer to another node.
- Load balancing can be achieved using the load index and frequency of user logins.
- The automatic scale-up and scale-down mechanism of a virtual cluster can be implemented.
- There are four steps to deploy a group of VMs onto a target cluster:
  - preparing the disk image,
  - configuring the VMs,
  - choosing the destination nodes,
  - executing the VM deployment command on every host.
- A template is a disk image that includes a preinstalled operating system with or without certain application software.

Users choose a proper template according to their requirements and make a duplicate of it as their own disk image.

Templates could implement the COW (Copy on Write) format. A new COW backup file is very small and easy to create and transfer. Therefore, it definitely reduces disk space consumption.

- Every VM is configured with a name, disk image, network setting, and allocated CPU and memory.
- The system configures the VMs according to the chosen profile.
- A strategy to choose the proper destination host for any VM is needed.
- The deployment principle is to fulfill the VM requirement and to balance workloads among the whole host network.

Live VM Migration Steps

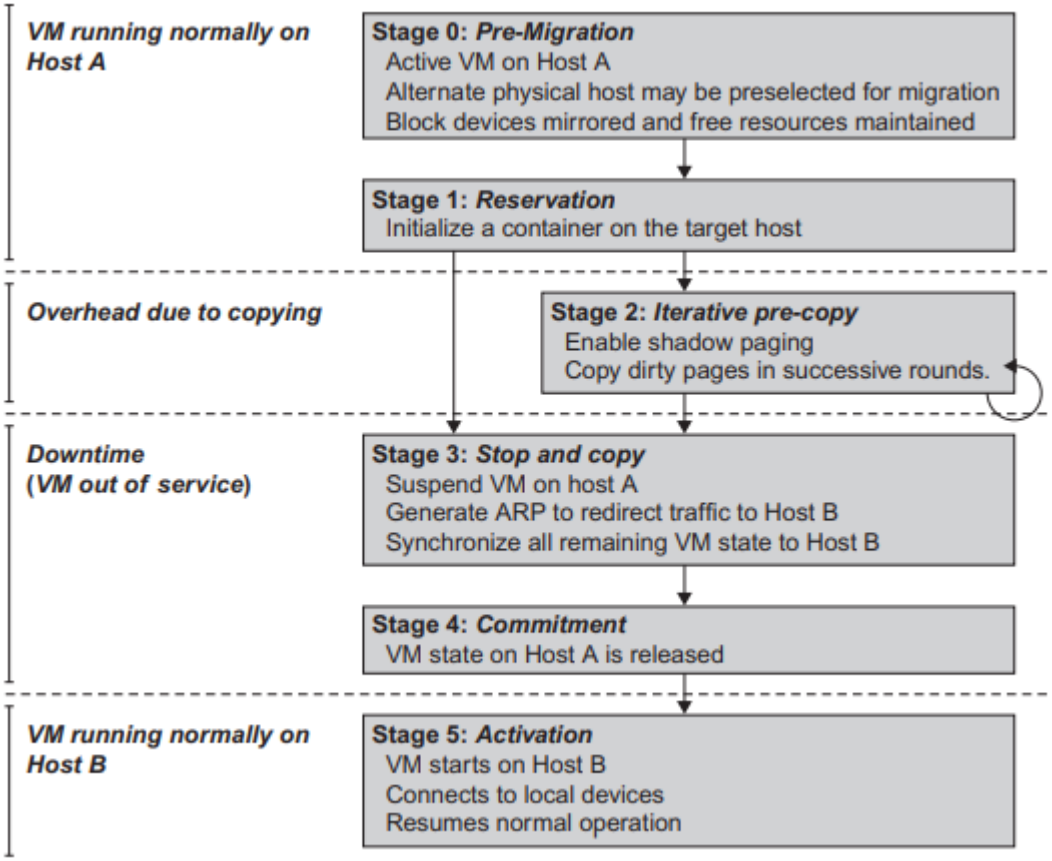


Figure 3.7 - Live migration process of a VM from one host to another

- VMs can be live-migrated from one physical machine to another as shown in figure 3.7; in case of failure, one VM can be replaced by another VM.
- The motivation is to design a live VM migration scheme with negligible downtime, the lowest network bandwidth consumption possible, and a reasonable total migration time.
- A VM can be in one of the following four states.
  - An ***inactive state*** - the VM is not enabled.
  - An ***active state*** - a VM has been instantiated at the to perform a real task.
  - A ***paused state*** - a VM that has been instantiated but disabled to process a task or paused in a waiting state.
  - A VM enters the ***suspended state*** if its machine file and virtual resources are stored back to the disk.

Live migration of a VM consists of the following steps:

- **Steps 0 and 1: Start migration.**
  - This step makes preparations for the migration, including determining the migrating VM and the destination host.
  - The migration is automatically started by strategies such as load balancing and server consolidation.
- **Steps 2: Transfer memory.**
  - Since the whole execution state of the VM is stored in memory, all of the memory data is transferred in the first round, and then the migration controller recopies the memory data which is changed in the last round.
  - These steps keep iterating until the dirty portion of the memory is small enough to handle the final copy.
- **Step 3: Suspend the VM and copy the last portion of the data.**
  - The migrating VM's execution is suspended when the last round's memory data is transferred.
  - During this step, the VM is stopped and its applications will no longer run.
  - This "service unavailable" time is called the "downtime" of migration.

- **Steps 4 and 5: Commit and activate the new host.**
  - After all the needed data is copied, on the destination host, the VM reloads the states and recovers the execution of programs in it, and the service provided by this VM continues.
  - Then the network connection is redirected to the new VM and the dependency to the source host is cleared.
  - The whole migration process finishes by removing the original VM from the source host.

Memory Migration

- Memory migration can be in a range of hundreds of megabytes to a few gigabytes, and it needs to be done in an efficient manner.
- The Internet Suspend-Resume (ISR) technique exploits temporal locality as memory states are likely to have considerable overlap in the suspended and the resumed instances of a VM.
- Temporal locality refers to the fact that the memory states differ only by the amount of work done.

File Migration

- The actual file systems themselves are not mapped onto the distributed file system.
- Instead, the VMM only accesses its local file system.
- The relevant VM files are explicitly copied into the local file system for a resume operation and taken out of the local file system for a suspend operation.

Network Migration

- The VMM maintains a mapping of the virtual IP and MAC addresses to their corresponding VMs. In
- general, a migrating VM includes all the protocol states and carries its IP address with it

7. Differentiate between Containers vs. Virtual Machines

- Containers vs. Virtual Machines
- Virtual Machine:
- Container:
- difference between Virtual machines and Containers
- Popular container providers

Containers vs. Virtual Machines

- Virtual machines and Containers are two ways of deploying multiple, isolated services on a single platform.

Virtual Machine:

- It runs on top of emulating software called the hypervisor which sit between the hardware and the virtual machine.
- The hypervisor is the key to enable virtualization.
- It manages the sharing of physical resources into virtual machines.
- Each virtual machine runs its own guest operating system.
- They are less agile and have low portability than containers.
- Refer Figure 3.8A for the structure of Virtual Machines

Container:

- Containers are lightweight software packages that contain all the dependencies required to execute the contained software application.
- These dependencies include things like system libraries, external third-party code packages, and other operating system level applications.
- The dependencies included in a container exist in stack levels that are higher than the operating system.
- It sits on the top of a physical server and its host operating system.
- They are more agile and have high portability than virtual machines.
- Refer Figure 3.8B for the structure of Container

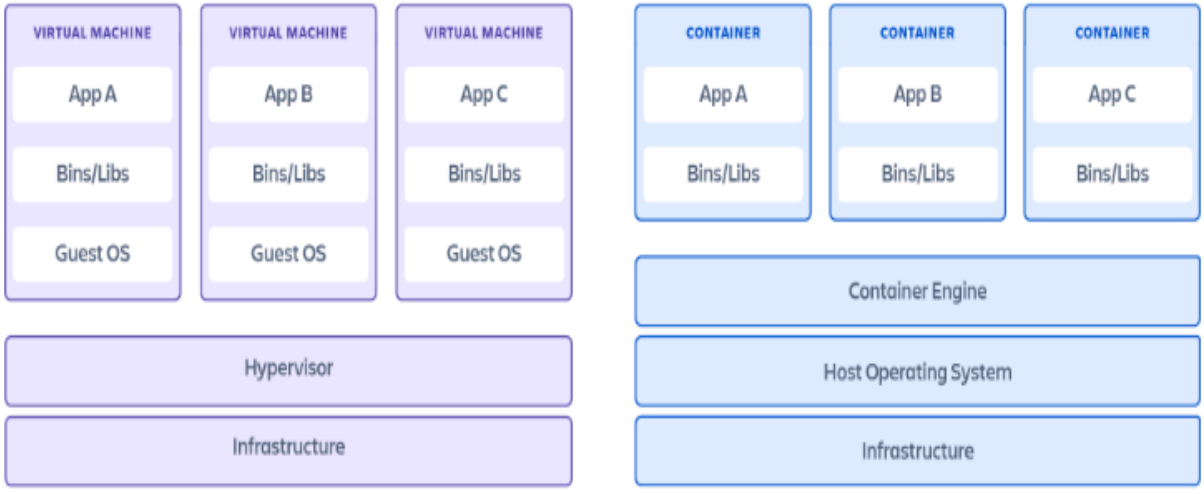


Figure 3.8A – Virtual Machines & Figure 3.8B - Container

Difference between Virtual machines and Containers.

S. No.	Virtual Machines(VM)	Containers
1	VM is piece of software that allows to install other software inside of it.	While a container is a software that allows different functionalities of an application independently.
2.	Applications running on VM system can run different OS.	While applications running in a container environment share a single OS.
3.	VM virtualizes the computer system.	While containers virtualize the operating system only.
4.	VM size is very large.	While the size of container is very light; i.e. a few megabytes.
5.	VM takes minutes to run, due to large size.	While containers take a few seconds to run.
6.	VM uses a lot of system memory.	While containers require very less memory.
7.	VM is more secure.	While containers are less secure.
8.	VM's are useful when we require all of OS resources to run various applications.	While containers are useful when we are required to maximize the running applications using minimal servers.
9.	Examples of VM are: KVM, Xen, VMware.	While examples of containers are: RancherOS, PhotonOS, Containers by Docker.

Container providers

Docker

- Docker is the most popular and widely used container runtime.
- Docker Hub is a giant public repository of popular containerized software applications.
- Containers on Docker Hub can instantly downloaded and deployed to a local Docker runtime.

RKT

- Pronounced "Rocket", RKT is a security-first focused container system.
- RKT containers do not allow insecure container functionality unless the user explicitly enables insecure features.

Linux Containers (LXC)

- The Linux Containers project is an open-source Linux container runtime system.
- LXC is used to isolate operating, system-level processes from each other.

8. Discuss in detail about Introduction to Docker and explain Docker Components.

What are the different Components of a Docker? Explain its need and use. Nov 2023

Introduction to Docker

- Docker is an open-source engine that automates the deployment of applications into containers.
- It was written by the team at Docker, Inc and released by them under the Apache 2.0 license.
- Docker adds an application deployment engine on top of a virtualized container execution environment.
- Docker's mission is to provide:
  - **An easy and lightweight way to model reality**
    - Docker is fast. Can Dockerize application in minutes.

- Docker relies on a copy-on-write model so that making changes to the application is also incredibly fast: only want to change gets changed.
- Can then create containers running the applications..

- **A logical segregation of duties**
  - With Docker, Developers care about their applications running inside containers, and Operations cares about managing the containers.
  - Docker is designed to enhance consistency by ensuring the environment in which the developers write code matches the environments into which the applications are deployed.
- **Fast, efficient development life cycl**e
  - Docker aims to reduce the cycle time between code being written and code being tested, deployed, and used. It aims to make applications portable, easy to build, and easy to collaborate on.
- **Encourages service orientated architecture**
  - Docker also encourages service orientated and micro services architectures.
  - Docker recommends that each container run a single application or process.
  - This promotes a distributed application model where an application or service is represents by a series of inter-connected containers.
  - This makes it very easy to distribute, scale, debug and introspect the applications.

Docker components

- The Docker client and server
- Docker Images
- Registries
- Docker Containers

➢ **Docker client and server**

- Docker is a client-server application.
- The Docker client talks to the Docker server or daemon, which, in turn, does all the work.

- Docker ships with a command line client binary, docker, as well as a full RESTful API.
- Can run the Docker daemon and client on the same host or connect the local Docker client to a remote daemon running on another host.
- Figure 3.9 depicts the Docker Architecture diagram in detail.

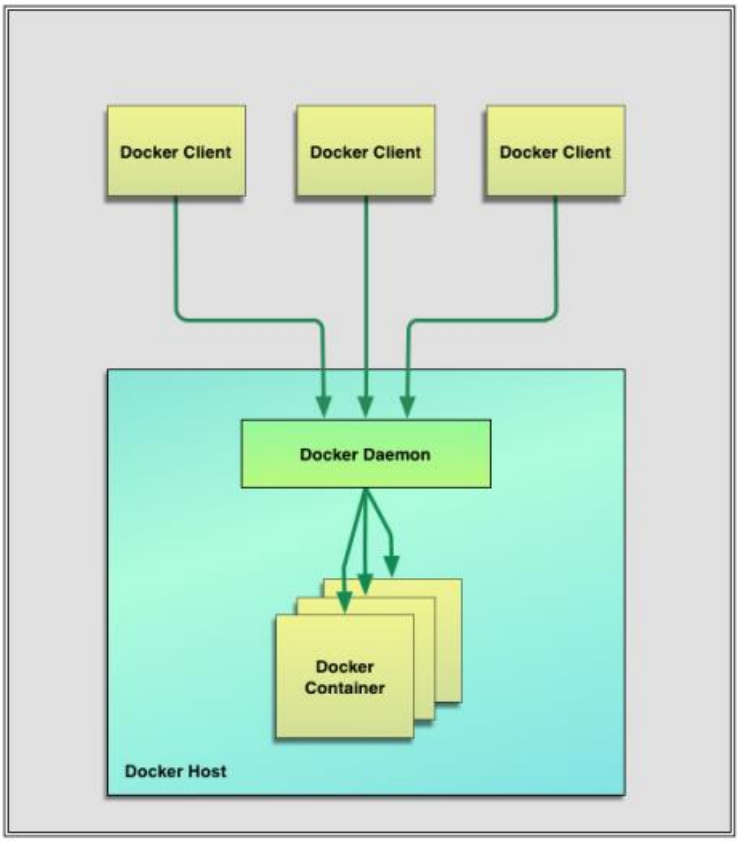


Figure 3.9 – Docker Architecture

➢ **Docker images**

- Images are the building blocks of the Docker world.
- Can launch the containers from images.
- Images are the "build" part of Docker's life cycle.
- They are a layered format, using Union file systems, that are built step-by-step using a series of instructions.
- For example:
  - Add a file.
  - Run a command.
  - Open a port.

