Data Center Architecture and Cloud Computing

Chapter-7

When designing a network for an organization, whether it's a small, medium, or large network, it's important to consider the following factors:

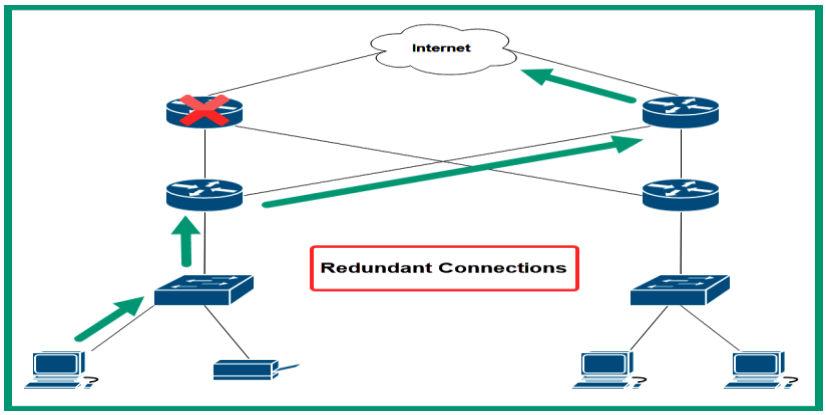
- Fault tolerance and redundancy
- Scalability
- Security
- QoS

Fault Tolerance

- Ability of a network/device to continue functioning even if a component fails.
- Ensures continuous service even during hardware failures like damaged routers, switches, or cables.
- Network must quickly detect failures and reroute traffic automatically.

Redundancy

- Achieved by adding **multiple paths** between source and destination devices.
- If one path or device fails, traffic is sent through alternative routes.
- Increases availability, reliability, and minimizes downtime.



The diagram shows that even if one router or link fails (marked with an X), the network continues forwarding traffic through alternate paths.

This demonstrates fault tolerance, where the network automatically reroutes data to maintain connectivity.

Scalability

- Scalability refers to the ability of a network to **grow and expand** without reducing performance.

- A scalable network can support **more users, devices, applications, and services** over time.
- Growth in organizations increases the number of **staff, computers, laptops, mobile devices**, etc.
- To handle this growth, network professionals add **more switches, routers, or access points**.
- Scalability ensures new additions **do not affect** the performance of existing network services.
- A scalable design reduces the need for major redesigns when the organization expands.
- It supports **future upgrades**, such as higher bandwidth, additional VLANs, or new servers.
- Helps maintain **consistent performance** as network traffic and device count increase.
- Important for long-term planning and cost-effective network management.

