1. What is Clustering?

Clustering refers to connecting multiple **nodes** (computers/servers) to work together as a single system to improve performance, availability, and scalability.

Types:

Physical clustering = Dedicated hardware for maximum control;

Virtual clustering = Shared resources for maximum flexibility.

Physical Clustering

- This involves connecting **physical machines** via a physical network like a LAN.
- Used in environments where dedicated hardware is required for high performance or security.

Virtual Clustering

- A **virtual cluster** is formed by grouping multiple **virtual machines (VMs)** running on one or more physical servers.
- The VMs can be distributed across different physical machines but work together logically.

Real-life example:
A cloud provider like **Amazon Web Services (AWS)** uses thousands of physical servers. On these, they create VMs that form virtual clusters for different customers—like one cluster for a web hosting company and another for a machine learning startup.

3. Problems in Virtual Clustering (and Solutions)

① Live Migration of VMs

- Moving a running VM from one physical host to another without downtime.

Real-life example:
If a physical server needs maintenance, VMs can be **live-migrated** to another server without shutting down the services—like moving a running app from one phone to another without closing it.

② Memory and File Management

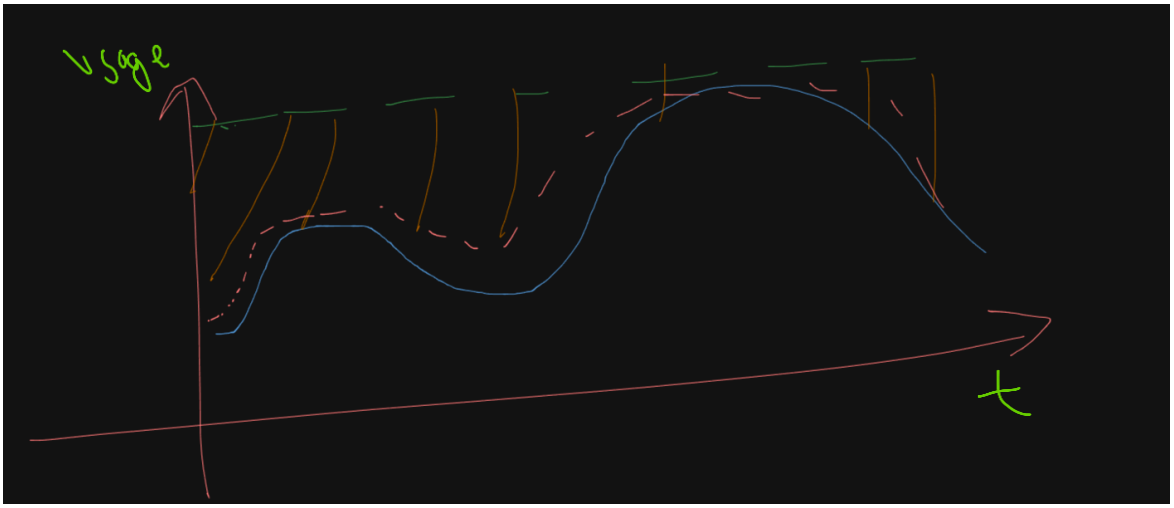
- Efficiently allocating memory and managing files across VMs.

Real-life example:
In a **virtualized database cluster**, memory must be shared wisely so that one VM doesn't hog resources and slow down others

③ Dynamic Deployment of Virtual Clusters

- Deciding which VM should run on which physical cluster based on load, location, or cost.

Real-life example:
Netflix uses dynamic clustering to deploy streaming VMs closer to users during peak hours—ensuring smooth playback without buffering.



*Traditional VMs require manual setup of:

- OS
- Memory
- Libraries
- Applications

This is time-consuming and error-prone.

Real-life example:
Before virtualization, a company setting up a new server had to manually install Windows, configure RAM, install SQL Server, and set up the application—a process that could take days. Now, with **templates and automation**, a VM can be spun up in minutes.

Key Aspects of Resource Management:

① High Performance (HP) Clusters

- HP clusters use computer clusters and supercomputers to solve advance computational tasks like scientific research, simulations, and AI training.

Real-life example:
NASA uses HP clusters to process satellite imagery and run climate models, where thousands of CPUs work in parallel to solve complex equations.