➢ **Registries**

- Docker stores the images you build in registries.
- There are two types of registries:
  - public
  - private.
- Docker, operates the public registry for images, called the Docker Hub.
- Can create an account on the Docker Hub and use it to share and store own images.

➢ **Containers**

- Docker helps to build and deploy containers inside of which can package the applications and services.
- A Docker container is:
  - An image format.
  - A set of standard operations.
  - An execution environment.
- Docker containers ship software.
- Each container contains a software image and allows a set of operations to be performed.
- For example, it can be created, started, stopped, restarted, and destroyed.

9. Discuss in detail about Docker Containers and Creating and managing containers.

- Ensuring Docker is ready
- Building the first container
- Container naming
- Starting a stopped container
- Stopping a daemonized container
- Deleting a container

➢ **Ensuring Docker is ready -**

- Check that the Docker binary exists and is functional:

Checking the docker binary works

```
$ sudo docker info
Containers: 0
Images: 0
Storage Driver: aufs
 Root Dir: /var/lib/docker/aufs
 Dirs: 144
Execution Driver: native-0.1
Kernel Version: 3.8.0-29-generic
Registry: [https://index.docker.io/v1/]
```

- Passed the info command to the Docker binary, which returns a list of any containers, any images, the execution and storage drivers Docker is using, and its basic configuration.
- Docker has a client-server architecture. As a client, the Docker binary passes requests to the Docker daemon and then processes those requests when they are returned.

➢ **Building the first container**

Docker run command is used to create a container.

Creating our first container

```
$ sudo docker run -i -t ubuntu /bin/bash
Pulling repository ubuntu from https://index.docker.io/v1
Pulling image 8↩
  dbd9e392a964056420e5d58ca5cc376ef18e2de93b5cc90e868a1bbc8318c1c ↩
  (precise) from ubuntu
Pulling 8↩
  dbd9e392a964056420e5d58ca5cc376ef18e2de93b5cc90e868a1bbc8318c1c ↩
  metadata
Pulling 8↩
  dbd9e392a964056420e5d58ca5cc376ef18e2de93b5cc90e868a1bbc8318c1c ↩
  fs layer
Downloading 58337280/? (n/a)
Pulling image ↩
  b750fe79269d2ec9a3c593ef05b4332b1d1a02a62b4accb2c21d589ff2f5f2dc↩
   (quantal) from ubuntu
Pulling image 27cf784147099545 () from ubuntu
root@fcd78e1a3569:/#
```

The docker run command

```
$ sudo docker run -i -t ubuntu /bin/bash
```

- Docker runs a command using docker run with two command line flags: -i and -t.
  - The -i flag keeps STDIN open from the container, for an interactive shell.
  - The -t flag is the other half and tells Docker to assign a pseudo-tty to the container.
- The ubuntu image is a stock image, also known as a "base" image, on the Docker Hub registry.
- Once Docker had found the image, it downloaded the image and stored it on the local host.
- Docker then used this image to create a new container inside a file system.
- The container has a network, IP address, and a bridge interface to talk to the local host.

When the container had been created, Docker ran the /bin/bash command inside it;

Listing 3.4: Our first container's shell

```
root@f7cbdac22a02:/#
```

Listing 3.5: Checking the container's hostname

```
root@f7cbdac22a02:/# hostname
f7cbdac22a02
```

.

Listing 3.6: Checking the container's /etc/hosts

```
root@f7cbdac22a02:/# cat /etc/hosts
172.17.0.4  f7cbdac22a02
127.0.0.1   localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Listing 3.7: Checking the container's interfaces

```
root@f7cbdac22a02:/#  ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 1500 qdisc noqueue state ↩
 UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
899: eth0: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast ↩
 state UP group default qlen 1000
    link/ether 16:50:3a:b6:f2:cc brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.4/16 scope global eth0
    inet6 fe80::1450:3aff:feb6:f2cc/64 scope link
       valid_lft forever preferred_lft forever
```

Listing 3.9: Installing a package in our first container

```
root@f7cbdac22a02:/# apt-get update && apt-get install vim
```

- The container only runs for as long as the command specified, /bin/bash, is running. Once exited the container, that command ended, and the container was stopped
- There are three ways containers can be identified: a short UUID (like f7cbdac22a02), a longer UUID (like f7cbdac22a02e03c9438c729345e54db9d20cf a2ac1fc3494b6eb60872e74778), and a name (like gray_cat).

➢ **Container naming**

Listing 3.10: Naming a container

```
$ sudo docker run --name bob_the_container -i -t ubuntu /bin/bash
root@aa3f365f0f4e:/# exit
```

- This would create a new container called bob_the_container.
- A valid container name can contain the following characters: a to z, A to Z, the digits 0 to 9, the underscore, period, and dash.
- Container names are useful to help us identify and build logical connections between containers and applications.
- Names are unique.
- To create two containers with the same name, the command will fail.

➢ **Starting a stopped container**

Listing 3.11: Starting a stopped container

```
$ sudo docker start bob_the_container
```

Listing 3.12: Starting a stopped container by ID

```
$ sudo docker start aa3f365f0f4e
```

- Creating daemonized containers can create longer-running containers.
- Daemonized containers are ideal for running applications and services.
- Let's start a daemonized container.

Listing 3.16: Creating a long running container

```
$ sudo docker run --name daemon_dave -d ubuntu /bin/sh -c "while ↩
  true; do echo hello world; sleep 1; done"
1333bb1a66af402138485fe44a335b382c09a887aa9f95cb9725e309ce5b7db3
```

➢ **Stopping a daemonized container**

Stopping the running Docker container

$ sudo docker stop daemon_dave

Stopping the running Docker container by ID

$ sudo docker stop c2c4e57c12c4

- The docker stop command sends a SIGTERM signal to the Docker container's running process.

➢ **Deleting a container**

- Stop it first using the docker stop command or docker kill command.

Deleting a container

$ sudo docker rm 80430f8d0921

Deleting all containers

docker rm `docker ps -a -q`

- The -a flag lists all containers, and the -q flag only returns the container IDs rather than the rest of the information about the containers.
- This list is then passed to the docker rm command, which deletes each container.

10. Discuss in detail about working with Docker Images and Repositories.

- Docker image
- Listing Docker images
- The Docker Pull Command
- Searching for images
- Building our own images
    - Using Docker commit to create images
    - Building images with a Dockerfile
        - Dockerfile Instructions
- Pushing images to the Docker Hub
- Deleting an image

➢ DOCKER IMAGE

- Docker images: the building blocks from which containers are Launched.
- A Docker image is made up of file systems layered over each other as shown in figure 3.10.
- At the base is a boot filesystem, bootfs, which resembles the typical Linux/Unix boot filesystem.
- Docker next layers a root filesystem, rootfs, on top of the boot filesystem. This rootfs can be one or more operating systems
- The root filesystem stays in read-only mode, and Docker uses a union mount to add more read-only filesystems onto the root filesystem.
- A union mount is a mount that allows several filesystems to be mounted at one time but appear to be one filesystem.
- Images can be layered on top of one another.
- The image below is called the parent image and the final image is called the base image.
- Finally, when a container is launched from an image,

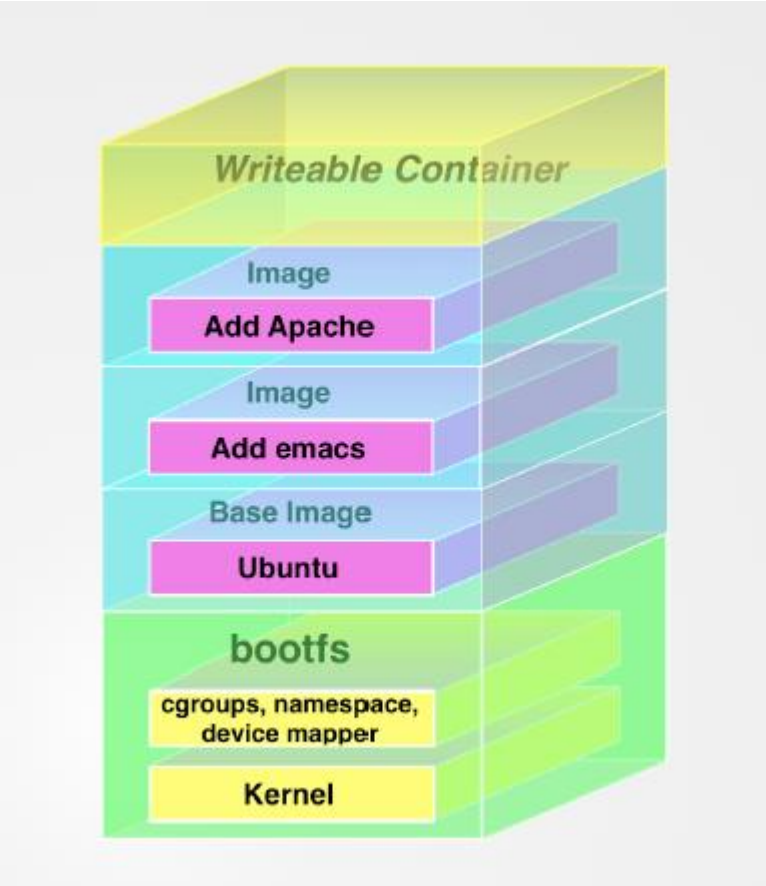


Figure 3.10 – The Docker filesystem layers

- When a container is created, Docker builds from the stack of images and then adds the read-write layer on top.
- That layer, combined with the knowledge of the image layers below it and some configuration data, form the container.

➢ LISTING DOCKER IMAGES

- The docker images command

```
$ sudo docker images
REPOSITORY TAG     IMAGE ID     CREATED     VIRTUAL SIZE
ubuntu     latest  c4ff7513909d 6 days ago  225.4 MB
```

- Local images live on the local Docker host in the repositories and repositories live on registries.
- The default registry is the public registry managed by Docker, Inc., Docker Hub.
- Inside Docker Hub images are stored in repositories.
- That image was downloaded from a repository.

- There are two types of repositories:
  - user repositories, which contain images contributed by Docker users,
  - top-level repositories, which are controlled by the people behind Docker.

The Docker Pull Command

- To pull down the entire contents of the ubuntu repository.

```
$ sudo docker pull ubuntu
Pulling repository ubuntu
c4ff7513909d: Pulling dependent layers
3db9c44f4520: Pulling dependent layers
75204fdb260b: Pulling dependent layers
```

- Listing all the ubuntu Docker images

```
$ sudo docker images
REPOSITORY TAG     IMAGE ID     CREATED     VIRTUAL SIZE
ubuntu     13.10   5e019ab7bf6d 2 weeks ago 180 MB
ubuntu     saucy   5e019ab7bf6d 2 weeks ago 180 MB
ubuntu     12.04   74fe38d11401 2 weeks ago 209.6 MB
ubuntu     precise 74fe38d11401 2 weeks ago 209.6 MB
ubuntu     12.10   a7cf8ae4e998 2 weeks ago 171.3 MB
ubuntu     quantal a7cf8ae4e998 2 weeks ago 171.3 MB
ubuntu     14.04   99ec81b80c55 2 weeks ago 266 MB
ubuntu     latest  c4ff7513909d 6 days ago  225.4 MB
ubuntu     trusty  99ec81b80c55 2 weeks ago 266 MB
ubuntu     raring  316b678ddf48 2 weeks ago 169.4 MB
ubuntu     13.04   316b678ddf48 2 weeks ago 169.4 MB
ubuntu     10.04   3db9c44f4520 3 weeks ago 183 MB
ubuntu     lucid   3db9c44f4520 3 weeks ago 183 MB
```

➢ **SEARCHING FOR IMAGES**

$ sudo docker search puppet

The above command will search images and return:

- Repository names
- Image descriptions
- Stars - these measure the popularity of an image
- Official - an image managed by the upstream developer
- Automated - an image built by the Docker Hub's Automated Build process

➢ BUILDING OWN IMAGES

There are two ways to create a Docker image:

• the docker commit command

• the docker build command with a Dockerfile

Using Docker commit to create images

- Create a container, make changes to that container and then commit those changes to a new image.

```
$ sudo docker commit 4aab3ce3cb76 jamtur01/apache2
8ce0ea7a1528
```

- Committing another custom container

```
$ sudo docker commit -m="A new custom image" --author="James ↩
 Turnbull" \
4aab3ce3cb76 jamtur01/apache2:webserver
f99ebb6fed1f559258840505a0f5d5b6173177623946815366f3e3acff01adef
```

  - -m option which allows to provide a commit message
  - the --author option to list the author of the image.
  - the ID of the container committing.
  - Finally, specified the username and repository of the image, jamtur01/apache2, and
  - added a tag, webserver, to the image.

- Docker executing instructions roughly follow a workflow:
  - Docker runs a container from the image.
  - An instruction executes and makes a change to the container.
  - Docker runs the equivalent of docker commit to commit a new layer.
  - Docker then runs a new container from this new image.
  - The next instruction in the file is executed, and the process repeats until all instructions have been executed.

Building images with a Dockerfile

- The Dockerfile uses a basic DSL with instructions for building Docker images.
- Then use the docker build command to build a new image from the instructions in the Dockerfile.

Creating a sample repository

```
$ mkdir static_web
$ cd static_web
$ touch Dockerfile
```

First Dockerfile

```
# Version: 0.0.1
FROM ubuntu:14.04
MAINTAINER James Turnbull "james@example.com"
RUN apt-get update
RUN apt-get install -y nginx
RUN echo 'Hi, I am in your container' \
    >/usr/share/nginx/html/index.html
EXPOSE 80
```

- Create a directory called static_web to hold Dockerfile.
- Docker will upload the build context, as well as any files and directories contained in it, to the Docker daemon when the build is run.
- This provides the Docker daemon with direct access to any code, files or other data.
- The Dockerfile contains a series of instructions paired with arguments.
- Each instruction, should be in upper-case and be followed by an argument:

Dockerfile Instructions

CMD, ENTRYPOINT, ADD, COPY, VOLUME, WORKDIR, USER, ONBUILD, and ENV.

CMD

- The CMD instruction specifies the command to run when a container is launched.

CMD ["/bin/true"]

- Can only specify one CMD instruction in a Dockerfile.
- If more than one is specified, then the last CMD instruction will be used.
- To run multiple processes or commands should use a service management tool like Supervisor.

ENTRYPOINT

The ENTRYPOINT instruction provides a command that isn't as easily overridden.

ENTRYPOINT ["/usr/sbin/nginx"]

- This allows to build in a default command to execute when container is run combined with override options and flags on the docker run command line.

WORKDIR

- The WORKDIR instruction provides a way to set the working directory for the container and the ENTRYPOINT and/or CMD to be executed when a container is launched from the image.

