CLOUD COMPUTING

**1) BROADBAND NETWORKS AND INTERNET ARCHITECTURE**

**3) VIRTUALIZATION**

Server, Storage, Network, Power

host OS -> hypervisor -> guest OS

Creation of Virtual Servers

1) Allocation of physical IT resources -> OS installation using virtualization

2) Virtual servers -> guest OS

Deployment of virtual servers

1) Virtualization software runs on physical server (host)

2) System services related to virtual management -> not normally found on standard OS -> VMM or hypervisor

A. Hardware Independence

host (hardware) -> virtual servers are easily portable -> easier to duplicate than duplicating physical hardware

B. Server Consolidation

same virtualization host can create multiple virtual servers

C. Resource Replication

copy, move, paste and replicate, migrate and backup the virtual server

D. OS-Based Virtualization

hardware-based IT-resources -> virtual IT-resources

system-based virtualization -> overhead needed to run V-software and host-OS: CPU, memory, calls (layered) -> decreased performance, licence

Eg: backup and recovery, integration and directory services, security management

E. Hardware-Based Virtualization

virtualization software installed on an OS, directly on physical host hardware

hypervisor -> simple user interface

issue: hardware compatibility

F. Virtualization Management

VIM -> Virtualization Infrastructure Management -> collectively manage virtual IT resources

G. Technical and Business Considerations

a. Performance Overhead

b. Special Hardware Compatibility

c. Portability -> portability gaps

**4) WEB TECHNOLOGY**

Basic technology: URL (uniform resource locator), HTTP (hyper text transfer protocol), HTML (hyper text markup language), XML (extensive markup language)

WWW – world wide web -> system of interlinked IT resources

Web components: web browser client, web server

URL: protocol (http) + subdomain (www) + domain-name + top-level-domain

HTTP: primary communication protocol used to exchange content and data through WWW

HTML: web-page presentation

XML: vocabularies + meta-deta

<NITT>

<Batch2025>

<CSE-B>

<CR>Prajwal</CR>

</CSE-B>

</Batch2025>

</NITT>

Static web-content: static

Dynamic web-content: AJAX

**5) MULTI-TENANT TECHNOLOGY**

multiple users (tenants) to access the same application

tenants cannot access data that is not their own

UI, Business Protocols, Data Model, Access Control

Usage Isolation, Data Security, Recovery, Application Upgrades, Scalability, Metered Usage, Data Tier Isolation

**6) SERVICE TECHNOLOGY**

assist customer service teams

Web Services, REST Services, Service Agents, Service Middleware

1) Web Services -> WSDL (Web Service Description Language) -> defines API (Application Program Interface) of a web service.

XML Schema Definition Language -> messages exchanged by web services are expressed in XML SDL

Message format -> SOAP (Simple Object Access Protocol)

UDDI (Universal Description, Discovery and Integration) -> regulation

2) REST (Representational State Transfer) Services

guide the design and development of architecture for www: defines set of constraints: client-server, stateless, cache, interface / uniform contract, layered system, code-on-demand

*Client-Server*: existence of client, based on the principle of separation of concerns

*Stateless*: perspective of a server, it should not remember the state of the application, all information from client must be present in the message: improved visibility, reliability, scalability

*Cache*: classify responses into cachable and non-cachable, if cachable then store it in client-cache else don’t: improves scalability, efficiency, user-perceived performance, and reduces reliability

*Uniform Interface*: interface must be as generic as possible, simplifies and decouples the architecture, improves visibility, reduces efficiency

*Layered System*: many layers between client and server known as intermediaries, new ones can be added at various points, they help in translation and caching.

Intermediates include proxies (client) and gateways (origin server).

Improves scalability, improves security by introducing firewall, but adds overhead and latency.

*Code on Demand*: optional, allows client to download and execute a code from a server

3) SERVICE AGENTS

event driven programs, designed to intercept messages at runtime

Active -> perform actions on reading message contents

Passive -> do not change message contents

4) MIDDLEWARE SERVICES

MOM -> Message Oriented Middleware -> help in integration

ESB (Enterprise Service Bus): routing, service brokerage, message queing, Orchestration Platform