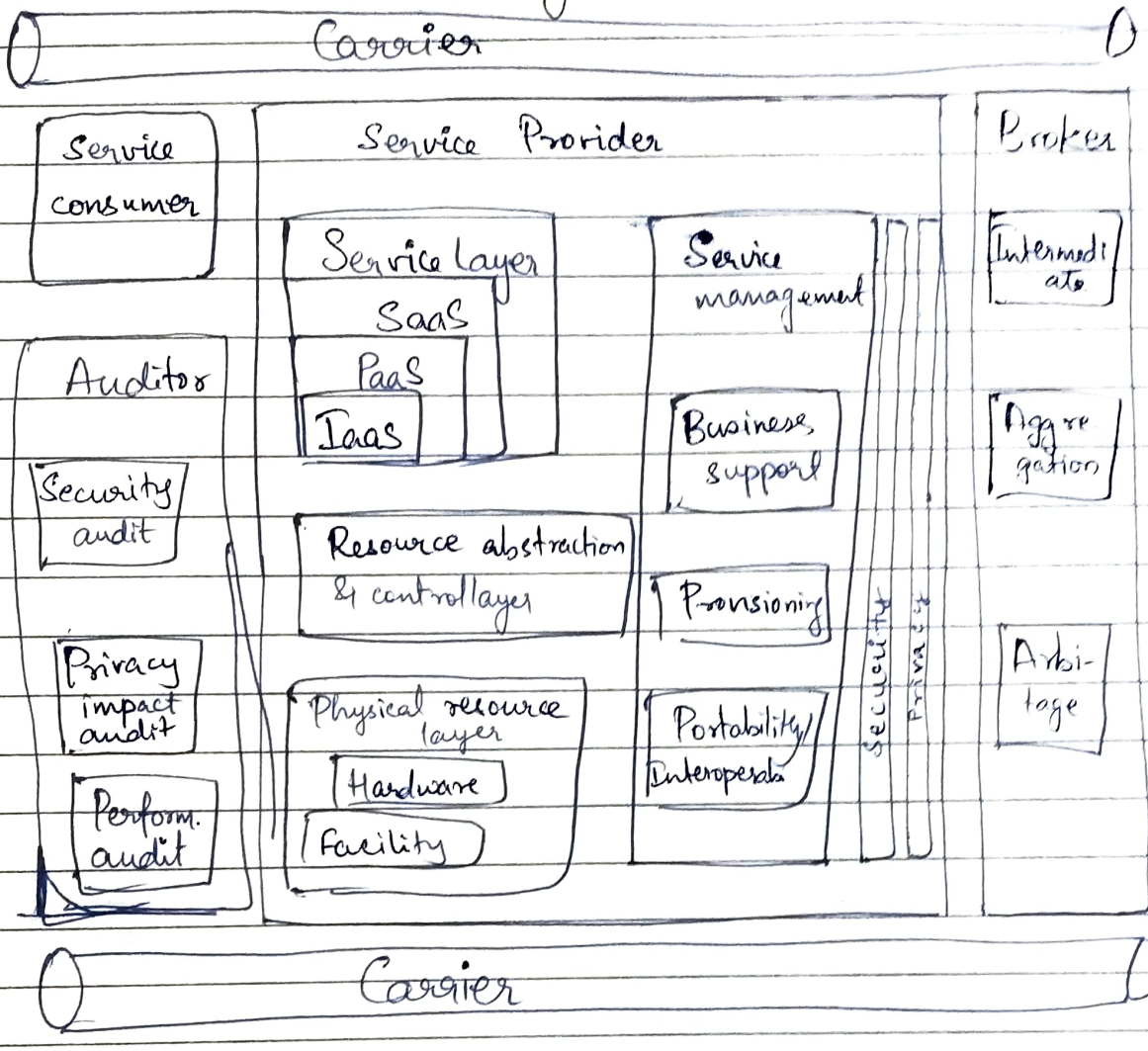
UNIT-1

Experiment No. :..... Date :.....