```
WORKDIR /opt/webapp/db
RUN bundle install
WORKDIR /opt/webapp
ENTRYPOINT [ "rackup" ]
```

- Have changed into the /opt/webapp/db directory to run bundle install and then changed into the /opt/webapp directory prior to specifying our ENTRYPOINT instruction of rackup.

ENV

The ENV instruction is used to set environment variables during the image build process.

ENV RVM_PATH /home/rvm/

- This new environment variable will be used for any subsequent RUN instructions.

USER

- The USER instruction specifies a user that the image should be run as;

USER nginx

This will cause containers created from the image to be run by the nginx user.

VOLUME

- The VOLUME instruction adds volumes to any container created from the image.
- A volume is a specially designated directory within one or more containers that bypasses the Union File System to provide several useful features for persistent or shared data:
  - Volumes can be shared and reused between containers.
  - A container doesn't have to be running to share its volumes.
  - Changes to a volume are made directly.

- Changes to a volume will not be included when an image is updated.
- Volumes persist until no containers use them.

VOLUME ["/opt/project"]

- This would attempt to create a mount point /opt/project to any container created from the image.

ADD

- The ADD instruction adds files and directories from build environment into image.
- The ADD instruction specifies a source and a destination for the files,

ADD software.lic /opt/application/software.lic

- This ADD instruction will copy the file software.lic from the build directory to /opt/application/software.lic in the image.
- The source of the file can be a URL, filename, or directory as long as it is inside the build context or environment.

COPY

- The COPY instruction is closely related to the ADD instruction.
- The key difference is that the COPY instruction is purely focused on copying local files from the build context and does not have any extraction or decompression capabilities.

COPY conf.d/ /etc/apache2/

- This will copy files from the conf.d directory to the /etc/apache2/ directory.

ONBUILD

- The ONBUILD instruction adds triggers to images.
- A trigger is executed when the image is used as the basis of another image
- The trigger can be any build instruction.
- For example:

