UNIT IV · LESSON 10

Cloud Computing

characteristics · service & deployment models · virtualisation · architecture · storage · security · providers · emerging trends

30
questions in Unit IV
OS · Networks · Cloud

NIST DEFINITION

- On-demand network access to a **shared pool** of configurable computing resources, rapidly provisioned and released with minimal management effort

Essential characteristic	Meaning
On-demand self-service	provision without human interaction
Broad network access	via standard devices / networks
Resource pooling	multi-tenant, location independence
Rapid elasticity	scale out/in quickly, appears unlimited
Measured service	metered, **pay-per-use**

- Roots: utility computing, grid computing, virtualisation, SOA, web services

WHO USES DEPLOYMENT MODELS

Model	Owned / used by	Note
Public	provider, open to all	economies of scale
Private	one organisation	control, security, costly
Community	organisations with shared concerns	shared cost
Hybrid	public + private bound together	**cloud bursting**
Multi-cloud	several public providers	avoid lock-in

SPI SERVICE MODELS

Model	Provider manages	User manages	Examples
IaaS	hardware, network, storage, virtualisation	OS, runtime, apps, data	AWS EC2, Azure VMs, Google Compute Engine
PaaS	+ OS, runtime, middleware	apps, data	Google App Engine, Heroku, Azure App Service
SaaS	everything	use / configure only	Gmail, Salesforce, Office 365, Dropbox

- Others: **FaaS / serverless** (AWS Lambda), DBaaS, STaaS, XaaS
- Control decreases IaaS → PaaS → SaaS; convenience increases

TRADE-OFF BENEFITS & CHALLENGES

- Benefits: lower **CapEx** (→ OpEx), scalability, availability, disaster recovery, global reach, faster deployment
- Challenges: **security & privacy**, data location/compliance, **vendor lock-in**, outages, latency, bandwidth cost, limited control
- **SLA**: uptime (99.9 % = 8.76 h downtime/year; 99.99 % = 52.6 min/year), penalties

COMPARE VMS VS CONTAINERS

Virtual machine	Container
own guest OS	shares host kernel
GBs, boots in minutes	MBs, starts in seconds
strong isolation	lighter isolation
hypervisor	container engine (Docker)

- **Kubernetes** orchestrates containers (pods, nodes, scaling, self-healing)
- **Microservices**: small independent services, often in containers

STORAGE TYPES CLOUD STORAGE & DATA

Type	Example
Object storage	Amazon **S3**, Azure Blob, Google Cloud Storage
Block storage	Amazon EBS
File storage	Amazon EFS, Azure Files

FOUNDATION VIRTUALISATION

- Abstract physical resources into multiple **virtual machines (VMs)**
- **Hypervisor / VMM**:
- **Type 1 (bare metal)**: VMware ESXi, Microsoft Hyper-V, Xen, KVM
- **Type 2 (hosted)**: VirtualBox, VMware Workstation

Technique	Idea
Full virtualisation	unmodified guest OS; binary translation
Para-virtualisation	guest OS modified to call hypervisor (Xen)
Hardware-assisted	Intel VT-x, AMD-V
OS-level (containers)	share host kernel: **Docker**, LXC

- Types by resource: server, storage (SAN/NAS), network (SDN, NFV), desktop (VDI), application
- **Live migration** moves a running VM between hosts

LAYERS CLOUD ARCHITECTURE

- Front end (clients, browsers) ↔ network ↔ back end (servers, storage, management)
- **Data centre** → virtualisation layer → resource management → service layer
- **Load balancing**, **auto-scaling**, **multi-tenancy**, **orchestration**
- **Service-oriented architecture (SOA)**, REST/SOAP web services, APIs
- Scaling: **vertical** (bigger machine, scale up) vs **horizontal** (more machines, scale out)

STORAGE TYPES CLOUD STORAGE & DATA (CONTD.)

Type	Example
NoSQL databases	DynamoDB, Bigtable, MongoDB Atlas
Big data	**Hadoop** (HDFS + **MapReduce**), Spark

- **MapReduce**: map (key, value) pairs → shuffle → reduce; developed at Google
- **CAP theorem**: a distributed store cannot guarantee Consistency, Availability and Partition tolerance all together
- Eventual consistency in many NoSQL stores

SHARED RESPONSIBILITY CLOUD SECURITY

- Provider secures **the cloud** (physical, hypervisor); customer secures **in the cloud** (data, IAM, apps) — share depends on IaaS/PaaS/SaaS
- **IAM** (identity & access management), MFA, encryption at rest & in transit, key management (KMS)
- Threats: data breach, misconfiguration, insecure APIs, account hijacking, insider threats, DoS, VM escape
- Compliance: ISO 27001, GDPR, HIPAA; audits

SERVICES MAJOR PROVIDERS

Provider	Compute	Storage	Serverless
AWS (2006)	EC2	S3, EBS	Lambda
Microsoft Azure	Virtual Machines	Blob	Functions
Google Cloud	Compute Engine	Cloud Storage	Cloud Functions
IBM, Oracle, Alibaba	—	—	—

- Open-source cloud platforms: **OpenStack**, CloudStack, Eucalyptus

EDGE & MORE RELATED MODELS

- **Edge / fog computing**: process near data source (IoT) → low latency
- **Mobile cloud**, **serverless**, **cloud-native** (DevOps, CI/CD)
- **Grid** computing: loosely coupled resources across organisations for big jobs
- **Cluster**: tightly coupled machines in one place
- Green cloud: energy-efficient data centres (PUE = total facility power / IT power)

Q → A RAPID RECALL

Model giving VMs to users	IaaS
Gmail is an example of	SaaS
Google App Engine is	PaaS
Pay-as-you-go characteristic	measured service
Bare-metal hypervisor type	Type 1

Q → A RAPID RECALL (CONTD.)

VirtualBox hypervisor type	Type 2
Containers share	host OS kernel
Public + private combination	hybrid cloud
Amazon S3 is	object storage
Programming model of Hadoop	MapReduce
Adding more servers	horizontal scaling (scale out)
Open-source IaaS platform	OpenStack

TRAP EXAM TRAPS

- Elasticity (automatic, short-term) ≠ scalability (capacity to grow)
- Containers are **not** VMs: no guest OS
- In SaaS the customer still owns responsibility for its **data** and user access
- Private cloud may be hosted off-site by a third party
- Grid ≠ cloud: grid shares resources for jobs, cloud sells services on demand