Q. NIST Cloud Computing Model



* entities involved in cloud computing are -

(i) Service Consumer - entity that maintains a business relationship with and uses service from service providers.

(ii) Service Provider - entity responsible for making a service available to service consumers.

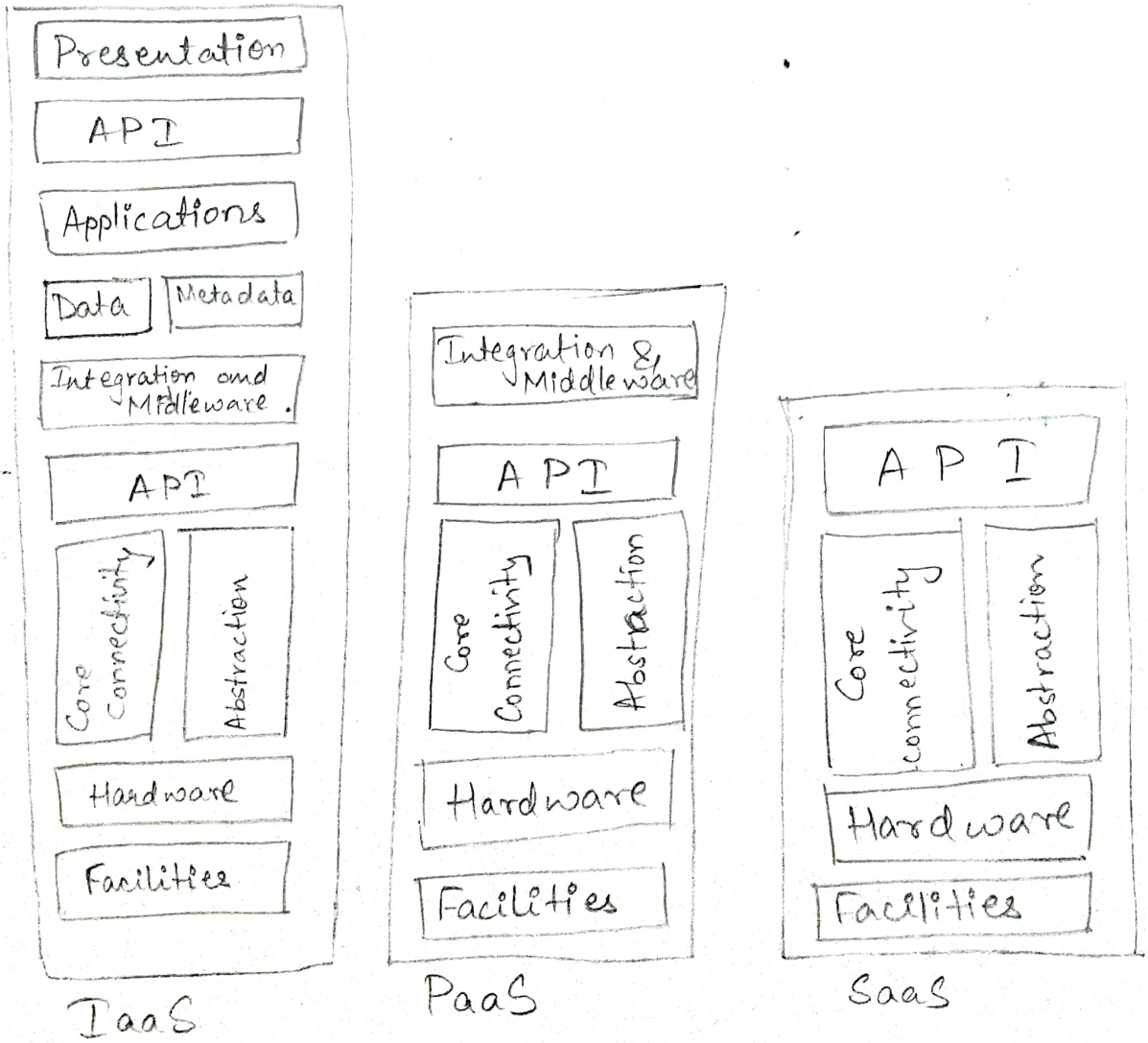
(iii) Carrier - the intermediate that provides connectivity and transport of cloud services between provider and consumers.