② Load-Balancing Clusters

- Distributes incoming traffic across multiple nodes to prevent overload.

Real-life example:
Google uses load-balancing clusters to distribute search queries across thousands of servers—so no single server gets overwhelmed, and responses are fast.

③ High Availability (HA) Clusters

- Ensures services remain available even if a node fails, using redundant backups.

Real-life example:
Online banking systems use HA clusters so that if one server fails, another immediately takes over without customers noticing any interruption.

What is HPC?

High-Performance Computing (HPC) uses **parallel processing** to run advanced applications much faster than a single computer could. Think of it as organizing **thousands of workers to complete a massive project simultaneously** rather than one person working alone.

Key Performance Metric:

- **Teraflop:** 10^{12} floating-point operations per second (baseline for HPC)
- **Petaflop:** 10^{15} operations per second (modern supercomputers)
- **Exaflop:** 10^{18} operations per second (next frontier)

Real-life analogy:
If a regular computer is like 2-4 cores, HPC ranges form as few as 4 nodes on 16 cores. Common cluster size in many businesses is between 16 & 64 cores or from 64 to 256 cores.

How Does HPC Work?

To create a high-performance computing architecture, multiple computer servers are networked together to form a compute cluster. Algorithms and software programs are executed simultaneously on the servers, and the cluster is networked to data storage to retrieve the results. All of these components work together to complete a diverse set of tasks.

To achieve maximum efficiency, each module must keep pace with others, otherwise, the performance of the entire HPC infrastructure would suffer.

Advantages of Cloud HPC

1. Cost & Financial

- **No upfront investment**
- **Pay-per-use** pricing
- **Cost-effective** scaling
- **No maintenance** costs

2. Scalability & Flexibility

- **On-demand** resources
- **Global availability** (any data center)
- **Hardware variety** (latest CPUs/GPUs)

3. Performance & Speed

4. Global access

5 Reliability & Resilience

- **High availability**
- **Fault tolerance (automatic recovery)**

6. Security

- **Encryption (at rest, in transit, in use)**
- **Private networking (VPC/VPN)**
- **Access controls (IAM)**
- **Audit logging**
- **Security updates automated**

Cloud Application considerations for backup, disaster recovery, fault tolerance

1.Backup:

a strategy for sending a copy of a physical or virtual file or database to a secondary, off-site location for preservation in case of equipment failure, site catastrophe or human malfeasance.

Types of backups:

- **Full Back up:** On **Sunday**, you photocopy the *entire notebook*.
  - copy the entire data set every time a backup is initiated. As a result, they provide the highest level of protection, though time and storage consuming
- **Incremental Backup:** On **Monday**, you photocopy *only Monday's changed pages*. On **Tuesday**, you photocopy *only Tuesday's changed pages*. On **Wednesday**, you photocopy *only Wednesday's changed pages*.
  - only back up the data that has been changed or updated since the last backup increment -- not the last full backup.
- **Differential Backup:** On **Monday**, you photocopy *all pages changed since Sunday*. On **Tuesday**, you photocopy *all pages changed since Sunday* (including Monday's changes). On **Wednesday**, you photocopy *all pages changed since Sunday* (including Mon + Tue changes).
  - differential backups back up data that has changed since the last full backup, rather than the last backup in general

2. Disaster Recovery

The ability to restore IT systems in the cloud after a disaster, eliminating the need for a physical secondary data center.

Stratergies:

❄ COLD DISASTER RECOVERY

Start from scratch → build everything → configure → connect.

A cold DR site provides only basic infrastructure with no pre-installed systems or active data. During a disaster, the entire environment—servers, software, and data—must be rebuilt from the beginning, leading to long recovery times but very low cost.

⛅ WARM DISASTER RECOVERY

Build basic setup → keep updating/replicating periodically → some parts are running.

A warm DR site has partially prepared systems with hardware and software already installed and data updated at intervals. Systems are not fully active, but can be brought online quickly by starting VMs, synchronizing recent data, and redirecting traffic.

🔥 HOT DISASTER RECOVERY

Everything is already built, running, updated — nothing to build or start.
Just switch traffic and continue instantly.