```
ONBUILD ADD . /app/src
ONBUILD RUN cd /app/src && make
```

➢ **PUSHING IMAGES TO THE DOCKER HUB**

- Push images to the Docker Hub using the docker push command
- To push a root image

$ sudo docker push static_web

2013/07/01 18:34:47 Impossible to push a "root" repository. ↩
Please rename your repository in <user>/<repo> (ex: jamtur01/↩
static_web)

- Pushes the image to the repository static_web

➢ **DELETING AN IMAGE**

- The following command is to delete images that are not needed anymore using the docker rmi command.

$ sudo docker rmi jamtur01/static_web
Untagged: 06c6c1f81534
Deleted: 06c6c1f81534
Deleted: 9f551a68e60f
Deleted: 997485f46ec4
Deleted: a101d806d694
Deleted: 85130977028d

- This only deletes the image locally.
- If the image is previously pushed to the Docker Hub, it'll still exist there.
- To delete an image's repository on the Docker Hub, need to sign in and delete it there using the Delete repository link.

MAILAM Engineering College

AApproved by AICTE, New Delhi, Permanently Affiliated to Anna University Chennai,Accredited by NBA, NAAC with A Grade and TCS)

DEPARTMENT OF

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

III YEAR / V SEM

CCS335 CLOUD COMPUTING

UNIT 4

CLOUD DEPLOYMENT ENVIRONMENT

SYLLABUS: Google App Engine – Amazon AWS – Microsoft Azure; Cloud Software Environments – Eucalyptus – OpenStack

PART A

1. **Summarize the service offerings by AWS.** **Nov 2023**
   - Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing.
2. **Depict the benefits of OpenStack Compute.** **Nov 2023**
   - Shorter time-to-market. With OpenStack, you do not have to wait for the networking services you need.
   - Faster innovation.
   - Comply with regulations easier.
   - Need not commit to a single vendor.
3. **What is Google app Engine?**
   - Google App Engine (GAE) is a platform-as-a-service product that provides web app developers and enterprises with access to Google's scalable hosting.
4. **What are the advantages of Google App Engine?**
   - Lower total cost of ownership
   - Fully featured SDK for local development
   - Ease of Deployment
5. **What are the benefits of Google App Engine? Benefits of GAE.**
   - Ease of setup and use.
   - Pay-per-use pricing.

- Scalability.
- Security.

6. What are the challenges of Google App Engine?

- Lack of control.
- Performance limits.
- Limited access.

7. What are the service provided by Google App Engine?

- Wide range of service available
- User service
- Task Queues
- Mail Service
- Image

8. What are the different ways of sorting application data in Google App Engine?

- Datastore
- Google Cloud SQL
- Google Cloud Storage

9. List some of the restrictions in Google App Engine.

- Read only access to file system
- Application cannot create new threads
- 10 MB request and response size limit
- 30 sec deadline for every request/response

10. What are the components of Google App Engine?

- SDK
- Language Runtime
- Web based Admin Console
- Scalable Infrastructure

11. What is Amazon Web Service (AWS)?

- Amazon web services is a collection of remote computing services (web services) that together make up a cloud computing platform offered over the internet by Amazon.com

12. What does Amazon Web Service offering?

- Low ongoing cost
- Elasticity and Flexible Capacity.
- Speed and Agility
- Open and Flexible

- Secure

13. What is Amazon Elastic Compute Cloud(EC2)?

- A web service that provides resizable compute capacity in the cloud EC2 allows creating virtual machine on-demand.

14. What are the three types of AMI? Image Type.

- Private AMI

  Images created by you, which are private by default. You can grant access to other users to launch your private images.
- Public AMI

  Images created by users and released to the AWS community, so anyone can launch instances based on them and use them any way they like.
- Paid QAMI

  Can create images providing specific functions that can be launched by anyone willing to pay you per each hour of usage on top of Amazon's charges.

15. What are the cloud services provided by Azure Platform?

- Live service - Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently.
- .NET service - This package supports application development on local hosts and execution on cloud machines.
- SQL Azure - This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud.
- SharePoint service - This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services.
- Dynamic CRM service - This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions.

16. What is Amazon Elastic Block Store(EBS)?

EBS provides block level storage volumes (1GB to 1TB) for use with Amazon EC2:

- Multiple Volumes can be mounted to the same instance
- EBS volumes are network attached and persist independently from the life of an instance.

17. What is Amazon Simple Storage Service(S3)?

- Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.

18. What is Amazon Elastic Map Reduce(EMR)?

- Amazon EMR is a web service that makes it easy to quickly and cost effectively process vast amounts of data using Hadoop. Amazon EMR distribute the data and processing across a resizable cluster of Amazon EC2 instances.

19. What is Amazon Relational Database Service(RDS)?

- Amazon RDS is a web service that makes it easy to set up, operate and scale a relational database in the cloud. It gives access to the capabilities of a familiar MySQL, Oracle or Microsoft SQL, Server database engine.

20. What is Amazon Dynamo DB?

- DynamoDB is a fast, fully managed NoSQL database service that makes it simple and cost-effective to store and retrieve any amount of data and serve any level of request traffic.

21. What is Eucalyptus?

- Eucalyptus is an open source software platform for implementing infrastructure as a service(IaaS) in a private or hybrid cloud computing environment.
- Eucalyptus is an acronym for Elastic Utility Computing Architecture for linking your Programs to useful systems.

22. List the features of Eucalyptus.

- Supports both Linux and windows virtual machines (VMs)
- Application Program interface-(API) compatible with Amazon EC2 platform
- Compatible with Amazon Web service(AWS) and simple storage(S3)
- Multiple clusters cab virtualized as a single cloud.

23. What are the components of Eucalyptus?

- Cluster Controller
- Cloud Controller
- Node Controller
- Storage Controller

24. What is Open Stack?

- Openstack is a free and open-source platform for cloud computing, mostly deployed as infrastructure as a service(IaaS), whereby virtual servers and other resources are made available to customers.

25. What is meant by Nimbus?

- Nimbus is a set of open source tools that together provide an IaaS cloud computing solution. Nimbus provides a special web interface known as Nimbus Web.
- Its aim is to provide administrative and user functions in a friendly interface.

26. What is cloud software environments?

- In a cloud environment, consumers can deploy and run their software applications on a sophisticated infrastructure that is owned and managed by a cloud provider (eg, Amazon Web Services, Microsoft Azure, and Google Cloud Platform).

27. What is Microsoft Azure?

- Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a broad range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications or run existing applications in the public cloud.

Part-B

1. What is GAE? Discuss in detail about the Google App Engine (GAE) ,its architecture and functional modules.

What is Google App Engine? Describe the major building blocks and functional modules of the Google Cloud Platform with a diagram. Nov 2023

- Google App Engine GAE
- Google Cloud Infrastructure
- GAE's key features
- GAE Architecture
- Functional Modules of GAE
- Google App Engine benefits and challenges
- GAE Applications

➢ **Google App Engine GAE**

- Google App Engine (GAE) is a platform-as-a-service product that provides web app developers and enterprises with access to Google's scalable hosting.
- GAE requires that applications be written in Java or Python, store data in Google Bigtable and use the Google query language.
- Google provides GAE free up to a certain amount of use for the following resources:
  - processor (CPU)
  - storage
  - application programming interface (API) calls
  - concurrent requests

➢ **Google Cloud Infrastructure**

- GAE web application platform for many small cloud service providers in supporting scalable (elastic) web applications.
- GAE enables users to run their applications on a large number of data centers associated with Google's search engine operations.

➢ **GAE's key features**

- **API selection**.

  GAE has several built-in APIs, including the following five:
  - **Blobstore** for serving large data objects;
  - **GAE Cloud Storage** for storing data objects;

- **Page Speed Service** for automatically speeding up webpage load times;
- **URL Fetch Service** to issue HTTP requests and receive responses for efficiency and scaling;
- **Memcache** for a fully managed in-memory data store.

- **Managed infrastructure.** Google manages the back-end infrastructure for users. This approach makes GAE a serverless platform and simplifies API management.
- **Several programming languages**. GAE supports a number of languages, including GO, PHP, Java, Python, NodeJS, .NET and Ruby.
- **Support for legacy runtimes.** GAE supports legacy runtimes, which are versions of programming languages no longer maintained.
- **Application diagnostics**. GAE lets users record data and run diagnostics on applications to gauge performance.
- **Security features.** GAE enables users to define access policies with the GAE firewall and managed Secure Sockets Layer/Transport Layer Security certificates for free.
- **Traffic splitting.** GAE lets users route requests to different application versions.
- **Versioning.** Every time code is deployed to a service with the corresponding GAE configuration files, a version of that service is created.

➢ **GAE Architecture**

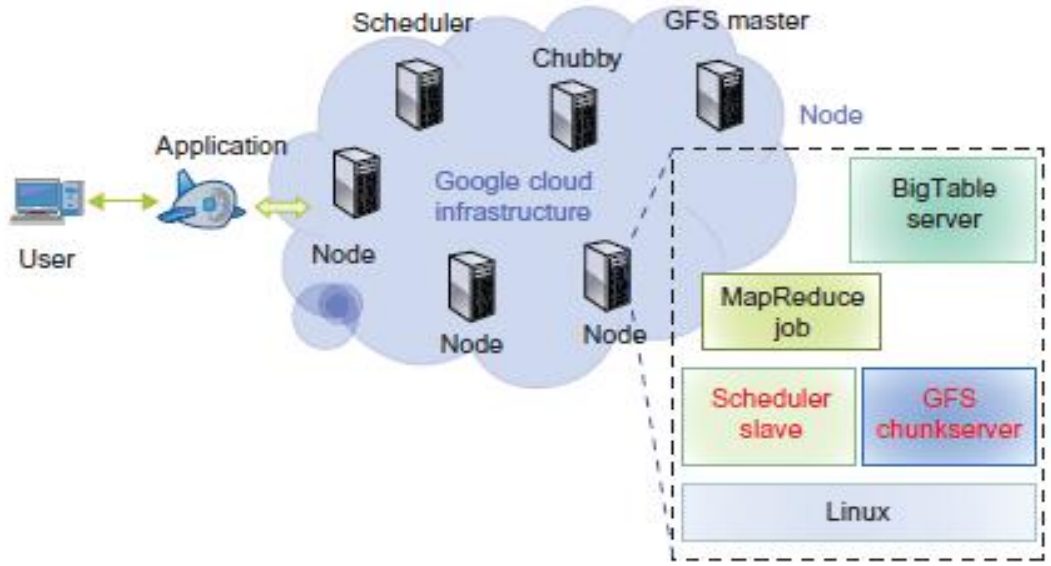


Figure 4.1 – Google's cloud Platform

- The building blocks of Google's cloud computing application as shown in figure 4.1 includes
  - **Google File System** for storing large amounts of data,
  - **MapReduce** programming framework for application developers,
  - **Chubby** for distributed application lock services,
  - **BigTable** as a storage service for accessing structural or semi structural data.
- With these building blocks, Google has built many cloud applications.
- **Third-party application providers** can use GAE to build cloud applications for providing services.

➢ **Functional Modules of GAE**

- The **datastore** offers object-oriented, distributed, structured data storage services and secures data management operations.
- The **application runtime environment** offers a platform for scalable web programming and execution. It supports two development languages: Python and Java.
- The **software development kit (SDK)** is used for local application development. The SDK allows users to execute test runs of local applications and upload application code.
- The **administration console** is used for easy management of user application development cycles.
- The **GAE web service infrastructure** provides special interfaces to guarantee flexible use and management of storage and network resources by GAE.

➢ **Google App Engine benefits and challenges**

Benefits of GAE

- Ease of setup and use.
- Pay-per-use pricing.
- Scalability.
- Security.

GAE challenges

- Lack of control.
- Performance limits.
- Limited access.

➢ **GAE Applications**

GAE supports many web applications.

- A storage service to store application-specific data in the Google infrastructure.
- GAE also provides Google-specific services, such as the Gmail account service.
- Thus, web applications built on top of GAE can use the APIs authenticating users and sending e-mail using Google accounts.

2. Explain in detail about Google File System (GFS) and its architecture.

- GFS was built primarily as the fundamental storage service for Google's search engine.
- GFS was designed for Google applications, and Google applications were built for GFS.

  GFS typically will hold a large number of huge files, each 100MB or larger, with files that are multiple GB in size.