(iv) Broker - entity that manages the use, performance and delivery of cloud services and negotiates relation between provider and consumer.

(iv) **Auditor** - entity that can conduct independent assessment of cloud services, information system operations, performance, and security of cloud implementation.

* Security audit - evaluates cloud security
* Performance audit - evaluates cloud performance
* Privacy-impact audit - evaluates cloud privacy assurance

Q. Cloud Computing Delivery Models



Experiment No. :..... Date :.....

(i) Software-as-a-Service [SaaS] -
- applications are supplied by the service provider
- user doesn't manage or control cloud infrastructure, or individual application capabilities
- not suitable for real-time applications or for those where data is not allowed to be hosted externally
- Eg: Gmail, Google Search engine
- Services included:
  1. Web 2.0 - metadata management, blogs, wiki, social media, etc.
  2. Enterprise services - workflow management, supply chain, customer resource management, desktop software, etc.

(ii) Platform-as-a-Service [PaaS] -
- allows cloud user to deploy consumer-created or acquired applications using programming languages and tools supported by the service provider.
- not useful when:
  * application must be portable
  * hardware and software must be customized to improve the performance of the application.
- Allows user-
  * to control over the deployed application and application hosting environment configurations.
  * doesnot allow user to manage / control the underlying cloud infrastructure.

3.

(iii) Infrastructure-as-a-Service [IaaS] -

- user is able to deploy and run arbitrary software, which can include OS and applications.
- user doesnot manage / control cloud infrastructure but has control over OS, storage, deployed apps & limited control of some networking components.
- Services offered - server hosting, web servers, storage, OS, virtual instances, internet access, etc.

Q. NETWORK CENTRIC CONTENT & NETWORK CENTRIC COMPUTING. + Adv. of Network Centric

(1) Network Centric Content -

- content: any type of media or of any volume, static or dynamic, modular or monolithic, stored or live, aggregated or mixed.
- "Future Internet" will be content-centric.
- Creation of audio or visual and its consumption is likely to transform the internet to support increased quality in terms of resolution, frame rate, color depth, etc.

Network Centric Computing -

- information computing can be done more efficiently on large farms of computing and storage systems.
  * Grid Computing - targeted at scientific computing
  * Utility Computing - targeted at enterprise computing
- focus of utility computing is on business model for providing computing services.
- embraced by major IT companies - Amazon, HP, IBM, etc.

Experiment No. :..... Date :.....

Similarities between Network Centric Content & Computing

* **Data intensive** : multimedia streaming transfers large volume of data.
* **Network intensive** : transfering large volumes of data requires high bandwidth network.
* **Thin Clients** : systems accessed using thin clients running on systems with few and limited resources.
* **Workflow Management** : applications infrastructure should support workflow management.
* **Low Latency Network** : low latency networks for data streaming, parallel computing, computation steering.

Advantages of Network Centric :

