**CCS335-CLOUD COMPUTING**

**PART B**

**UNIT I – CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE**

**Q1. Explain the NIST Cloud Computing Reference Architecture with its components.**

**Answer:** The NIST Cloud Computing Reference Architecture defines standard roles and interactions in a cloud ecosystem:

**Actors:**

* **Cloud Consumer**: Uses cloud services.
* **Cloud Provider**: Offers cloud services.
* **Cloud Broker**: Intermediary between provider and consumer.
* **Cloud Auditor**: Performs audits on security, privacy.
* **Cloud Carrier**: Connects cloud provider and consumer.

**Service Models:**

* **IaaS**: Infrastructure as a Service – e.g., AWS EC2.
* **PaaS**: Platform as a Service – e.g., Google App Engine.
* **SaaS**: Software as a Service – e.g., Gmail, Office 365.

**Deployment Models:**

* Private, Public, Hybrid, Community.

The NIST model promotes a standardized understanding of cloud systems.

**Q2. Discuss different types of cloud deployment models with examples.**

**Answer:** Cloud deployment models define how cloud services are made available:

1. **Private Cloud**:
   * Used by one organization.
   * Secure, customizable.
   * Example: Banking applications.
2. **Public Cloud**:
   * Services over the internet.
   * Multi-tenant and cost-effective.
   * Example: AWS, Azure.
3. **Hybrid Cloud**:
   * Combines private and public.
   * Flexible, scalable.
   * Example: Data stored privately but processed publicly.
4. **Community Cloud**:
   * Shared by several organizations with common goals.
   * Example: Government departments.

Each model supports different business needs and data sensitivities.

**Q3. Describe different cloud service models: IaaS, PaaS, and SaaS.**

**Answer:** **1. IaaS (Infrastructure as a Service)**:

* Provides virtual machines, storage, networks.
* User manages OS and apps.
* Example: Amazon EC2, Google Compute Engine.

**2. PaaS (Platform as a Service)**:

* Provides platforms for app development.
* Abstracts hardware and OS.
* Example: Google App Engine, Heroku.

**3. SaaS (Software as a Service)**:

* Provides software applications over internet.
* User only uses the app; everything else is managed.
* Example: Gmail, Dropbox.

These models provide scalability, flexibility, and reduce IT management effort.

**Q4. What are the architectural design considerations for compute and storage clouds?**

**Answer:**

**Compute Cloud Architecture:**

* **Virtualization Layer**: Provides VM abstraction.
* **Resource Scheduler**: Allocates CPU, memory.
* **Load Balancer**: Manages traffic and performance.
* **Elasticity Controller**: Scales up/down resources.

**Storage Cloud Architecture:**

* **Block Storage**: For VM boot volumes.
* **Object Storage**: For unstructured data (e.g., S3).
* **File Storage**: Shared file systems (e.g., NFS).

**Design Considerations:**

* Scalability
* Fault tolerance
* Redundancy
* Data consistency
* Cost-efficiency

Both compute and storage clouds must be designed for performance, availability, and reliability.

**Q5. Explain the key design challenges in cloud infrastructure.**

**Answer:**

**1. Security and Privacy**:

* Protecting data in shared environments.
* Secure access control and encryption.

**2. Scalability**:

* Auto-scaling resources as demand grows.

**3. Availability and Reliability**:

* Redundancy and failover mechanisms.

**4. Interoperability**:

* Supporting different platforms and APIs.

**5. Resource Management**:

* Efficiently managing compute, storage, and network.

**6. Cost Management**:

* Balancing performance with budget.

Designing cloud infrastructure requires balancing flexibility, performance, and security across diverse users and applications.

**UNIT II – VIRTUALIZATION BASICS**

**Q1. Explain the taxonomy and structure of virtual machines.**

**Answer:** **Taxonomy of VMs:**

1. **System VMs** – Provide complete system platform (e.g., VMware).
2. **Process VMs** – Run a single process or application (e.g., Java Virtual Machine).

**Structure of VMs:**

* **Hardware** → **Host OS** → **Hypervisor** → **Guest OS** → **Applications**

Hypervisors can be:

* **Type 1 (Bare-metal)** – Runs directly on hardware (e.g., Xen, VMware ESXi).
* **Type 2 (Hosted)** – Runs on host OS (e.g., VirtualBox).

This structure isolates virtual environments, enabling multiple OSes on a single physical machine.