- Google has its file data block size to be 64MB.
- Reliability is achieved by using replications.

Architecture of Google File System (GFS)

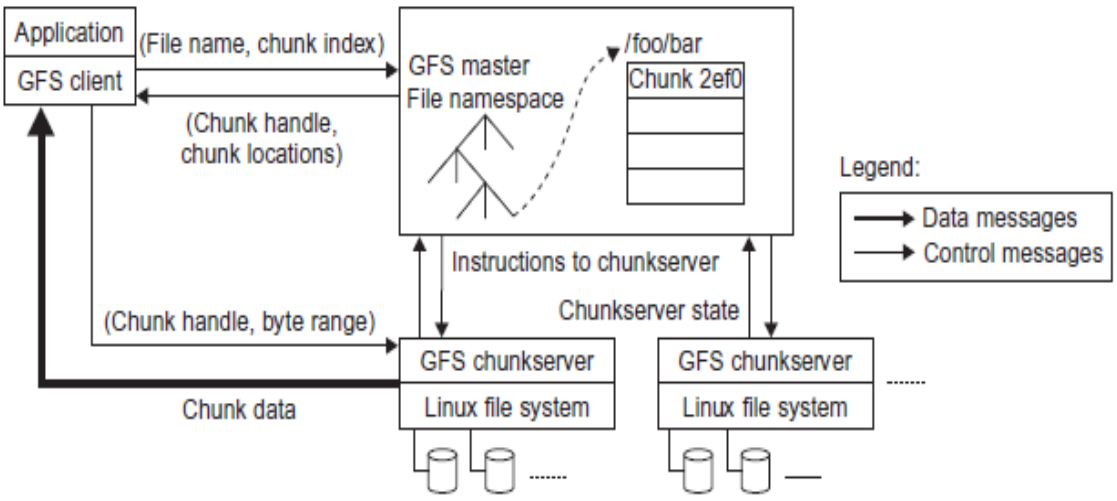


Figure 4.2 – Google File System (GFS)

- Figure 4.2 shows the GFS architecture.
- There is a single master in the whole cluster.

- Other nodes act as the chunk servers for storing data, while the single master stores the metadata.
- The file system namespace and locking facilities are managed by the master.
- The master periodically communicates with the chunk servers to collect management information as well as give instructions to the chunk servers to do work such as load balancing or fail recovery.
- The master has enough information to keep the whole cluster in a healthy state.
- The single GFS master could be the performance bottleneck and the single point of failure.
- To mitigate this,
- Google uses a shadow master to replicate all the data on the master, and the design guarantees that all the data operations are performed directly between the client and the chunk server.
- The control messages are transferred between the master and the clients and they can be cached for future use.
- The single master can handle a cluster of more than 1,000 nodes.

Data Mutation (Write, Append Operations) In GFS

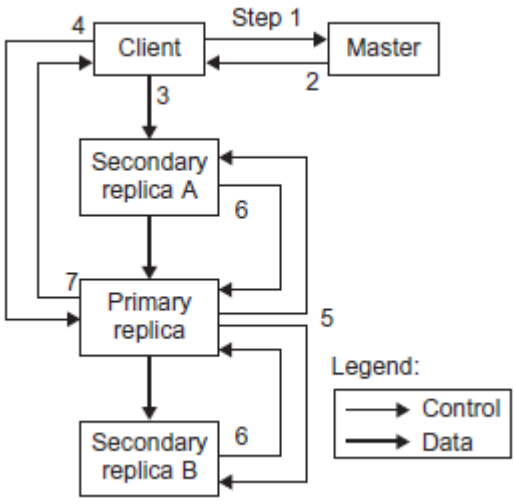


Figure 4.3 – Data Mutation in GFS

The mutation takes the following steps as shown in figure 4.3:

1. The client asks the master which chunk server holds the current lease for the chunk and the locations of the other replicas. If no one has a lease, the master grants one to a replica it chooses
2. The master replies with the identity of the primary and the locations of the other (secondary) replicas.
3. The client pushes the data to all the replicas. Each chunk server will store the data in an internal LRU buffer cache until the data is used or aged out.
4. Once all the replicas have acknowledged receiving the data, the client sends a write request to the primary. The primary assigns Consecutive serial numbers to all the mutations it receives, possibly from multiple clients, which provides the necessary serialization.
5. The primary forwards the write request to all secondary replicas. Each secondary replica applies mutations in the same serial number order assigned by the primary.
6. The secondary's all reply to the primary indicating that they have completed the operation.
7. The primary replies to the client. Any errors encountered at any replicas are reported to the client. Our client code handles such errors by retrying the failed mutation.

3. Discuss in detail about BigTable, Google's NOSQL System.

- BigTable was designed to provide a service for storing and retrieving structured and semi structured data.
- BigTable applications include storage of web pages, per-user data, and geographic locations.
- BigTable can be viewed as a distributed multilevel map.
- It provides a fault-tolerant and persistent database as in a storage service.
- The BigTable system is scalable, which means the system has thousands of servers, terabytes of in-memory data, petabytes of disk-based data, millions of reads/writes per second, and efficient scans.
- Also, BigTable is a self-managing system (i.e., servers can be added/removed dynamically and it features automatic load balancing).

- The BigTable system is built on top of an existing Google cloud infrastructure. BigTable uses the following building blocks:
  1. GFS: stores persistent state
  2. Scheduler: schedules jobs involved in BigTable serving
  3. Lock service: master election, location bootstrapping
  4. MapReduce: often used to read/write BigTable data

BigTable system structure

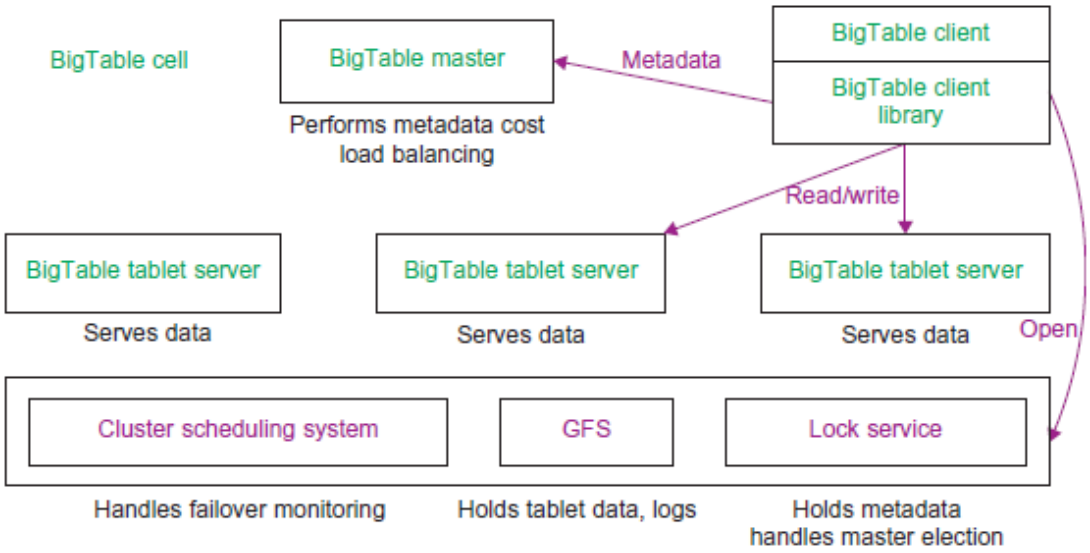


Figure 4.4 – Big Table Structure

- Figure 4.4 shows the BigTable system structure.
- A BigTable master manages and stores the metadata of the BigTable system.
- BigTable clients use the BigTable client programming library to communicate with the BigTable master and tablet servers.
- BigTable relies on a highly available and persistent distributed lock service called Chubby.

Tablet Location Hierarchy

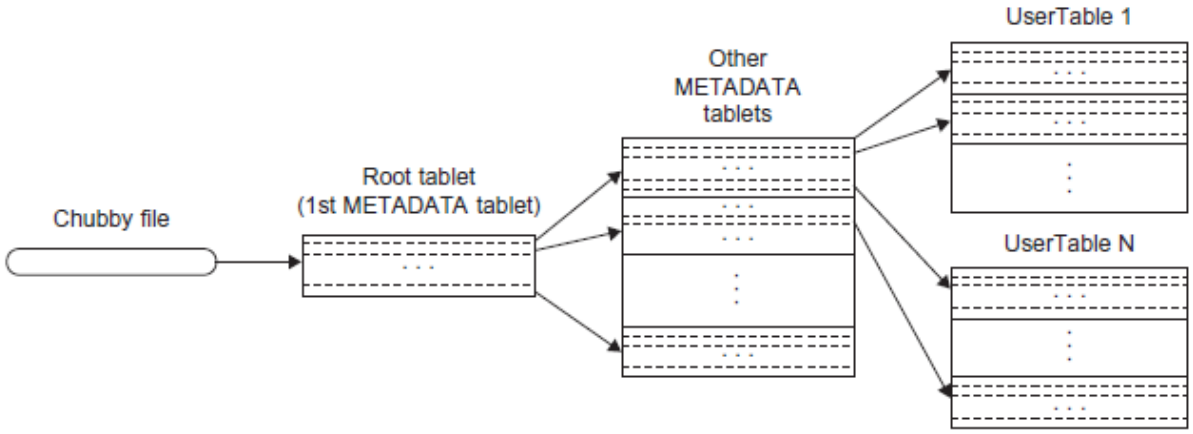


Figure 4.5 – Tablet location hierarchy in using the BigTable.

- The first level is a file stored in Chubby that contains the location of the root tablet.
- The root tablet contains the location of all tablets in a special METADATA table.
- Each METADATA tablet contains the location of a set of user tablets.
- The root tablet is just the first tablet in the METADATA table, but is treated specially; it is never split to ensure that the tablet location hierarchy has no more than three levels.
- The METADATA table stores the location of a tablet under a row key that is an encoding of the tablet's table identifier and its end row.
- The BigTable master can quickly scan the tablet servers to determine the status of all nodes.

Chubby, Google's Distributed Lock Service

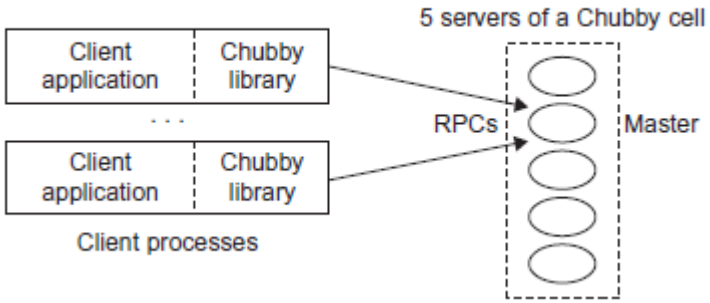


Figure 4.6 – Overall architecture of the Chubby system.

- Chubby is intended to provide a coarse-grained locking service.
- It can store small files inside Chubby storage which provides a simple namespace as a file system tree.
- The files stored in Chubby are quite small compared to the huge files in GFS.
- Figure 4.6 shows the overall architecture of the Chubby system.
- Each Chubby cell has five servers inside.
- Each server in the cell has the same file system namespace.
- Clients use the Chubby library to talk to the servers in the cell.
- Client applications can perform various file operations on any server in the Chubby cell.
- Chubby has become Google's primary internal name service.
- GFS and BigTable use Chubby to elect a primary from redundant replicas.

4. Describe the installation of the Google App Engine Software Development Kit (SDK) on a Microsoft Windows and running a simple "hello world" application.

Procedure:

1. Go to the Google App Engine official website (Refer Figure 4.7)

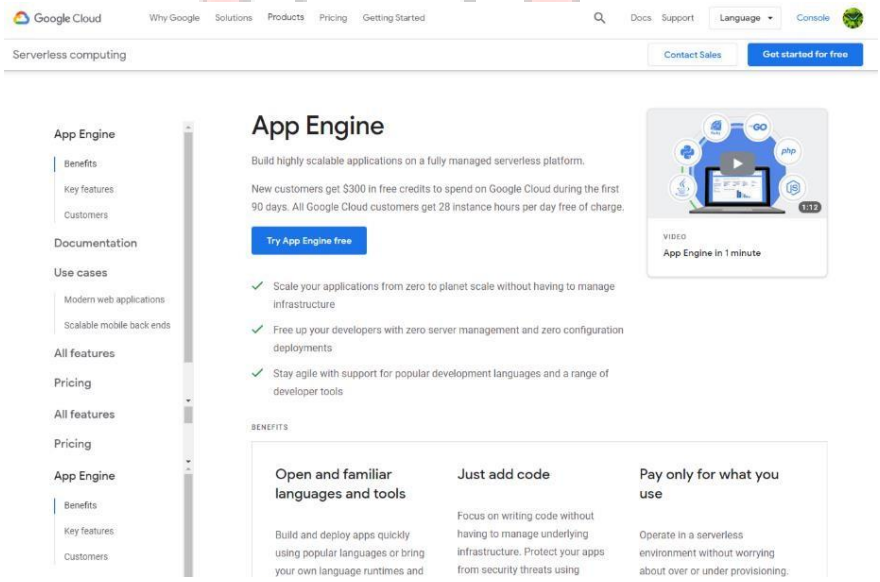


Figure 4.7 – Google App Engine official website

2. Login to official Gmail account by user name and password.
3. Click console for go to the project sight.
4. Create a new project “My Project” as shown in Figure 4.8.

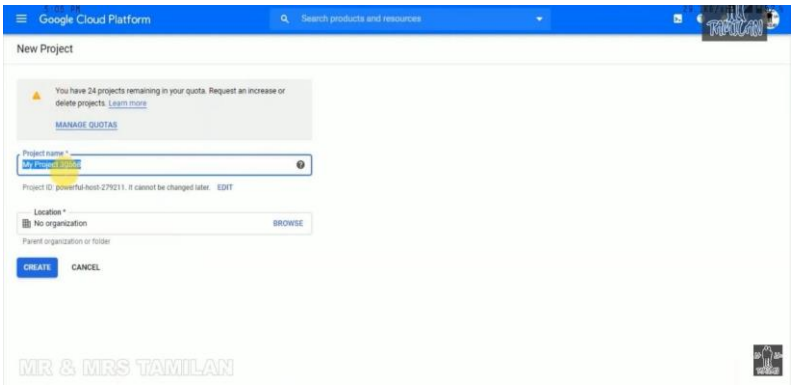


Figure 4.8 – My Project

5. The project was created in name “My Project”.
6. Click “create application” for develop an application as shown in Figure 4.9.

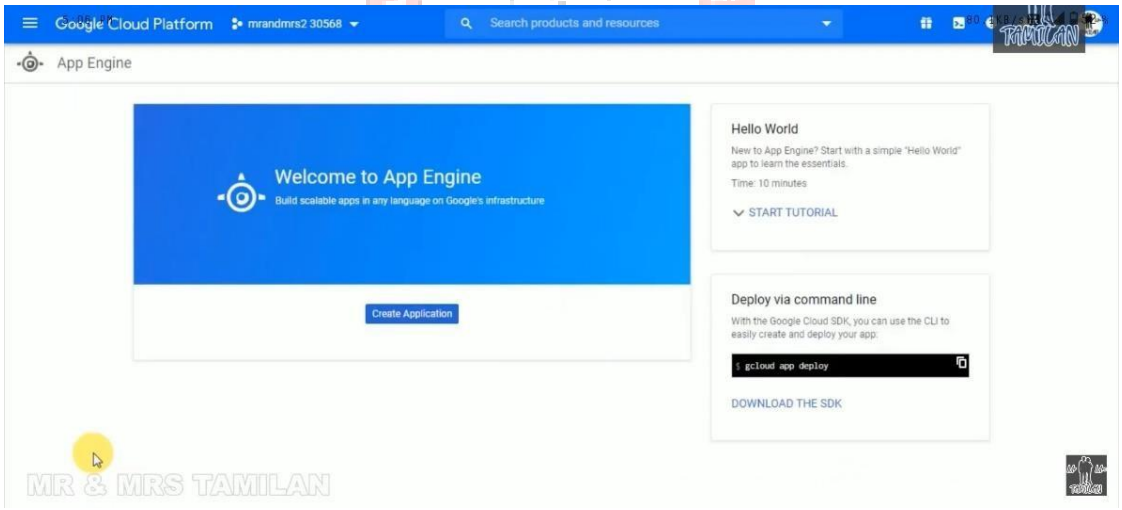


Figure 4.9 – create application

7. Select a programming language as per need.
8. Select python programming language for creating the “hello world” Program as shown in Figure 4.10.

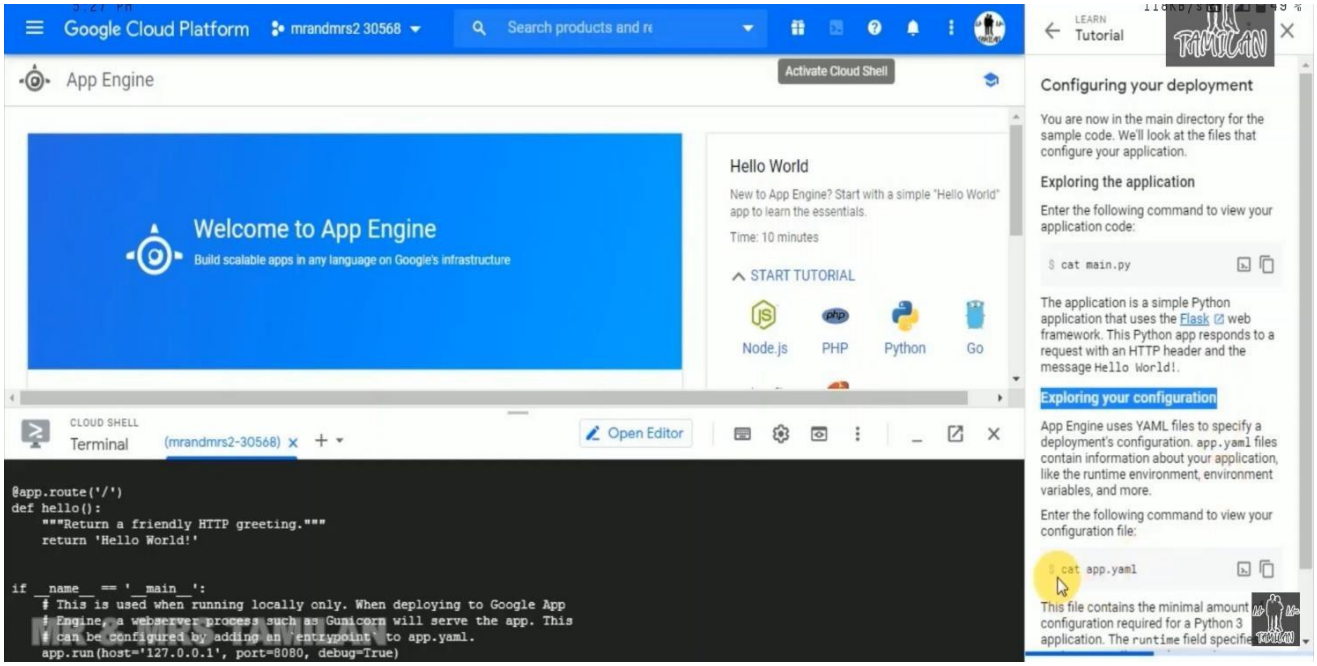


Figure 4.10 – Hello World Program

9. Type the python coding for displaying "Hello World" by using the GAE tools.
10. Deploy the application to registered account without any errors.
11. Save the python program file as "cat_main.py" and explore the configuration.
12. Create an .yml file for executing the program in the terminal.
13. Edit your application by the editor.
14. To run the application by clicking "preview on port 8080" as including in the terminal topic in Figure 4.11.

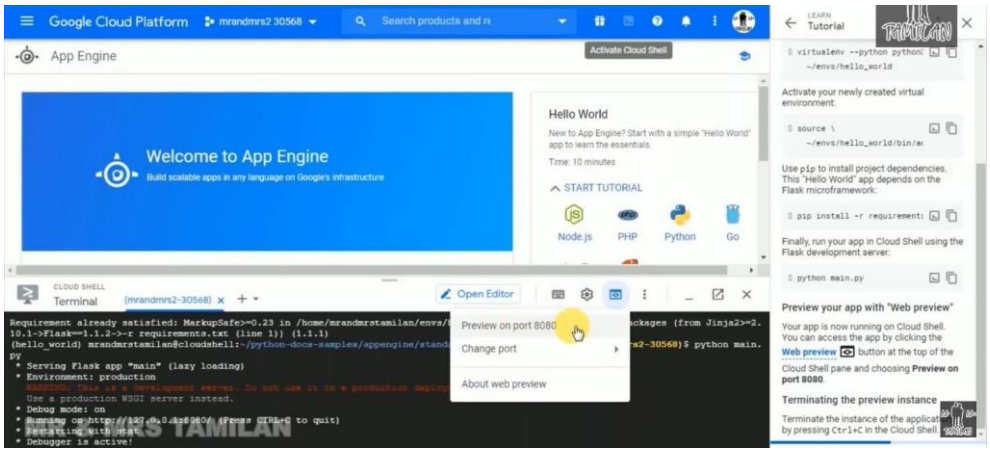


Figure 4.11 – Preview on port 8080

15. The output will be displayed as "Hello World" in a runtime window as shown in Figure 4.12

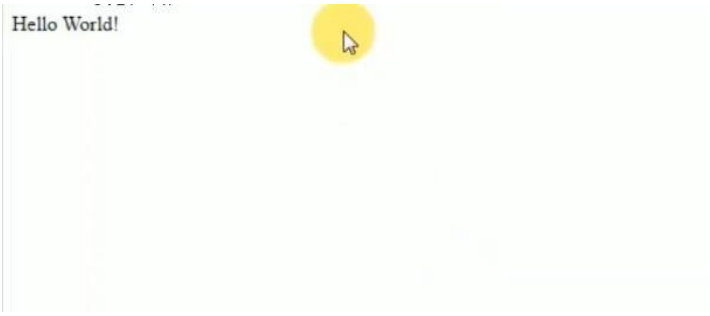


Figure 4.12 – Hello World Screen

5. Describe in detail about Amazon Web Services (AWS) or Amazon AWS.

- AWS
- AWS architecture
- AWS Services Offered

➢ **AWS**

- AWS stands for Amazon Web Services.
- The AWS service is provided by the Amazon that uses distributed IT infrastructure to provide different IT resources available on demand.
- It provides different services such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS).
- AWS services can offer an organization tools such as compute power, database storage and content delivery services.

➢ **AWS architecture.**

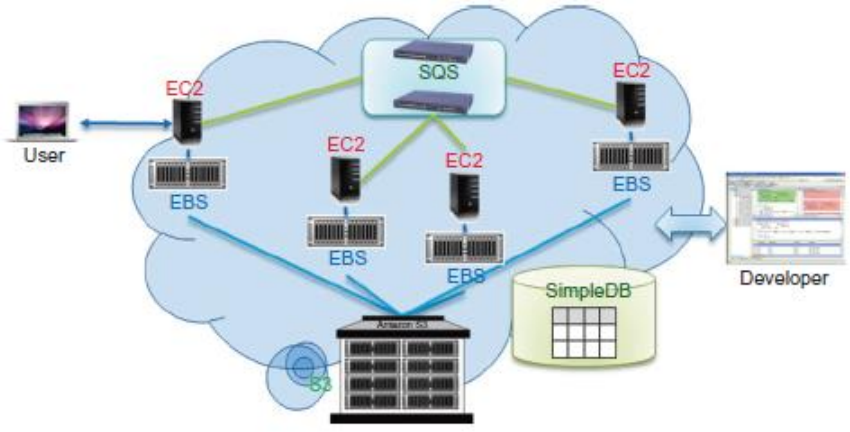


Figure 4.13 – Amazon cloud computing infrastructure

- Figure 4.13 shows the AWS architecture.
- EC2 provides the virtualized platforms to the host VMs where the cloud application can run.
- S3 (Simple Storage Service) provides the object-oriented storage service for users.
- EBS (Elastic Block Service) provides the block storage interface which can be used to support traditional applications.
- SQS stands for Simple Queue Service, and its job is to ensure a reliable message service between two processes.
- Users can access their objects through SOAP with either browsers or other client programs which support the SOAP standard.

➢ **AWS Services Offered**

Service Area	Service Modules and Abbreviated Names
o Compute	Elastic Compute Cloud (EC2), Elastic MapReduce,
o Messaging	Simple Queue Service (SQS), Simple Notification Service (SNS) Storage Simple Storage Service (S3), Elastic Block Storage (EBS), AWS Import/Export
o Content Delivery	Amazon CloudFront
o Monitoring	Amazon CloudWatch
o Support	AWS Premium Support
o Database	Amazon SimpleDB, Relational Database Service (RDS)
o Networking	Virtual Private Cloud (VPC) , Elastic Load Balancing
o Web Traffic	Alexa Web Information Service, Alexa Web Sites
o E-Commerce	Fulfillment Web Service (FWS)
o Payments and Billing	Flexible Payments Service (FPS), Amazon DevPay
o Workforce	Amazon Mechanical Turk

Programming on Amazon EC2

- Amazon was the first company to introduce VMs in application hosting.

- Customers can rent VMs instead of physical machines to run their own applications.
- By using VMs, customers can load any software of their choice.
- The elastic feature of such a service is that a customer can create, launch, and terminate server instances as needed, paying by the hour for active servers.
- Instances are often called Amazon Machine Images (AMIs) which are preconfigured with operating systems based on Linux or Windows, and additional software.
- AMIs are the templates for instances, which are running VMs.
- The workflow to create a VM is

  Create an AMI→Create Key Pair→Configure Firewall→Launch
- This sequence is supported by public, private, and paid AMIs

Three Types of AMI

Image Type	AMI Definition
Private AMI	Images created by you, which are private by default. You can grant access to other users to launch your private images.
Public AMI	Images created by users and released to the AWS community, so anyone can launch instances based on them and use them any way they like.
Paid QAMI	Can create images providing specific functions that can be launched by anyone willing to pay you per each hour of usage on top of Amazon's charges.

6. Describe in detail about Microsoft Azure.

- Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform.
- It provides a broad range of cloud services, including compute, analytics, storage and networking.
- Microsoft launched a Windows Azure platform to meet the challenges in cloud computing.
- This platform is built over Microsoft data centers.
- The platform is divided into three major component platforms as shown in figure 4.14.

- Windows Azure offers a cloud platform built on Windows OS and based on Microsoft virtualization technology.
- Applications are installed on VMs deployed on the data-center servers.
- Azure manages all servers, storage, and network resources of the data center.
- On top of the infrastructure are the various services for building different cloud applications.
- Cloud-level services provided by the Azure platform are introduced below.

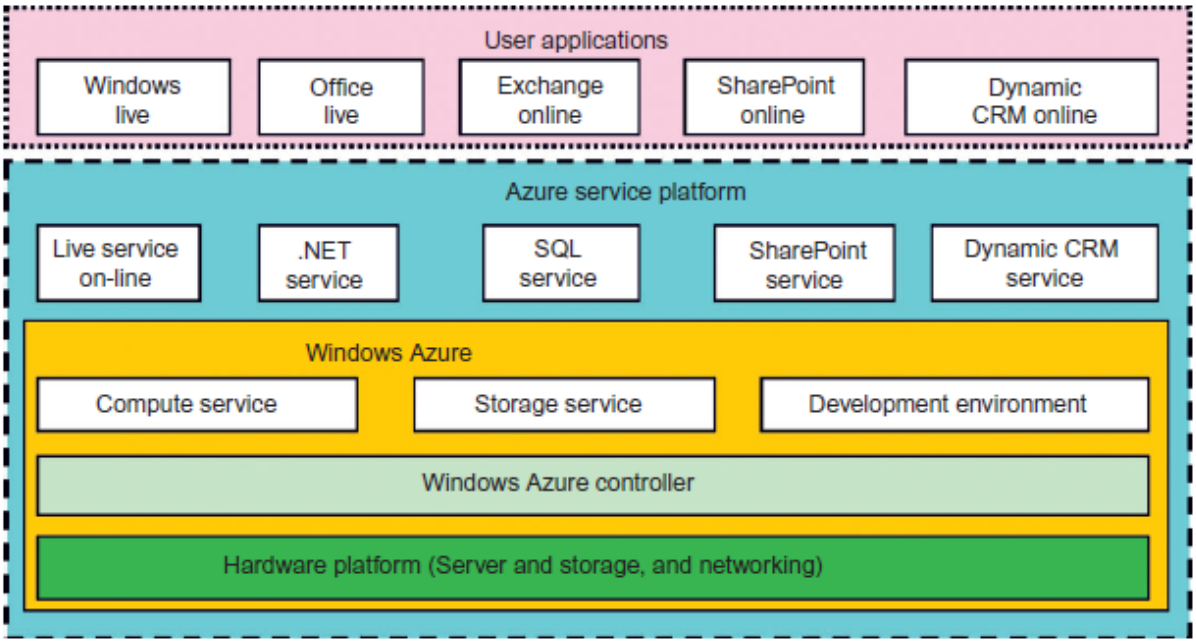


Figure 4.14 – Amazon cloud computing infrastructure

Cloud-level services provided by the Azure platform are introduced below.

- **Live service** - Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently.
- **.NET service** - This package supports application development on local hosts and execution on cloud machines.
- **SQL Azure** - This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud.
- **SharePoint service** - This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services.
- **Dynamic CRM service** - This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions.

- The Azure platform applies the standard web communication protocols SOAP and REST.
