**CCS335-CLOUD COMPUTING**

**PART C**

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

**Q1. Explain the NIST Cloud Computing Reference Architecture in detail.**

**Answer:** The **NIST Cloud Computing Reference Architecture** provides a conceptual framework based on five key actors:

**1. Cloud Consumer**:

* Uses services from the cloud provider (IaaS, PaaS, SaaS).

**2. Cloud Provider**:

* Delivers cloud services.
* Manages infrastructure, platforms, and applications.

**3. Cloud Auditor**:

* Evaluates cloud services for compliance and security.

**4. Cloud Broker**:

* Manages service usage, performance, and delivery.
* Acts as intermediary between provider and consumer.

**5. Cloud Carrier**:

* Connects provider and consumer via network.

**Three Service Models**:

* **IaaS**: Infrastructure (e.g., VMs, storage).
* **PaaS**: Platform (e.g., application deployment).
* **SaaS**: Software access (e.g., Google Docs).

**Four Deployment Models**:

* **Private Cloud**
* **Public Cloud**
* **Hybrid Cloud**
* **Community Cloud**

This architecture enhances understanding and implementation of secure cloud systems.

**Q2. Compare and explain cloud service models: IaaS, PaaS, and SaaS.**

**Answer:**

| **Feature** | **IaaS** | **PaaS** | **SaaS** |
| --- | --- | --- | --- |
| Service | Infrastructure | Platform | Software |
| Access | VMs, storage, networks | App hosting environments | Web-based apps |
| Control | High (OS, runtime) | Medium (only apps) | Low (only config) |
| Example | AWS EC2, Google Compute | Google App Engine, Heroku | Gmail, Salesforce |

* **IaaS**: Offers virtualized hardware.
* **PaaS**: Simplifies app development/deployment.
* **SaaS**: Delivers ready-to-use software via internet.

Each model suits different levels of developer and user control.

**Q3. Discuss architectural design challenges of compute and storage clouds.**

**Answer:**

**Compute Cloud Challenges**:

* **Scalability**: Elastic demand requires dynamic resource allocation.
* **Fault Tolerance**: Must handle failures gracefully.
* **Load Balancing**: Distribute traffic evenly.
* **VM Management**: Efficient migration and isolation.

**Storage Cloud Challenges**:

* **Data Consistency**: ACID compliance in distributed systems.
* **Latency**: High-performance access over networks.
* **Redundancy**: Backup and replication mechanisms.
* **Security**: Data encryption, access control.

Designing robust, scalable, and secure compute and storage layers is critical for cloud efficiency.

**UNIT II – VIRTUALIZATION BASICS**

**Q1. Describe the taxonomy and implementation levels of virtualization.**

**Answer:**

**Taxonomy**:

* **System Virtualization**: Whole OS virtualized (VMware).
* **Process Virtualization**: Single process virtualized (JVM).

**Types**:

* **Full Virtualization**: Guest OS runs unmodified.
* **Para-Virtualization**: Guest OS modified to cooperate with hypervisor.
* **Hardware Virtualization**: Uses CPU instructions (Intel VT).

**Implementation Levels**:

1. **Instruction Set Architecture Level**
2. **Hardware Level**
3. **OS Level**
4. **Library Level**
5. **Application Level**

Each level has pros and cons for performance, complexity, and portability.

**Q2. Explain the structure and working of a hypervisor in detail.**

**Answer:**

**Hypervisor**:

* Software that creates and manages virtual machines by abstracting hardware.

**Types**:

* **Type 1**: Bare-metal hypervisor (runs directly on hardware).
* **Type 2**: Runs on host OS (e.g., VirtualBox).

**Functions**:

* **Resource Allocation**
* **Isolation and Security**
* **Live Migration**
* **Snapshot Management**

**Working**:

1. Hosts multiple VMs.
2. Schedules VM access to hardware.
3. Manages memory, CPU, I/O for each VM.

Hypervisors form the core of cloud and virtualization technologies.

**Q3. Explain CPU, memory, and I/O virtualization mechanisms.**

**Answer:**

**CPU Virtualization**:

* Time-sharing.
* Hypervisor intercepts privileged instructions.
* Hardware assists like Intel VT-x improve performance.

**Memory Virtualization**:

* Virtual Address → Guest Physical → Host Physical.
* Techniques:
  + Shadow Page Tables
  + Nested Paging

**I/O Virtualization**:

* Emulates I/O devices for VMs.
* Uses direct device mapping or device drivers.

All three are essential to abstract physical resources for VM use.