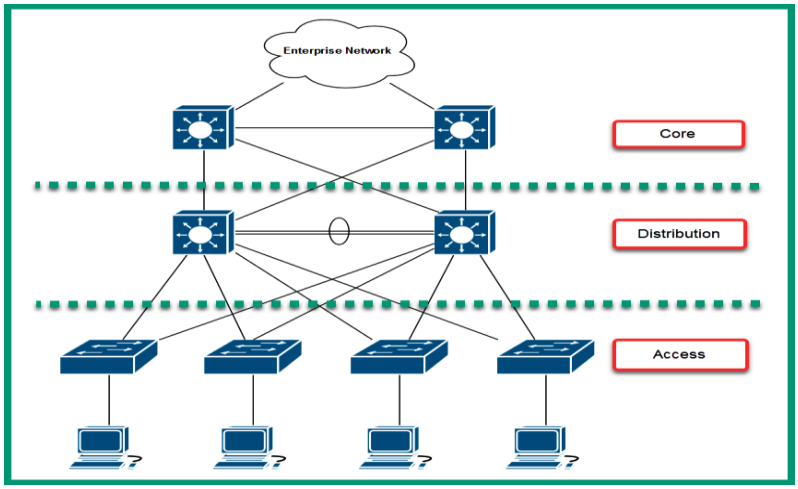
Security

- Security is essential for all organizations due to increasing cyberattacks and evolving threats.
- Network security protects critical **assets, data, and systems** from threat actors such as hackers.
- A secure network design includes layers of protection using various security devices and policies.
- Important security solutions include:
    - **Firewalls** – control incoming and outgoing traffic based on security rules.
    - **Intrusion Prevention Systems (IPS)** – detect and automatically block malicious activities.
    - **Endpoint security** – protects user devices like laptops and desktops from malware and attacks.
- Without security solutions, organizations may not detect intrusions until damage has already occurred.

Quality of Service (QoS)

- Quality of Service (QoS) is a network technology used to **prioritize bandwidth** for important traffic types.
- Network devices like routers and switches can be configured to give higher priority to specific applications.
- Many application-layer protocols use **TCP**, which provides **reliable and guaranteed delivery** between source and destination.
- Other protocols use **UDP**, which is **connectionless** and does **not guarantee delivery** of datagrams.
- During network congestion, **UDP packets are more likely to be dropped** because the protocol does not ensure recovery.
- Packet loss or delay in these applications results in **poor audio/video quality** and communication issues.
- Organizations use collaboration tools (e.g., telepresence) that require strict timing and minimal packet loss.
- QoS ensures these real-time applications receive **dedicated bandwidth and higher priority** over normal traffic.

Cisco 3-tier architecture



Layer 3 switches are implemented within the Distribution and Core layers of the network architecture as they provide both Layer 2 and Layer 3 functionality such as routing

Access Layer (Edge Layer)

- Provides **network connectivity** for end devices (PCs, laptops, printers, etc.).
- Uses **Layer 2 switches** (operating at OSI Layer 2).
- Responsible for allowing devices to access **network resources**.
- End devices usually have a **single NIC**, so **no redundancy** exists at this layer.
- Connects users directly to the network.

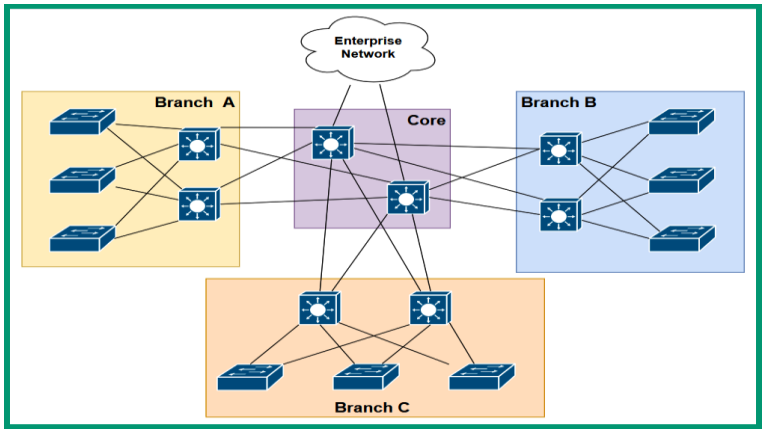
Distribution Layer (Aggregation Layer)

- Sits between the access layer and core layer.
- Provides **redundancy**, **load balancing**, and **link aggregation** for access layer switches.
- Each access switch connects to **both distribution switches** to avoid single points of failure.
- Handles **policy control**, including filtering, QoS, and routing between VLANs.
- Most traffic in a branch flows between **access ↔ distribution** layers.

Core Layer

- The **core layer** is the **high-speed backbone** of the network.
- Its main job is to provide **fast, reliable interconnectivity** between distribution layers.
- Ensures **high redundancy** and **high availability** across the entire organization.
- Carries traffic between **different branch offices** or major network segments.
- Optimized for **speed and efficiency**, avoiding complex processing or filtering.

Example



The diagram shows how **multiple branch offices (A, B, and C)** connect to a central **core layer** to form a scalable and redundant enterprise network.