A hot DR site is a fully operational, real-time mirror of the primary environment. All systems and data are continuously synchronized, allowing instant failover with almost no downtime or data loss.

3) Fault Tolerance:

isolating and then working on that fault
make sure no single point of failure to ensure no cascading

Creating a blueprint for continuous work when some components fail or become unavailable. It assists businesses in assessing their infrastructure needs and requirements, as well as providing services if the relevant equipment becomes unavailable for whatever reason. The capacity of an operating system to recover and recognize errors without failing can be can be achieved by hardware, software, or a mixed approach that uses load balancers. As a result, fault tolerance solutions are most commonly used for mission-critical applications or systems.

Main Concepts behind Fault Tolerance in Cloud Computing System

- **Redundancy:** When a system or system part fails or goes down, it is critical to have backup systems

● **Replication:** Load balancing- The fault-tolerant system operates on the principle of running multiple replicas for every service. As a result, if one component of the system fails, additional instances can be used to maintain it operational.

How does a cloud platform work?

1. Cloud platforms create a virtual pool of shared resources to provide compute, data storage, and network services over the internet.
2. Customers can access resources on the cloud platform as needed, paying only for the resources they need.
3. Cloud platforms use virtualization technologies that create multiple virtual machines, or VMs, on a single server, making it possible to run separate operating systems and applications for various customers on one physical server.
4. Customers may access compute services from both public and private cloud platforms.
5. Cloud platforms allow organizations to store, back up, and recover data; test and build apps; access cloud databases; analyze big data sets; deliver software on demand on a global scale; access business intelligence; and create cloud-native applications.

What is Cloud Connect?

- **Cloud Connect** is a solution designed to ensure that employees and systems can access cloud services **reliably and seamlessly**, without interruptions.
- It provides a **dedicated, private connection** between an organization's infrastructure and the cloud provider.
- Unlike traditional access, which relies on the **public internet**, Cloud Connect bypasses it to reduce risks like lag, downtime, or unstable connections.

⚙️ **How Cloud Connect Works (from your page)**

1. **Dedicated Pathway:** Instead of routing traffic through the public internet, Cloud Connect establishes a **direct private link** to the cloud provider.
2. **Reliability:** This ensures that critical services (like SaaS applications, backups, or databases) are always accessible.
3. **Performance:** By avoiding congestion and variability of the public internet, Cloud Connect delivers **faster speeds and lower latency**.
4. **Security:** A private connection reduces exposure to cyber threats compared to public internet routes.

Cloud printing is a service that allows users to print documents from **any device connected to the network** (smartphones, laptops, tablets, desktops).

App Engine ON Cloud

The App Engine SDK facilitates the testing and professionalization of applications

by emulating the production runtime environment and allowing developers to

design and test applications on their own PCs

The App Engine SDK facilitates the testing and professionalization of applications by emulating the production runtime environment and allowing developers to design and test applications on their own PCs. When an application is finished being produced, developers can quickly migrate it to App Engine, put in place quotas to control the cost that is generated, and make the programmer available to everyone. Python, Java, and Go are among the languages that are currently supported.

- **Google App Engine (GAE)** is a **fully managed Platform as a Service (PaaS)**.
- It provides a **scalable runtime environment** for running web and mobile applications.
- Developers focus on building apps, while Google handles infrastructure, scaling, and reliability.

🔹 **Core Capabilities**

- **Dynamic Scaling**: Apps automatically scale up/down with demand.

- **Secure Execution Environment**: Hosted on Google's highly secure servers.
- **App Engine SDK**: Lets developers emulate production locally for testing.
- **Language Support**: Python, Java, and Go.
- **Built-in Services**: Cron jobs, communication tools, scalable data stores, work queues, caching.

🔹 **Advantages**

1. **Security Infrastructure** – Strong protection against unauthorized access.
2. **Built in features**- Dialogflow, firebase- auth, firestore- db, vertex ai etc.
3. **Faster Time to Market** – Quick deployment accelerates product release.
4. **Quick Start** – No hardware setup required.
5. **Ease of Use** – Integrated tools for building, testing, launching, updating.
6. **Rich APIs & Services** – Enables feature-rich applications.
7. **Scalability** – Uses Google's internal tech (e.g., GFS, BigTable).
8. **Performance & Reliability** – Backed by Google's global infrastructure.
9. **Cost Savings** – No need for server management staff.
10. **Platform Independence** – Easy migration due to minimal dependencies.
    - ➢ Users can route traffic to different app versions to A/B testing (It is a research methodology applicable in determining user experience.