**UNIT III – VIRTUALIZATION INFRASTRUCTURE AND DOCKER**

**Q1. Compare containers and virtual machines. Explain Docker architecture.**

**Answer:**

**VMs**:

* Heavyweight.
* Each has full OS.
* Slower boot time.

**Containers**:

* Lightweight.
* Share host OS.
* Fast and efficient.

**Docker Architecture**:

* **Docker Engine**: Core component.
* **Docker Daemon**: Manages containers.
* **Docker CLI**: Command interface.
* **Docker Images**: Templates for containers.
* **Docker Hub**: Cloud-based registry.

Docker improves application deployment, isolation, and scalability.

**Q2. Explain virtualization infrastructure: desktop, network, and storage virtualization.**

**Answer:**

**Desktop Virtualization**:

* Centralized desktops accessible via thin clients.

**Network Virtualization**:

* VLANs, software-defined networks (SDN).
* Abstracts physical network into logical units.

**Storage Virtualization**:

* Abstracts storage resources.
* Pools data from multiple devices.
* Managed via SAN/NAS.

These enable flexible, cost-effective IT environments.

**Q3. Describe containers vs. VMs, Docker components, and image lifecycle.**

**Answer:**

**Containers vs. VMs**:

* Containers are lightweight, share OS.
* VMs are heavy, require full OS.

**Docker Components**:

* **Docker Engine**
* **Images**
* **Containers**
* **Registries (Docker Hub)**

**Image Lifecycle**:

1. Create Dockerfile
2. Build image
3. Push to registry
4. Pull on host
5. Run container

Docker simplifies deployment with reproducible environments.

**UNIT IV – CLOUD DEPLOYMENT ENVIRONMENT**

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

**Answer:**

**Architecture**:

* Supports languages like Python, Java, Go.
* Managed runtime environment.
* Integrated data store, queue, and cache.

**Key Services**:

* **App Hosting**
* **Datastore (NoSQL)**
* **Task Queues**
* **Memcache**

**Features**:

* Auto-scaling
* Version control
* Google Cloud integration

GAE is a PaaS offering allowing developers to focus only on code.

**Q2. Compare Amazon AWS, Google Cloud, and Microsoft Azure.**

**Answer:**

| **Feature** | **AWS** | **Google Cloud** | **Microsoft Azure** |
| --- | --- | --- | --- |
| Launch Year | 2006 | 2011 | 2010 |
| Compute | EC2 | GCE | Azure VMs |
| Storage | S3 | Cloud Storage | Blob Storage |
| Database | RDS | Cloud SQL | SQL Database |
| AI/ML | SageMaker | AI Platform | Azure ML |

All offer IaaS, PaaS, SaaS, and DevOps tools.

**Q3. Explain OpenStack architecture and its components.**

**Answer:**

**OpenStack**:

* Open-source IaaS platform.

**Core Components**:

* **Nova**: Compute
* **Swift**: Object Storage
* **Cinder**: Block Storage
* **Neutron**: Networking
* **Keystone**: Authentication
* **Horizon**: Dashboard
* **Glance**: Image Service
* **Heat**: Orchestration

Highly modular and suitable for private/public clouds.

**UNIT V – CLOUD SECURITY**

**Q1. Discuss IAM in cloud computing. Explain architecture and challenges.**

**Answer:**

**IAM Architecture**:

* Identity provisioning
* Authentication (MFA, tokens)
* Authorization (RBAC, ABAC)
* Audit and compliance logs

**Challenges**:

* Federation across cloud services
* Role sprawl and access complexity
* Compliance (GDPR, HIPAA)
* Scalability with user growth

IAM is crucial for controlling access and maintaining data security.

**Q2. Explain data security and storage issues in cloud computing.**

**Answer:**

**Data Security Concerns**:

* Data breach
* Unauthorized access
* Insider threats

**Storage Issues**:

* Data location
* Data loss
* Redundancy

**Solutions**:

* Encryption (AES, RSA)
* Tokenization
* Access controls
* Backup and replication

Security policies must address confidentiality, integrity, and availability.

**Q3. Describe virtualization-specific attacks and their mitigation strategies.**

**Answer:**

**Types**:

* **Guest Hopping**: Attack from one VM to another.
* **Hyperjacking**: Install rogue hypervisor.
* **VM Escape**: Escape VM boundary to host.
* **VM Migration Attack**: Intercept during live migration.

**Mitigation**:

* Patch hypervisors regularly
* Use encryption for VM transfers
* Strong VM isolation
* Monitor system logs

Virtualization adds attack surfaces but can be secured with layered defense.