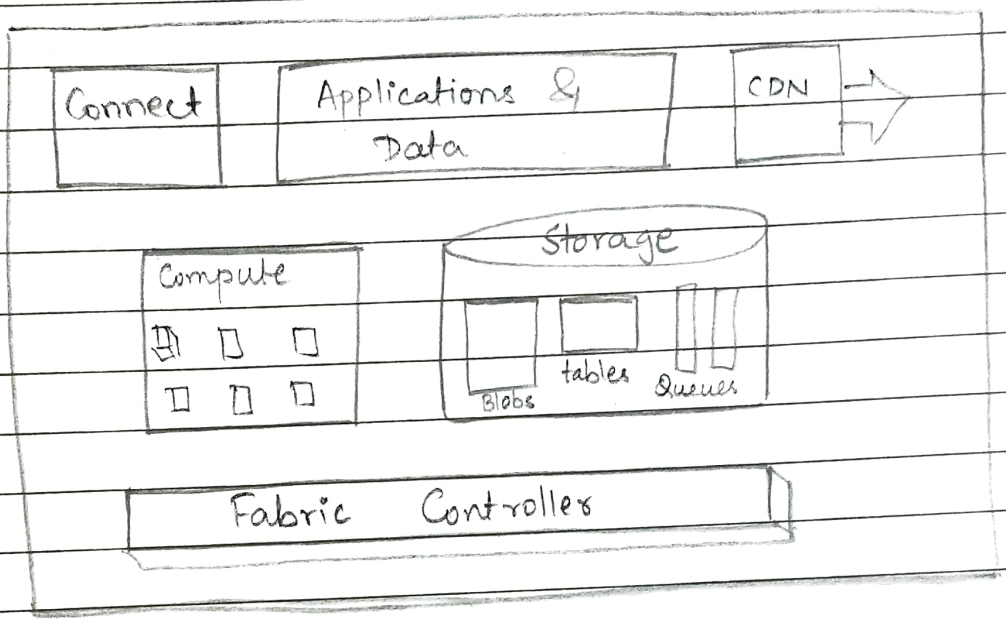
1. Computing and communication resources are shared and can be aggregated to support data intensive apps.
2. Data sharing ~~and~~ facilities and collaborative activities.
3. Cost reduction, i.e., pay as you go for computing.
4. User convenience and elasticity, i.e., able to accomodate increasing workloads.

Q. Responsibilities of User and Cloud Service Provider

* SaaS → Cloud service provider supplies both hardware and application software, & the user has direct access to these services through Web interface & no control over cloud resources.
  Eg: Gmail, GDocs,
* PaaS → provides only a platform including hardware and system software (OS, dB). Service provider is responsible for system updates, patches & s/w maintenance
  Eg: Google App. Engine, Microsoft Azure.
* IaaS → Service provider supplies the hardware (server, storage, network) and system software (OS, dB) and ensures system attributes (security, load balancing, fault tolerance).
  Eg: AWS.

Experiment No. :..... Date :.....

Q. MICROSOFT AZURE ARCHITECTURE :



* Azure is PaaS cloud platform from Microsoft.
+ Windows Azure → It is an operating system.
SQL Azure → cloud based version of SQL Server.
Azure AppFabric → collection of services for cloud appl".
* Windows Azure : has 3 components.
(i) Compute → provides a computation environment
(ii) Storage → for scalable storage
(iii) Fabric Controller → deploys, manages & monitors applications.
↳ interconnects nodes consisting of servers, high speed connections and switches.

→ CDN maintains cache copies of data to speedup computation.
→ The Connect subsystem supports IP connections b/w the users and their applications running on Windows Azure
→ The API interface is built on REST, HTTP and XML.
It includes 5 services:

Live services	AppFabric	Dynamics CRM.
SQL Azure	SharePoint	

Q. AWS Services.

(1) Amazon AWS Management Console –

allows users to access services offered by AWS.

(2) EC2 –

→ elastic cloud computing is a web service for launching instances of an application under several OS (Linux, Windows, Open Solaris, FreeBSD, NetBSD).

→ a user can load EC2 instance with a custom application environment.

→ a user can manage network's access permissions.

→ a user can run the image using as many or as few systems as desired

→ users can access images provided by Amazon

→ users can customize an image & store in S3.

→ Resources provided by EC2 →

(i) Virtual Computers VC – virtual systems running the instance

(ii) Compute Units CU – measure computing power of each system

(iii) Memory

(iv) I/O Capabilities.