**Q2. Describe different levels of virtualization implementation.**

**Answer:**

1. **Instruction Set Architecture (ISA) Level**:
   * Emulates hardware architecture.
   * Example: Bochs.
2. **Hardware Level**:
   * Uses hypervisors to share hardware.
   * Example: VMware ESXi.
3. **Operating System Level**:
   * Uses container-based virtualization.
   * Example: Docker, LXC.
4. **Library Level**:
   * Modifies app libraries to interact with virtual layer.
5. **Application Level**:
   * Virtualizes at app level (e.g., Java Virtual Machine).

Each level has different performance trade-offs and application scenarios.

**Q3. What is a hypervisor? Explain its types and functions.**

**Answer:** A **hypervisor** is software that creates and manages virtual machines by abstracting hardware.

**Types of Hypervisors:**

1. **Type 1 (Bare-metal)**:
   * Installed directly on hardware.
   * High performance.
   * Examples: Hyper-V, Xen.
2. **Type 2 (Hosted)**:
   * Runs on an OS.
   * Easy to use but slower.
   * Examples: VirtualBox, VMware Workstation.

**Functions:**

* Manage CPU, memory, and I/O.
* Isolation and scheduling of VMs.
* Live migration and resource allocation.

Hypervisors enable multi-tenancy and scalability in cloud systems.

**Q4. Compare full virtualization and para-virtualization.**

**Answer:**

| **Feature** | **Full Virtualization** | **Para Virtualization** |
| --- | --- | --- |
| Guest OS | Unmodified | Modified |
| Performance | Slower due to binary translation | Faster due to direct calls |
| Hardware Support | Required | Not strictly required |
| Hypervisor Role | Emulates entire hardware | Provides hypercalls to OS |
| Example | VMware, VirtualBox | Xen with modified Linux |

Para-virtualization offers higher efficiency at the cost of OS modification.

**Q5. Explain virtualization of CPU, memory, and I/O devices.**

**Answer:**

**CPU Virtualization**:

* Time-sharing of CPU among VMs.
* Hardware features (Intel VT-x) assist virtualization.

**Memory Virtualization**:

* Virtual Address → Guest OS Physical → Host Physical.
* Techniques: Shadow page tables, nested paging.

**I/O Virtualization**:

* Emulates I/O devices like network cards and disks.
* Uses device drivers or direct I/O mapping.

These enable efficient and isolated resource sharing in virtualized environments.

**UNIT III – VIRTUALIZATION INFRASTRUCTURE AND DOCKER**

**Q1. Explain desktop, network, and storage virtualization.**

**Answer:**

**Desktop Virtualization**:

* Centralized desktops accessed remotely.
* Example: VDI (Virtual Desktop Infrastructure).

**Network Virtualization**:

* Combines hardware and software networks.
* Enables multiple virtual networks.
* Example: VLAN, SDN.

**Storage Virtualization**:

* Pools physical storage into a virtual storage device.
* Example: SAN, NAS.

These forms increase resource utilization, manageability, and flexibility.

**Q2. Describe system-level and application-level virtualization.**

**Answer:**

**System-level Virtualization**:

* Virtualizes entire OS.
* Example: VirtualBox, VMware.

**Application-level Virtualization**:

* Virtualizes individual applications.
* App runs in isolated environment.
* Example: Java VM, Docker.

System-level provides full OS emulation, while application-level focuses on isolated apps.

**Q3. Explain virtual clusters and resource management in cloud computing.**

**Answer:**

**Virtual Clusters**:

* Group of VMs working as a unit.
* Used for high availability, load balancing.

**Resource Management**:

* Scheduling of CPU, memory, storage, and bandwidth.
* Techniques:
  + Resource pooling
  + Load balancing
  + VM migration

Efficient resource management ensures QoS and cost-effectiveness.

**Q4. Compare containers and virtual machines.**

**Answer:**

| **Feature** | **Containers** | **Virtual Machines** |
| --- | --- | --- |
| OS Sharing | Shared host OS | Separate OS |
| Size | Lightweight | Heavy |
| Boot Time | Seconds | Minutes |
| Isolation | Less than VMs | Full isolation |
| Performance | Near-native | Lower due to hypervisor |

Containers are ideal for microservices and DevOps, while VMs are suited for full OS use cases.

**Q5. Explain Docker architecture and its components.**

**Answer:**

**Docker Architecture**:

* **Docker Engine**: Core service.
* **Docker Daemon**: Background process that manages containers.
* **Docker CLI**: Command-line tool to interact with daemon.
* **Docker Images**: Read-only templates for containers.
* **Docker Containers**: Running instances of images.
* **Docker Hub**: Repository for sharing images.

Docker enables consistent, scalable application deployment across environments.