- The Azure service applications allow users to integrate the cloud application with other platforms or third-party clouds.

7. Discuss in detail about Emerging Cloud Software Environment like Open Source Eucalyptus and Nimbus.

What is Eucalyptus? Discuss its architecture by mentioning their role in filtering incoming traffic. Nov 2023

Open Source Eucalyptus and Nimbus

- Eucalyptus is a product from Eucalyptus Systems that was developed out of a research project at the University of California, Santa Barbara.
- Eucalyptus was initially aimed at bringing the cloud computing paradigm to academic supercomputers and clusters.
- Eucalyptus provides an AWS-compliant EC2-based web service interface for interacting with the cloud service.
- Eucalyptus provides services, such as the AWS-compliant Walrus, and a user interface for managing users and images.

Eucalyptus Architecture

- The Eucalyptus system is an open software environment.
- The Eucalyptus architecture for VM image management architecture was presented in a Figure 4.15

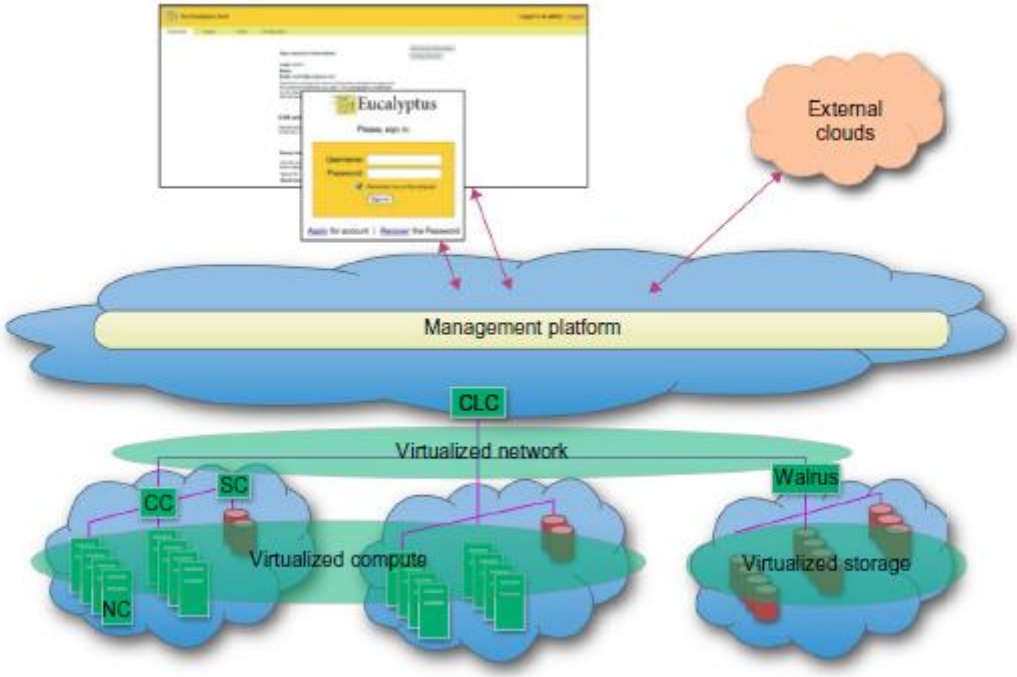


Figure 4.15 The Eucalyptus architecture for VM image management

- The system supports cloud programmers in VM image management.
- The system has been extended to support the development of both the computer cloud and storage cloud.

VM Image Management

- Eucalyptus stores images in Walrus, the block storage system.
- Any user can bundle her own root file system, and upload and then register this image and link it with a particular kernel and ramdisk image.
- This image is uploaded into a user-defined bucket within Walrus, and can be retrieved anytime from any availability zone.
- This allows users to create specialty virtual appliances and deploy them within Eucalyptus with ease.
- The Eucalyptus system is available in a commercial proprietary version, as well as the open source version.

Nimbus

- Nimbus is a set of open source tools that together provide an IaaS cloud computing solution.

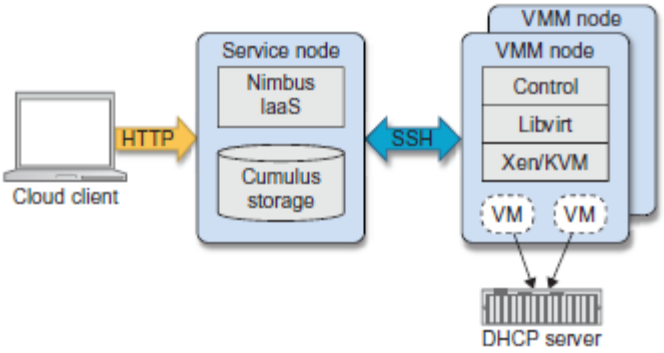


Figure 4.16 The Nimbus architecture

- Figure 4.16 shows the architecture of Nimbus, which allows a client to lease remote resources by deploying VMs on those resources and configuring them to represent the environment desired by the user.
- Nimbus provides a special web interface known as Nimbus Web.
- Its aim is to provide administrative and user functions in a friendly interface.
- Nimbus Web is centered around a Python Django web application that is intended to be deployable completely separate from the Nimbus service.
- A storage cloud implementation called Cumulus has been tightly integrated with the other central services, although it can also be used stand-alone.
- The Nimbus cloud client uses the Java Jets3t library to interact with Cumulus.
- Nimbus supports two resource management strategies.
  - The first is the default "resource pool" mode. In this mode, the service has direct control of a pool of VM manager nodes and it assumes it can start VMs.
  - The other supported mode is called "pilot." Here, the service makes requests to a cluster's Local Resource Management System (LRMS) to get a VM manager available to deploy VMs.
- Nimbus also provides an implementation of Amazon's EC2 interface that allows users to use clients developed for the real EC2 system against Nimbus-based clouds.

9 Discuss in detail about Emerging Cloud Software Environment like Open Stack.

OpenStack

- OpenStack was been introduced by Rackspace and NASA in July 2010.
- OpenStack focuses on the development of two aspects of cloud computing to address compute and storage aspects with the OpenStack Compute and OpenStack Storage solutions.
- "OpenStack Compute is the internal fabric of the cloud creating and managing large groups of virtual private servers"
- "OpenStack Object Storage is software for creating redundant, scalable object storage using clusters of commodity servers to store terabytes or even petabytes of data."
- The image repository contains an image registration and discovery service and an image delivery service.
- Together they deliver images to the compute service while obtaining them from the storage service.

OpenStack Compute

- OpenStack is developing a cloud computing fabric controller, a component of an IaaS system, known as Nova.
- The architecture for Nova is built on the concepts of shared-nothing and messaging-based information exchange.
- To achieve the shared-nothing paradigm, the overall system state is kept in a distributed data system.
- Nova is implemented in Python while utilizing a number of externally supported libraries and components. This includes boto, an Amazon API provided in Python, and Tornado, a fast HTTP server used to implement the S3 capabilities in OpenStack.
- Figure 4.17 shows the main architecture of Open Stack Compute.

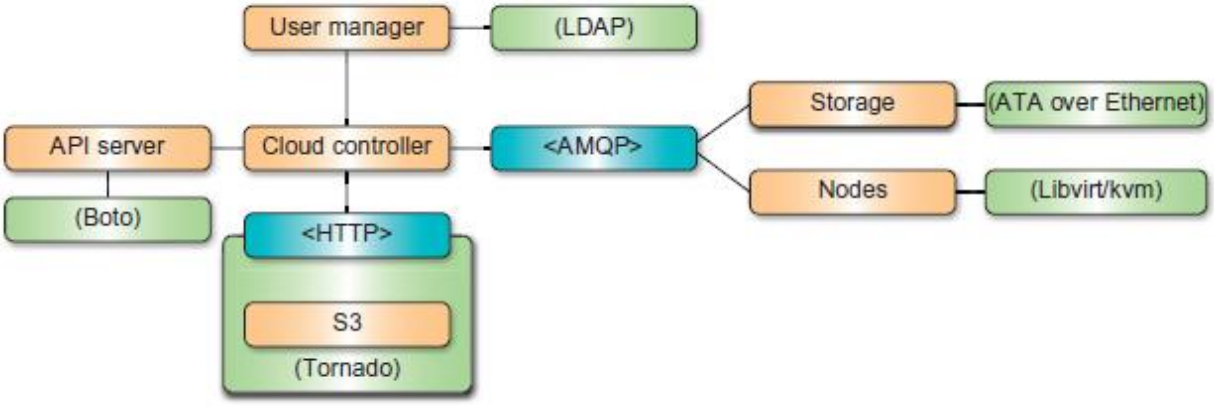


Figure 4.17 - OpenStack Nova system architecture

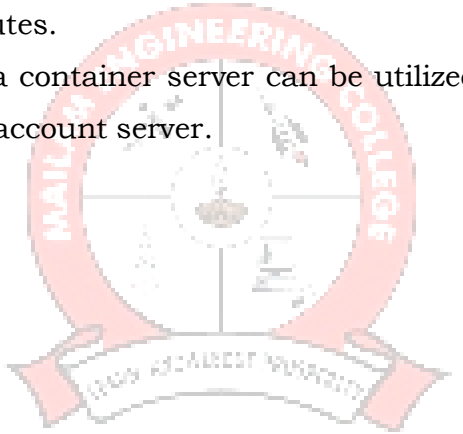
- In this architecture, the API Server receives HTTP requests from boto, converts the commands to and from the API format, and forwards the requests to the cloud controller.
  The cloud controller maintains the global state of the system, ensures authorization while interacting with the User Manager via Lightweight Directory Access Protocol (LDAP), interacts with the S3 service, and manages nodes, as well as storage workers through a queue.
- Nova integrates networking components to manage private networks, public IP addressing, virtual private network (VPN) connectivity, and firewall rules.

- It includes the following types:
  - Network Controller manages address and virtual LAN (VLAN) allocations.
  - Routing Node governs the NAT (network address translation) conversion of public IPs to private IPs, and enforces firewall rules
  - Addressing Node runs Dynamic Host Configuration Protocol (DHCP) services for private networks
  - Tunneling Node provides VPN connectivity
- The network state consists of the following:
  - VLAN assignment to a project
  - Private subnet assignment to a security group in a VLAN
  - Private IP assignments to running instances

- Public IP allocations to a project
- Public IP associations to a private IP/running instance

OpenStack Storage

- The OpenStack storage solution is built around a number of interacting components and concepts, including a proxy server, a ring, an object server, a container server, an account server, replication, updaters, and auditors.
- The role of the proxy server is to enable lookups to the accounts, containers, or objects in OpenStack storage rings and route the requests.
- A ring represents a mapping between the names of entities stored on disk and their physical locations.
- Separate rings for accounts, containers, and objects exist.
- Objects are stored as binary files with metadata stored in the file's extended attributes.
- To list objects, a container server can be utilized. Listing of containers is handled by the account server.



MAILAM Engineering College

Approved by AICTE, New Delhi, Permanently Affiliated to Anna University
Chennai,Accredited by NBA, NAAC with A Grade and TCS)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

III YEAR / V SEM

CCS335 CLOUD COMPUTING

UNIT 5

CLOUD SECURITY

SYLLABUS:

Virtualization System-Specific Attacks: Guest hopping – VM migration attack – hyper jacking. Data Security and Storage; Identity and Access Management (IAM) - IAM Challenges - IAM Architecture and Practice.

PART A

1. **List out the IAM challenges.** **Nov 2023**
   - Lack of centralized view.
   - Difficulties in User Lifecycle Management.
   - Keeping Application Integrations Updated.
   - Compliance Visibility into Third Party SaaS Tools.
2. **What is meant by Hyperjacking?** **Nov 2023**
   - Hyperjacking takes control of the hypervisor to gain access to the VMs and their data. It is typically launched against type 2 hypervisors that run over a host OS.
3. **What is meant by Virtualization?**
   - Virtualization is technology that you can use to create virtual representations of servers, storage, networks, and other physical machines Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine.
4. **State the types of VM migration**

   Non-Live VM Migration/Cold Migration

- Non-live migration is a VM migration procedure in which the VM must be shut down in the host machine before transferring VM from the host computer to the destination computer.

Live VM Migration/Hot Migration

- Live VM migration is a VM migration procedure for transferring virtual machines (VMs) from one machine to another without causing any downtime. No need to shutdown host machine.

5. State the Data security threats in cloud computing.

- Unsecure application programming interfaces (APIs)
- Account hijacking or takeover
- Insider threats

6. What are the Benefits of Cloud Data Security?

- Mitigate Data Breach Risk
- Protect Brand Reputation
- Enhance Customer Trust
- Avoid Fines and Fees

7. What is meant by Identity and Access Management (IAM)?

- The Identity and access management is the security framework composed of policy and governance components used for creation, maintenance and termination of digital identities with controlled access of shared resources.

8. What is meant by Authentication and Authorization?

Authentication

- Authentication is the process of verifying the identity of a user or system.

Authorization

- Authorization is the process of determining the privileges the user or system is entitled to once the identity is established.

9. What are the various challenges of IAM?

- Identity Provisioning / De-provisioning
- Maintaining a single ID across multiple platforms and organizations
- Security when using 3rd party or vendor network
- Lack of centralized view
- Keeping Application Integrations Up to Date

10. What is meant by guest hopping attack?

- In this type of attack, an attacker will try to get access to one virtual machine by penetrating another virtual machine hosted in the same hardware. One of the possible mitigations of guest hopping attack is the Forensics and VM debugging tools to observe the security of cloud.

11. What is meant by VM migration attack?

- Migration works by sending the state of the guest virtual machine's memory and any virtualized devices to a destination host physical machine. Live Migration has many security vulnerabilities. The security threats could be on the data plane, control plane and migration plane.

12. What are the different types of VM migration attack?

- Denial of Service (DoS) Attacks.
- Co-Residential Attacks.
- Cache-Based Side Channel Attacks.

13. What is meant by data security?

- Cloud data security is the practice of protecting data and other digital information assets from security threats, human error, and insider threats. It leverages technology, policies, and processes to keep your

data confidential and still accessible to those who need it in cloud-based environments.

14. What are the four 4 elements of data security?

- Protection
- Detection
- Verification
- Reaction.

15. What are the two types of attacks in man –in –the-middle attack?

- In active attacks, the attacker intercepts the connection and efforts to modify the message's content. In passive attacks, the attacker observes the messages, then copy and save them and can use it for malicious purposes

16. State application of IAM.

- Identity and access management (IAM) ensures that the right people and job roles in your organization (identities) can access the tools they need to do their jobs. Identity management and access systems enable your organization to manage employee apps without logging into each app as an administrator.

17. What are the 4 components of IAM?

AM components can be classified into four major categories:

- Authentication
- Authorization
- user management
- central user repository.

18. What is meant by Identity Authentication?

- Authentication is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials.

19. What are the three 3 common identification and authentication methods?

The three authentication factors are:

- Knowledge Factor – something you know, e.g., password.
- Possession Factor – something you have, e.g., mobile phone.