→ EC2 instances boot from an AMI digitally signed and stored in S3.

→ Instance Types in EC2 –

(i) Standard instances → StdM (micro), StdS (small), StdL (large), StdXL (extra large). Small is default

(ii) High Memory instances. → HmXL, Hm2XL, Hm4XL

(iii) High CPU instances → HcpuXL.

(iv) Cluster Computing → Cl4XL.

Experiment No. :..... Date :.....

(3) Simple Storage System – S3

* service designed to store large objects. 1 byte – 5 TB.
* an object is stored in a bucket and retrieved via a unique developer-assigned key. Buckets are stored in a Region selected by the user.
* Object names are global
* Supports minimal set of func^n – read, write, delete.
  Doesn't support – copy, rename, move to another bucket.
* authentication mechanisms ensure data is kept secure.
* S3 computes MD5 of every object written & returns it in a field called ETag.
* Objects can be made public, & rights can be granted to other users.

(4) Elastic Block Store – EBS

* provides persistent block level storage volumes for use with EC2 instances.
* Suitable for database applications, file systems and applications using raw data devices.
* Volume appears as raw, unformatted & reliable physical disk to an application (1GB – 1 TB)
* An instance may have multiple volumes, but a volume cannot be shared among multiple instances.
* The volumes are grouped together in Availability Zones and are automatically replicated in each zone.

(5) Simple Queue Service – SQS.

* Hosted message queues are accessed through standard SOAP and Query interfaces.
* Supports automated workflows - EC2 instances can coordinate by sending & receiving SQS messages
* Applications using SQS can run independently and asynchronously, and donot need to be developed with the same tech.
* A receive message is "locked" during processing. If process fails, the lock expires & message is available again.
* Queue sharing can be restricted by IP addr and time-of-day

(7) List of AWS Services –

Till 2011

1. EC2
2. S3
3. EBS
4. SQS
5. Simple DB
6. CloudWatch
7. Auto Scaling
8. Virtual Private Cloud

added in 2012

9. Route53
10. Elastic MapReduce (EMR)
11. Simple Workflow Service (SWS)
12. ElastiCache
13. Dynamo DB.
14. Cloud Front
15. Elastic Load Balancer
16. CloudFormation

(8) Elastic Beanstalk –

* Handles deployment, capacity provisioning, load balancing, autoscaling, & monitoring functions
* service available using – Java platfor, PHP server side lang, .NET framework
* Management functions provided –
  1. Deploy new application version
  2. Access to results reported by CloudWatch
  3. Email notifications when appl^n status changes
  4. Acess to server log files.

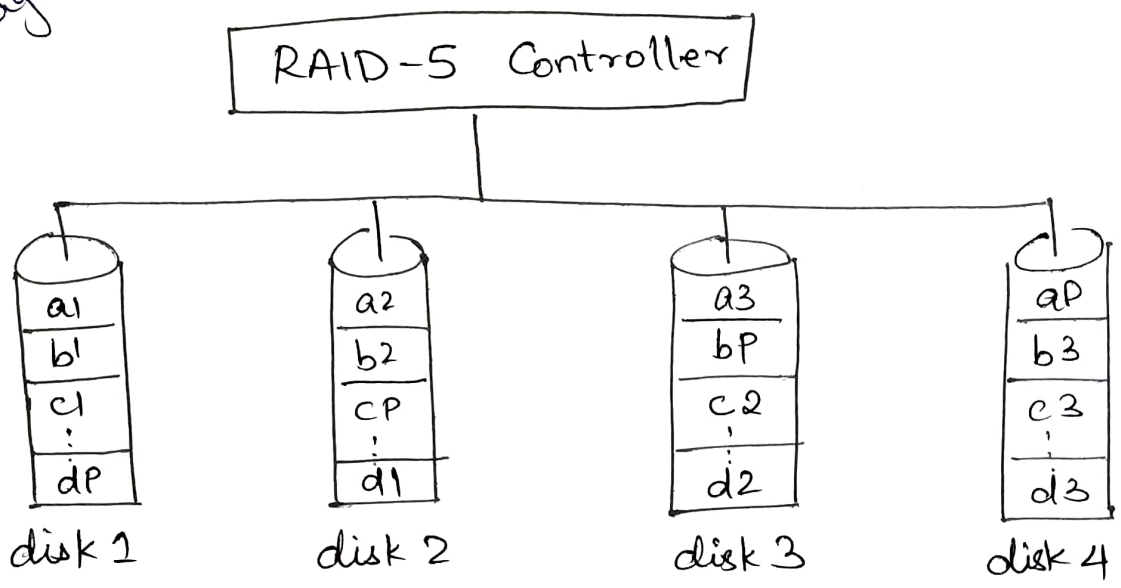
Experiment No. :..... Date :.....