1. Branch Offices (Branch A, B, and C)

Each branch has its own internal network consisting of:

- **Access layer switches** (bottom layer): connect end devices like PCs, printers, etc.
- **Distribution layer switches** (middle layer): connect access switches and provide redundancy.

Inside each branch:

- Every access layer device connects to **multiple distribution switches**.
- This provides **high availability**, ensuring that if one distribution switch fails, traffic can still flow through another.

2. Core Layer (Center of the Diagram)

- The core layer acts as the **high-speed backbone** of the entire enterprise network.
- It connects all branch offices together.
- The core switches have **multiple interconnections** with distribution switches in every branch.
- This ensures:
  - Redundancy
  - Fast data transport
  - Low latency
  - Smooth communication between branches

3. How Traffic Flows

- Traffic **inside a branch** moves between Access → Distribution → Access.
- Traffic **between branches** goes through the **Core**.

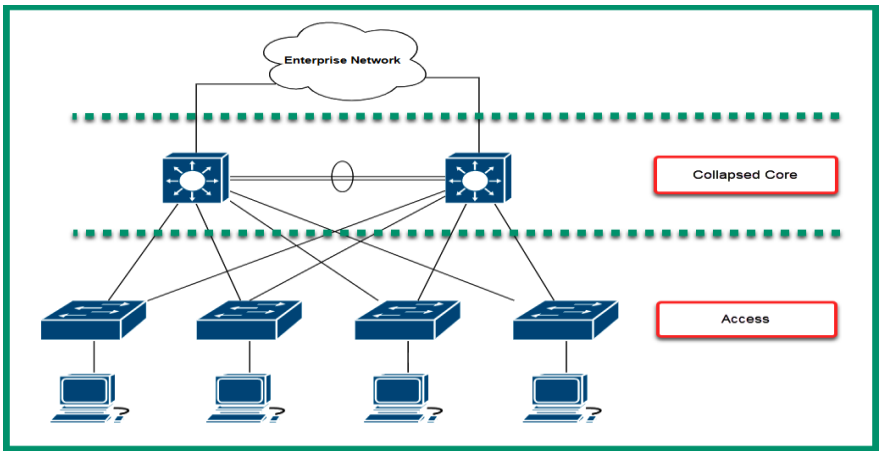
- Example: A device in Branch A sending data to Branch B first sends it to the distribution layer → core → distribution of Branch B → access layer → destination device.

Benefits

The **3-tier network architecture** provides several benefits for large organizations

- First, it uses a **multi-layered design**—core, distribution, and access layers—which improves **scalability**, **performance**, and **redundancy**.
- Second, it supports **modularity**, allowing network engineers to design repeatable and consistent building blocks across the organization.
- Finally, it removes the limitations of a **flat network**, enabling better fault isolation, improved traffic management, and reliable growth for enterprise networks.

Cisco 2-tier architecture



- The **2-tier architecture** has only the **collapsed core layer and access layer**, instead of three layers.
- It still provides key benefits like **scalability, fault tolerance, redundancy, security, and QoS**, similar to a 3-tier design, but in a **smaller and simpler network structure**.
- In a **2-tier network architecture**, the **core and distribution layers are combined** into a single **collapsed core layer**, which handles high-speed switching and routing.
- The **access layer** connects end devices to the network, and each access switch connects to all collapsed-core switches to ensure **fault tolerance and redundancy**.

Fundamentals of cloud computing

• Cloud computing is the practice of paying for and using computing resources (servers, storage, applications, services) from a **cloud service provider's data center**.

• Instead of owning and maintaining physical servers, organizations **rent resources** from providers such as:

- **Microsoft Azure**

- **Amazon Web Services (AWS)**
- **Google Cloud Platform (GCP)**

The term **cloud** refers to **remote resources** (servers, apps, services) hosted in a provider's d**ata center**. Customers usually **do not see or physically access** the hardware. Access is done **via the internet**, which is made of interconnected public networks managed by ISPs.