What is A/B Testing?

A/B testing is an **experimental method** used to compare two versions of something (like a webpage, app feature, or marketing campaign) to see which performs better.

- **Version A** = the current or "control" version.
- **Version B** = the modified or "treatment" version.

📊 **Example**

Imagine you run an e-commerce site:

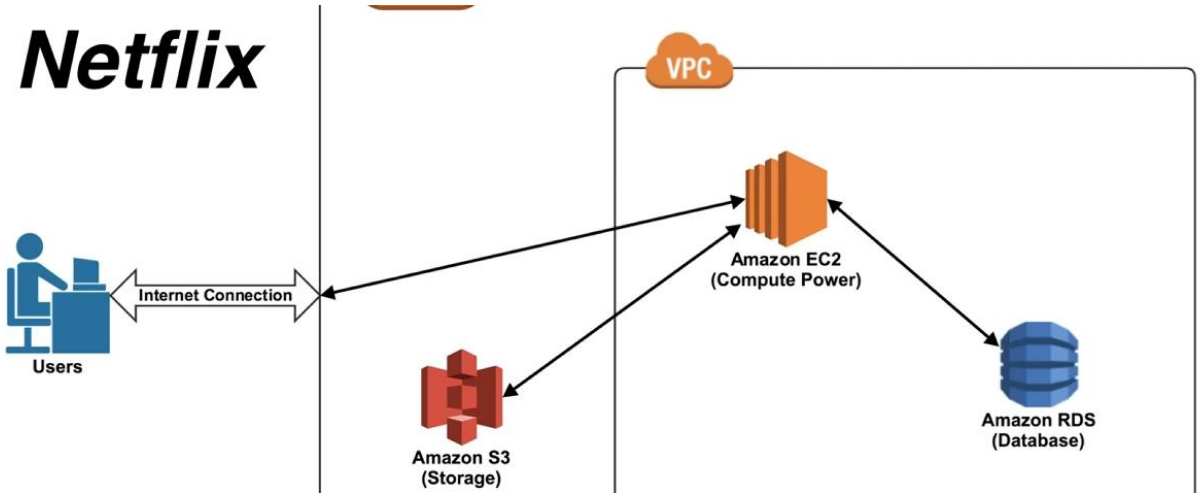
- **A (Control)**: "Buy Now" button is blue.

- **B (Variant)**: “Buy Now” button is green. If Version B leads to a **10% higher purchase rate**, you’d adopt the green button.

Elastic Compute Cloud

EC2 stands for **Elastic Compute Cloud**, an on-demand computing service on the AWS cloud.

It allows users to **rent virtual computers** with flexible configurations (RAM, storage, etc.).



Web Services on Cloud

Any software, application, or cloud technology that uses a standardized Web protocol (HTTP or HTTPS) to connect, interoperate, and exchange data messages over the Internet-usually XML (Extensible Markup Language) is considered a Web service.

🧩 Components of Web Services

1. SOAP (Simple Object Access Protocol)

SOAP is a **transport-independent** (SOAP messages can be sent using **different network protocols**, not only HTTP. {FTP, smtp etc.}) XML**-based messaging protocol** used to send structured information between a client and a server.
Each SOAP message is an XML document with:

- **Envelope** (root element)
    - **Header** → routing & metadata

- **Body** → the actual message content
  SOAP works over HTTP, making it compatible with standard web communication.

2. WSDL (Web Services Description Language)

WSDL is an **XML-based description file** that explains:

- What the web service does
- Where it is located
- How a client can access it

A client application reads the WSDL file to understand how to correctly invoke the service. Without WSDL, a web service cannot be implemented by a client.

3. UDDI (Universal Description, Discovery, and Integration)

UDDI is a **registry system** that allows:

- Publishing of web services
- Searching and discovering available services
- Storing WSDL files

It works like an **online directory** that tells clients where a web service is located and how to interact with it.

So wsdl is the directory and uddi is searching protocol