- Inherence Factor – something you are, e.g., fingerprint.

20. What are the common types of Authorization?

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Role-Based Access Control (RBAC)

21. What is meant by user's access keys in IAM?

- Passwords are used to encrypt or hide files. If you have many passwords, you can store all of them in a list, open with one "master password". That master password is called Access Key (to a list of passwords).

22. What is an identity directory service?

- A directory service is a database for storing and maintaining information about users and resources.
- Directory Services are often referred to as directories, user stores, Identity Stores, or LDAP Directory, and they store information such as usernames, passwords, user preferences, information about devices, and more.

23. How do you monitor user activity with IAM?

- Turn on AWS CloudTrail in each account, and use it in each supported Region.

- Store AWS CloudTrail log in a centralized logging account with very restricted access.
- Periodically examine CloudTrail log files.
- Enable Amazon S3 bucket logging to monitor requests made to each bucket.

24. What are the IAM processes operational activities?

- Provisioning
- Credential and Attribute Management
- Entitlement Management
- Compliance Management
- Identity federation Management

25. What is meant by Denial-of-Service(DoS) attack?

- Attacker will create many VMs on the host OS just to overload the host OS, which will not be able to accept anymore migrated VM's

26. What is data integrity and Availability?

- Integrity is the accuracy and consistency of data as well as the completeness and reliability of systems. Availability is the ability for users to access systems and information when needed, even under duress.

Part-B

1. **Explain in detail about Virtualization System's security attacks.**

Virtualization as a concept is experiencing more utility in recent times due to its ability to offer improved efficiency and scalability while reducing costs. The increased adoption of virtualization has also led to increased concerns about the security risks associated with virtualization.

Critical Virtualization Vulnerabilities or Security Attacks

1. **VM sprawl:** Is the unplanned proliferation of VMs. This unchecked proliferation can lead to VMs with sensitive information being compromised because they are not being actively managed and updated. Attackers can take advantage of poorly monitored resources.
2. **Hyperjacking:** Hyperjacking takes control of the hypervisor to gain access to the VMs and their data. It is typically launched against type 2 hypervisors that run over a host OS.
3. **VM escape:** A guest OS escapes from its VM encapsulation to interact directly with the hypervisor. This gives the attacker access to all VMs and, if guest privileges are high enough, the host machine as well.
4. **Denial of service:** These attacks exploit many hypervisor platforms and range from flooding a network with traffic to sophisticated leveraging of a host's own resources.
5. **Incorrect VM isolation:** To remain secure and correctly share resources, VMs must be isolated from each other. Poor control over VM deployments can lead to isolation breaches in which VMs communicate.
6. **Unsecured VM migration:** This occurs when a VM is migrated to a new host, and security policies and configuration are not updated to reflect the change. Potentially, the host and other guests could become more vulnerable.
7. **Host and guest vulnerabilities:** Host and guest interactions can magnify system vulnerabilities at several points.
8. **Access Controls**

An attacker gaining access to the virtual infrastructure, whether via physically accessing host servers or via a compromised user account on the management platform, can cause a lot of damage to the systems.

2. Explain in detail about Guest Hopping Attack of Virtualization System's security attacks or Virtual Machine Hyper Jumping Attack.
Write a note about Guest hopping and VM Migration attacks. Provide realtime case studies for the same. Nov 2023

Guest Hopping Attack

- A situation where the attacker penetrates two virtual machines is called a guest-hopping attack.
- In a guest-hopping attack, an attacker will try to identify two virtual machines likely to be hosted on the same physical hardware.
- Assuming the attacker is interested in data from virtual machine A, but is unable to directly penetrate virtual machine A, the attacker will try to penetrate virtual machine B, and then try to gain access to virtual machine A.

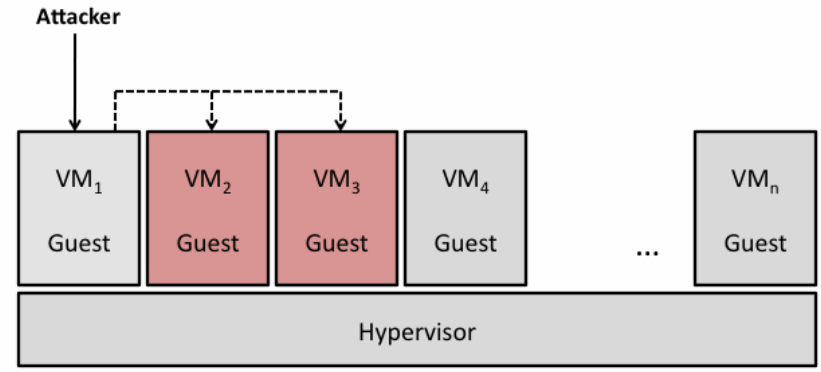


Figure 5.1 – Guest Hopping Attack

- In figure 5.1, the attacker can gain control of one of the guest virtual instance, and then make use of a famous exploit called as hypervisor escape.
- This vulnerability can let attacker traverse from one guest system to another, called as guest hopping, thus infecting the whole infrastructure.
- This attack is usually used to create zombie machines which are then used to plant a distributed denial of service attack.

Virtual Machine Hyper Jumping Attack

- Virtual machine hyper jumping exploits are designed to compromise a VM, which is then used to access or launch attacks against other VMs or hosts.
- This is usually done by targeting and accessing a less secure VM on a host, which is then used as the launch point for further attacks on the system.
- These attacks can occur due to:
  - Insecure operating systems like older versions of Windows, which do not have modern security features such as protection against poison cookies, memory address layout randomization and hardened stack
  - VM traffic to and from an external network utilizes the two-layer bridge, where all traffic passes through the same set of network interface cards (NICs). An attacker may overload the switch, and in order to preserve its performance, the switch pushes all data packets out on its ports. This action makes it a dumb hub, with no security usually offered by a switch.
- Virtual machine hyper jumping can be prevented using various methods, including:
  - Grouping and separating the uplinks to separate the Web-facing traffic from the database traffic and prevent the database server from directly accessing the internal network
  - Using private VLANs to hide the VMs from one another and only allow the guest machines to talk to the gateway
  - Using the latest and most secure operating systems with up-to-date security patches

3. Explain in detail about VM Migration Attacks in Virtualization Systems.

VM Migration

- VM migration, allows users to transfer operating system instances from one physical computer machine to multiple physical computer machines when the memory is full.

- Virtual machine downtime is the time at which users cannot use services of the VM; during VM downtime the VM services are not available to the users.
- The time taken by VM to migrate from host machine to destination machine is called VM migration time.

Types of VM Migration

Non-Live VM Migration/Cold Migration

- Non-live migration is a VM migration procedure in which the VM must be shut down in the host machine before transferring VM from the host computer to the destination computer.

Live VM Migration/Hot Migration

- Live VM migration is a VM migration procedure for transferring virtual machines (VMs) from one machine to another without causing any downtime. No need to shutdown host machine.

Security Issues in Live Migration – VM Migration Attack

- A live migration framework can be divided into three planes as shown in Figure 5.2.
  - **Control plane:** performs communication mechanisms during live Migration of virtual machines
  - **Data plane:** Data communication occurs through data plane
  - **Migration module:** The VMM migration, activation and monitoring is performed by the migration module.

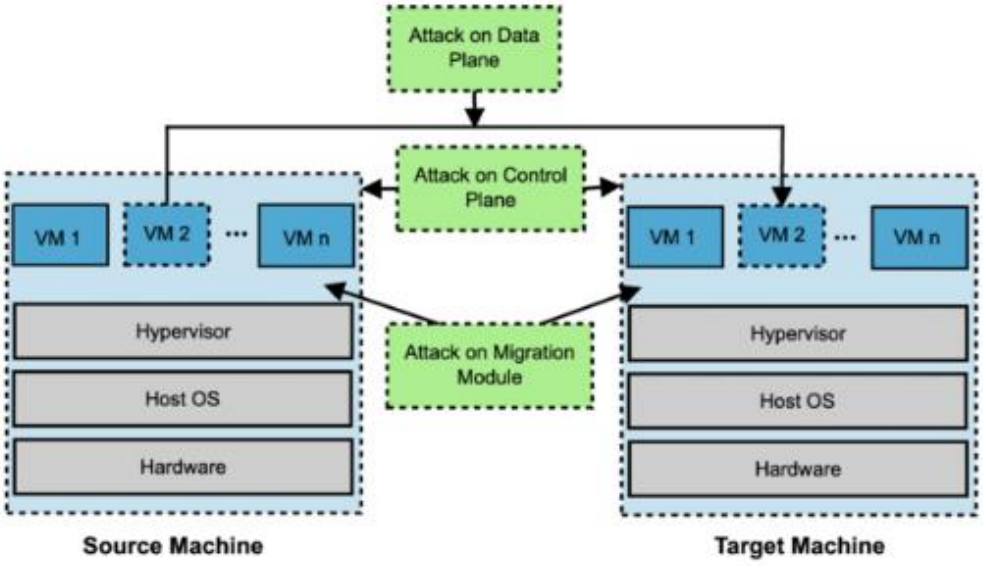


Figure 5.2 – Live Migration Framework

VM Migration Attack

- Migration attack is an attack on the network during VM migration from one place to another, since VM images are easily moved between physical machines through the network.
- For example, VMs from a canceled customer may be moved to a backup data center, and VMs that need maintenance may be moved to a testing data center for changes. Thus, when VMs are on the network between secured perimeters, attackers can exploit the network vulnerability to gain unauthorized access to VMs. Similarly, the attackers can plant malicious code in the VM images to plant attacks on data centers that VMs travel between.

Hyper Jacking Attack

- A virtual machine is a system that uses software to run programmes and deploy apps.
- Hyperjacking is an attack in which a hacker takes malicious control over the hypervisor that creates the virtual environment within a virtual machine (VM) host.
- The point of the attack is to target the operating system that is below that of the virtual machines so that the attacker's program can run and the applications on the VMs above it will be completely oblivious to its presence.
- **For a hyperjacking attack to succeed, an attacker would have to take control of the hypervisor by the following methods:**
    - Injecting a rogue hypervisor beneath the original hypervisor
    - Directly obtaining control of the original hypervisor
    - Running a rogue hypervisor on top of an existing hypervisor
- **Three ways in which hyper jacking can be a potential threat to virtual machines:**

- Hyperjacking can slow down VMs by allowing attackers to abuse the virtual machine's resources, such as CPU, memory, and disc space.
- Hyperjacking can be used to launch DoS attacks on virtual machines, preventing legitimate users from accessing them.
- Hyperjacking can allow attackers to escalate their privileges within the virtual machine.

- **Ways in which hyper-jacking happens are:**
  - Wired Hyperjacking: This type of hyperjacking involves physical access to the hypervisor. The attacker gains access to the hypervisor through the physical network interface
  - Remote Hyperjacking: This type of hyperjacking involves exploiting vulnerabilities in the hypervisor software to gain control of the system remotely.
  - Hypervisor-level Rootkits: This type of hyperjacking involves using hypervisor-level rootkits to gain control of the hypervisor layer. The attacker can then use the rootkit to hide their activities from the virtual machines running on the system.
- **Methods to prevent hyper jacking**
  - Keep hypervisors and virtual machines up to date.
  - Use network segmentation techniques.
    - Dividing a network into smaller segments can limit the attacker's access to the hypervisor. It becomes easier to detect and respond to suspicious activity.
  - Implement access controls.
    - Implement strong password policies, two-factor authentication, and least privilege principles to reduce the risk of hyper-jacking attacks.

- Implement security policies.
    - Organizations must have clear security policies that cover the use of virtualization technologies and the protection of hypervisors and virtual machines.
- Intrusion detection and prevention systems.
    - IDPSs monitor network traffic and alert system administrators when suspicious activity is detected. They can also block traffic from suspicious sources and prevent attacks from succeeding.
- Use wired hyper-jacking prevention mechanisms.
    - Use network interface card (NIC) filtering, switch port security, and cable locks to prevent wired hyper jacking.
- Conduct regular security audits.
    - Regular security audits to identify potential vulnerabilities in the virtual infrastructure.
- Practice virtual machine encryption.
    - Encrypting virtual discs can prevent unauthorised access to the virtual machine and its data.

4. Explain in detail about Data Security and Storage in Cloud computing systems.

Data Security

- Data security is the practice of protecting digital information from unauthorized access, corruption or theft throughout its entire lifecycle.

Several aspects of data security, including:

- Data-in-transit - to ensure that a protocol provides confidentiality as well as integrity particularly if the protocol is used for transferring data across the Internet.