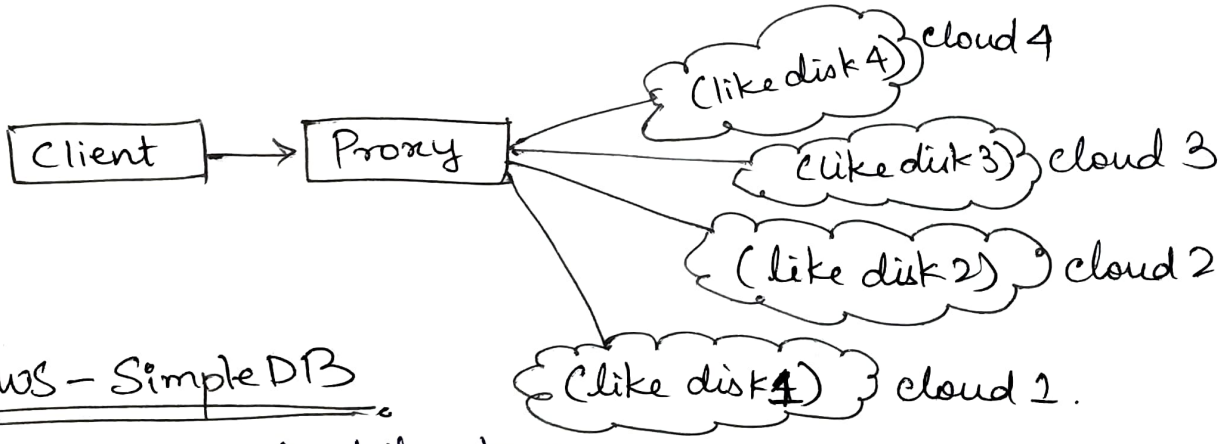
Q. RAID-5.

* A RAID-5 system uses block level stripping with distributed parity over a disk array.
* The disk controller distributes the sequential blocks of data to the physical disks and computes a parity block by bitwise XOR-ing of the data blocks.
* The parity block is written on a different disk for each file to avoid the bottle-neck possible when all parity blocks are written to a dedicated disk.
* This technique allows to recover data after a single disk loss.
* The system strips the data across 4 clusters.
* The access to data is controlled by PROXY that carries out some of the func^n of a RAID controller as well as authentication and other security func^{n}s.
* Proxy ensures before and after atomicity as well as all-or-nothing for data access.
* Proxy buffers, converts data manipulation commands, optimized data access, converts data to format specific to each cloud.

+ diagram.

RAID-5 Diagram.





Q. AWS – SimpleDB

* non relational data store
* Supports store & query func's. traditionally provided only by relational databases.
* Supports high performance Web applications.
* Users can store & query data via Web service requests.
* Creates multiple geographically distributed copies of all data
* Manages →
  (i) infrastructure provisioning
  (ii) hardware & software maintenance
  (iii) replication & indexing of data items.
  (iv) performance tuning