**UNIT IV – CLOUD DEPLOYMENT ENVIRONMENT**

**Q1. Explain the architecture and services of Google App Engine.**

**Answer:**

**Google App Engine (GAE)**:

* Platform as a Service (PaaS).
* Supports languages: Python, Java, Go.

**Architecture**:

* App uploaded via SDK.
* Deployed to managed runtime.
* Auto-scaling and load balancing built-in.

**Services**:

* Datastore, Task Queues, Memcache, APIs.
* Integrated with Google Cloud Storage and Pub/Sub.

GAE abstracts infrastructure, allowing developers to focus on coding.

**Q2. Describe the key features and services of Amazon AWS.**

**Answer:**

**AWS Core Services**:

* **EC2**: Elastic Compute Cloud (VMs).
* **S3**: Scalable object storage.
* **RDS**: Relational Database Service.
* **Lambda**: Serverless function execution.

**Features**:

* Pay-as-you-go model.
* Auto-scaling and load balancing.
* High availability across global regions.

AWS supports all cloud models (IaaS, PaaS, SaaS) and offers strong IAM features.

**Q3. Explain Microsoft Azure architecture and deployment models.**

**Answer:**

**Azure Architecture**:

* Global data centers across regions.
* Services via Azure Resource Manager (ARM).
* Virtual Network for connectivity.

**Deployment Models**:

* **IaaS**: VMs, storage, networking.
* **PaaS**: App services, databases.
* **SaaS**: Office 365, Dynamics.

Azure supports hybrid cloud and DevOps integrations.

**Q4. What is OpenStack? Explain its architecture.**

**Answer:**

**OpenStack**:

* Open-source cloud computing platform.
* Provides IaaS.

**Components**:

* **Nova**: Compute management.
* **Swift**: Object storage.
* **Cinder**: Block storage.
* **Neutron**: Networking.
* **Keystone**: Identity management.
* **Horizon**: Web UI.

Highly modular, scalable, and supports private and public cloud environments.

**Q5. Write short notes on Eucalyptus cloud platform.**

**Answer:**

**Eucalyptus**:

* Stands for Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems.
* Open-source IaaS cloud platform.

**Components**:

* **Cloud Controller**: Manages high-level scheduling.
* **Cluster Controller**: Manages VM clusters.
* **Node Controller**: Executes VMs.
* **Walrus**: Object storage (like Amazon S3).
* **Storage Controller**: Block storage management.

Compatible with AWS APIs; used for building private clouds.

**UNIT V – CLOUD SECURITY**

**Q1. Explain virtualization system-specific attacks in cloud computing.**

**Answer:**

**1. Guest Hopping**:

* Attacker gains access to another VM on same host.

**2. VM Migration Attack**:

* Attacks during live migration (data sniffing, hijacking).

**3. Hyperjacking**:

* Malware installs a rogue hypervisor.

**Prevention**:

* Strong hypervisor security.
* Encrypted VM migrations.
* Regular patching.

These attacks target the virtual environment, requiring specialized defenses.

**Q2. Describe identity and access management (IAM) architecture.**

**Answer:**

**Components**:

* **User Management**: Create/delete users.
* **Authentication**: Verifies identity (passwords, MFA).
* **Authorization**: Grants access based on roles/policies.
* **Audit Logs**: Track user actions.

**IAM Flow**: User → Authentication → Policy Check → Access Grant → Logging

Ensures secure and authorized access to cloud resources.

**Q3. What are the major challenges in implementing IAM in the cloud?**

**Answer:**

1. **Identity Federation** – Multiple identity sources.
2. **Access Control Complexity** – Dynamic cloud roles.
3. **Compliance Requirements** – Vary by industry and country.
4. **Scalability** – Managing millions of users.
5. **Insider Threats** – Risk from within the organization.

Robust IAM frameworks are required for effective access governance.

**Q4. Explain data security and storage challenges in the cloud.**

**Answer:**

**Security Concerns**:

* **Data Breach**
* **Data Loss**
* **Unauthorized Access**

**Storage Challenges**:

* **Multi-Tenancy**: Isolation between tenants.
* **Data Location**: Legal issues with data stored in other countries.
* **Redundancy & Backup**: Ensuring availability.

**Solutions**:

* Encryption
* Tokenization
* Access controls

**Q5. What is the role of encryption and access control in cloud security?**

**Answer:**

**Encryption**:

* Protects data in transit and at rest.
* Algorithms: AES, RSA.

**Access Control**:

* Prevents unauthorized access.
* Models: RBAC, ABAC, MAC.

Together, they secure sensitive data from internal and external threats.