- Data-at-rest - refers to computer data in digital form, such as cloud storage, file hosting services, databases, or data warehouses.

  To prevent access to, modification, or theft of data at rest, organizations typically use security controls such as password protection, data encryption, physical controls, and monitoring.

- Processing of data, including multitenancy - For any application to process data, that data must be unencrypted., IBM had developed a fully homomorphic encryption scheme which allows data to be processed without being decrypted.

- Data lineage - It might be required (for audit or compliance purposes) to know exactly where and when the data was specifically located within the cloud. Following the path of data (mapping application data flows or data path visualization) is known as data lineage, and it is important for an auditor's assurance (internal, external, and regulatory).

- Data provenance - Provenance means that the data is computationally accurate; that is, the data was accurately calculated.

- Data remanence - Data remanence is the residual representation of data that has been nominally erased or removed. This residue may be due to data being left intact by a nominal delete operation, or through physical properties of the storage medium.

Data security threats in cloud computing

- Unsecure application programming interfaces (APIs)
- Account hijacking or takeover
- Insider threats

Benefits of Cloud Data Security

- Mitigate Data Breach Risk

- Protect Brand Reputation
- Enhance Customer Trust
- Avoid Fines and Fees

Best Practices for Implementing Cloud Data Security

- Identify Sensitive Data
- Classify Data Using Context
- Limit Access to Resources
  - Role-based access controls (RBAC)
  - Attribute-based access controls (ABAC)
- Encrypt Data-in-Transit and Data-at-Rest
- Implement Data Loss Prevention (DLP)
- Harden Data Posture
- Continuously Monitor Real-Time Data Risk
- Create a Single Source for Continuous Monitoring, Remediation, and Documentation

Storage

- The three information security concerns are associated with the data stored in the cloud: confidentiality, integrity, and availability.

1. Confidentiality
2. Protecting the data from unauthorized access and disclosure.
3. Access control consists of both authentication and authorization.
4. Authentication is the process of verifying the identity of a user or system.
5. Authorization is the process of determining the privileges the user or system is entitled to once the identity is established.

6. Protection of data stored in the cloud involves the use of symmetric encryption as in figure 5.3.
7. Symmetric encryption involves the use of a single secret key for both the encryption and decryption of data.

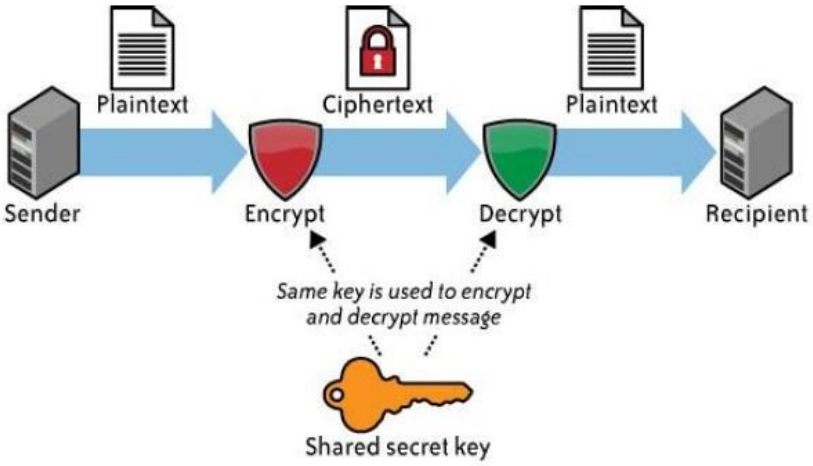


Figure 5.3 - Symmetric Encryption

8. Only symmetric encryption has the speed and computational efficiency to handle encryption of large volumes of data.
9. With symmetric encryption, the longer the key length (i.e., the greater number of bits in the key), the stronger encryption.

Integrity

10. Safeguard the data from unauthorized modification so it can be trusted It requires the use of message authentication codes (MACs).

Availability

11. Assuming that a customer's data has maintained its confidentiality and integrity, must also be concerned about the availability of the data.
12. Ensuring the data is fully available and accessible when it's needed
13. All three of these considerations (confidentiality, integrity, and availability) should be encapsulated in a CSP's service-level agreement (SLA) to its customers.

5. Discuss in detail about Identity and Access Management (IAM) and its Challenges.

Identity and Access Management (IAM) Definition

14. The Identity and access management is the security framework composed of policy and governance components used for creation, maintenance and termination of digital identities with controlled access of shared resources.

15. It focuses on two parts Identity management and access Management.

16. Identity and Access management (IAM) aid in Authentication, Authorization, and Auditing (AAA) of users accessing cloud services.

Authentication

17. Authentication is the process of verifying the identity of a user or system.

Authorization

- Authorization is the process of determining the privileges the user or system is entitled to once the identity is established.

Auditing

- Auditing entails the process of review and examination of authentication, authorization records, and activities to determine the adequacy of IAM system controls, to verify compliance with established security policies and procedures to detect breaches in security services and to recommend any changes that are indicated for countermeasures.
- Organizations invest in IAM practices to improve operational efficiency and to comply with regulatory, privacy, and data protection requirements:

Improve operational efficiency

- Properly architected IAM technology and processes can improve efficiency by automating user on-boarding and other repetitive tasks

Regulatory compliance management

- To protect systems, applications, and information from internal and external threats and to comply with various regulatory, privacy, and data protection requirements.
- IAM processes and practices can help organizations to meet objectives in the area of access control and operational security

In addition to improving operational efficiencies and effective compliance management, IAM can enable new IT delivery and deployment models

IAM Challenges

1. Identity Provisioning / De-provisioning

   Providing a secure and timely management of on-boarding (provisioning) and off-boarding (de-provisioning) of users in the cloud.

2. Maintaining a single ID across multiple platforms and organizations. Keeping Application Integrations Updated

   By enabling a single sign on facility, the organization can extend IAM processes and practices to the cloud and implement a standardized federation model to support single sign-on to cloud services.

3. Compliance Visibility: Who has access to what

   There should be a central visibility and control across all your systems for auditing purposes.

4. Security when using 3rd party or vendor network

A lot of services and applications used in the cloud are from 3rd party or vendor networks but can't guarantee that their security is adequate..

5. Lack of centralized view

   As companies switch from storing their data on-site to storing it in the cloud, centralized on-site data has become decentralized.

6. Keeping Application Integrations Up to Date
7. Different Administration Models for Different Applications
8. Excessive permissions granted to users with no business need
9. Misconfigurations of cloud environments and customer security Policies
10. Public exposure of assets without proper (or any) security controls
11. Malicious access by unauthorized 3rd-parties to the cloud environment

6. Diagrammatically discuss about Identity and Access Management (IAM) Architecture and its processes in detail.

What is Identity and Access Management? Describe its Architecture. Depict the procedure to carry out IAM in AWS cloud platform.Nov 2023

IAM Processes:

The processes supported by IAM are given as follows.

- User management - It provides processes for managing the identity of different entities.
- Authentication management - It provides activities for management of the process for determining that an entity is who or what it claims to be.
- Access management - It provides policies for access control in response to request for resource by entity.

- Data management - It provides activities for propagation of data for authorization to resources using automated processes.
- Authorization management - It provides activities for determining the rights associated with entities and decide what resources an entity is permitted to access in accordance with the organization's policies.
- Monitoring and auditing - Based on the defined policies, it provides monitoring, auditing, and reporting compliance by users regarding access to resources.

Lifecycle of IAM:

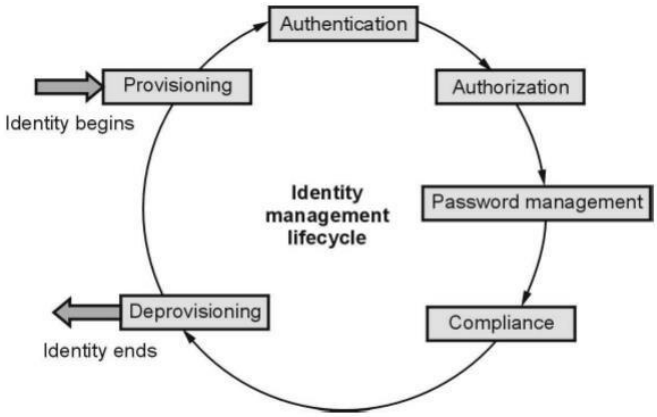


Figure 5.4 - Lifecycle of IAM

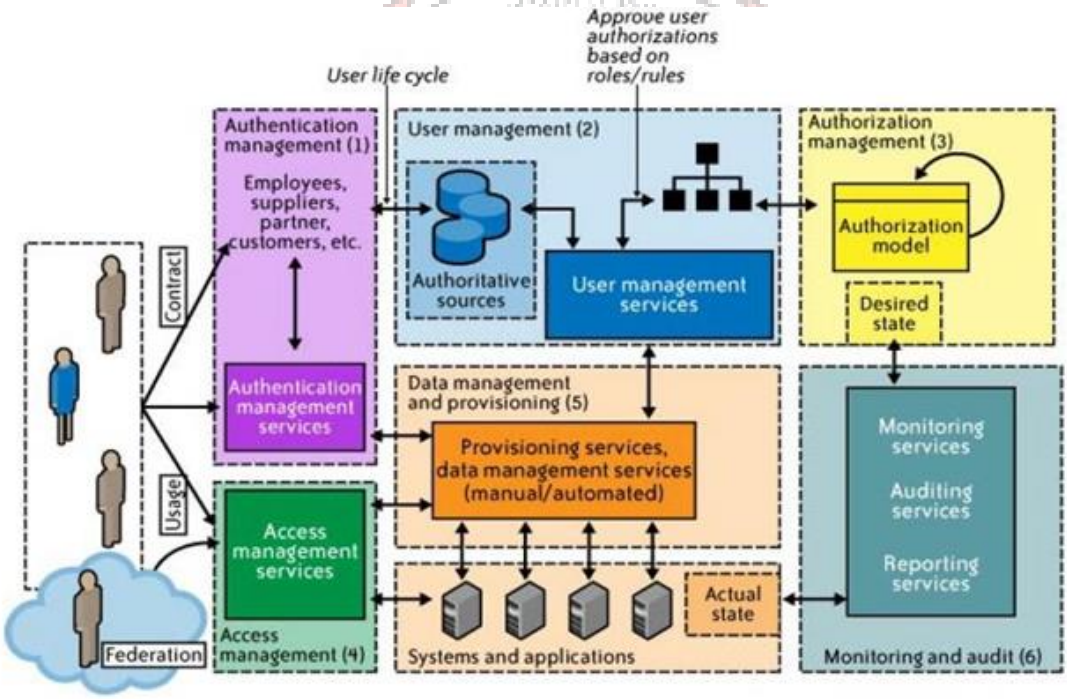


Figure 5.5 - Enterprise IAM functional architecture

The activities supported by IAM (in Figure 5.5) are given as follows.

a) Provisioning - The provisioning has essential processes that provide users with necessary access to data and resources. It supports management of all user account operations like add, modify, suspend, and delete users with password management..

b) Credential and attribute management - The Credential and attribute management prevents identity impersonation and inappropriate account use. The Credential and attribute management processes include provisioning of static or dynamic attributes that comply with a password standard, encryption management of credentials and handling access policies for user attributes.

c) Compliance management - The Compliance management is the process used for monitoring the access rights and privileges and tracked to ensure the security of an enterprise's resources. It includes practices like access monitoring, periodic auditing, and reporting.

d) Identity federation management - Identity federation management is the process of managing the trust relationships beyond the network boundaries where organizations come together to exchange the information about their users and entities.

e) Entitlement management - In IAM, entitlements are authorization policies. The Entitlement management provides processes for provisioning and deprovisioning of privileges needed for the users to access the resources including systems, applications, and databases.

Reg. No. : 4 2 1 6 2 1 2 0 5 0 1 8

Question Paper Code : 20398

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2023.

Fifth Semester

Computer Science and Design

CCS 335 – CLOUD COMPUTING

(Common to : Computer Science and Engineering/Computer Science and Engineering (Artificial Intelligence and Machine Learning)/Computer Science and Engineering (Cyber Security)/Computer and Communication Engineering/Artificial Intelligence and Data Science/Computer Science and Business Systems and Information Technology)

(Regulations 2021)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Tabulate the Design challenges of cloud computing.
2. Distinguish between Public and Private Clouds.
3. What is the hypervisor?
4. Write the role of CPU Virtualization.
5. What are the benefits of Network Virtualization?
6. What is a Docker in Cloud Computing?
7. Summarize the service offerings by AWS.
8. Depict the benefits of OpenStack Compute.
9. What is hyperjacking?
10. List out the IAM challenges.

PART B — (5 × 13 = 65 marks)

11. (a) What are Peer-to-Peer Network Families? Describe the NIST cloud computing reference architecture with its components.

Or

(b) Discuss the various cloud service, deployment models with neat sketch.

12. (a) What are the different implementation levels of Virtualization? Explain.

Or

(b) Compare the terms Full Virtualization and Para Virtualization and depict the process of virtualizing CPU, memory, I/O devices.

13. (a) Compare and contrast the Physical Clusters and the Virtual Clusters and depict how resource management could be carried out in virtual machines.

Or

(b) What are the different components of a Docker? Explain its need and use.

14. (a) What is Google App Engine? Describe the major building blocks and functional modules of the Google Cloud Platform with a diagram.

Or

(b) What is Eucalyptus? Discuss its architecture by mentioning their role in filtering incoming traffic.

15. (a) Write a note about Guest hopping and VM Migration attacks. Provide realtime case studies for the same.

Or

(b) What is Identity and Access Management? Describe its architecture. Depict the procedure to carry out IAM in AWS cloud platform.

PART C — (1 × 15 = 15 marks)

16. (a) Depict the taxonomy of virtual machines and narrate steps for launching a virtual server in AWS cloud platform.

Or

(b) Depict the system requirements, software configuration, memory requirements for creating an e-commerce website in cloud platform with S3 storage and hosting the same in a laptop.

———————