Pay-As-You-Go Model : Cloud services follow a **pay-per-use** approach: Pay **per minute** or **per hour**, depending on provider.

Tenant → A single customer using a cloud provider's platform.

Multi-tenant → Many customers share the same cloud provider's platform.

Tenant isolation → Even though customers share the same resources in the data center, each customer's data and services are kept separate and secure so they cannot access each other's resources.

Key Advantages

- **Faster deployment** (minutes instead of weeks).
- **Scalability** – easily add/remove resources.
- **Cost efficiency** – no upfront hardware costs, only pay for usage.
- **Reduced IT workload** – no need to manage physical servers.
- **Global access** – services accessible from anywhere via internet.

Scalability

Definition:
The ability of a system to **increase or decrease its capacity** (compute power, storage, network resources) to handle growing workload **in a planned and gradual manner**.

Key Points:

- Usually **manual or semi-automated**.
- Used for **long-term workload growth**.
- Involves adding/removing resources like VMs, CPUs, RAM.
- Common in enterprise planning (e.g., increasing server capacity at the end of each quarter).

Example:
An e-commerce company adds more virtual machines before a festival season because they expect more traffic.

Elasticity

Definition:
The ability of a system to **automatically scale resources up or down** in real-time **based on demand**.

- **Fully automated** (usually).
- Ideal for **unpredictable or sudden workload changes**.
- Resources expand instantly when workload increases and shrink when workload reduces.
- Minimizes cost by using only what is needed.

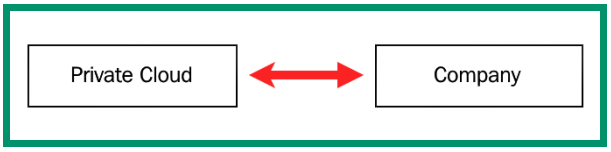
Example:
During a flash sale, when thousands of users suddenly visit the website, extra compute instances automatically spin up. After the sale ends, they automatically shut down.

Cloud Deployment Models

Cloud Deployment = Method of delivering cloud services to users.

When deploying a cloud computing solution for an organization, there are **four main deployment models**:

1. Private Cloud Model



- A **private cloud** is a cloud infrastructure that is **dedicated to a single organization**.
- It is **owned, managed, and operated** either by the organization's own IT team or by a third-party provider, but the resources are **not shared** with outsiders.
- Access is restricted to the employees or authorized users of that organization only.

Advantages

- Enhanced **security and privacy** (no resource sharing).
- **Better performance** since resources are not divided among multiple customers.
- **Regulatory compliance** (suitable for industries like banking, healthcare, defense).
- Full **control over infrastructure**.

Limitations

- High **initial cost** (servers, storage, networking).
- Requires **skilled IT staff** to manage.
- Less scalable compared to public cloud.

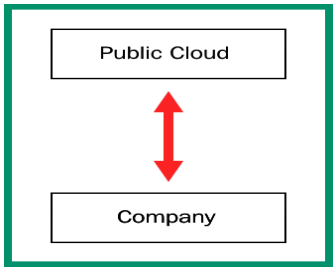
Example of Private Cloud

1. **Banking Sector**:
    - A bank sets up its **own private cloud** to host critical applications such as online banking, transaction processing, and customer databases.

    - Since banks deal with **highly sensitive financial data**, they cannot risk storing it on a public cloud.
    - The bank's IT department manages this private cloud in its own **secure data center**.
2. **Government Organizations**:
    - Defense or intelligence agencies use private clouds to store **classified data**, ensuring no external party has access.

Public Cloud Model

- A **public cloud** is a cloud infrastructure that is **owned and operated by third-party service providers** and is **available to anyone on the internet**.
- Resources like servers, storage, and applications are hosted in the provider's data centres and **shared among multiple users (multi-tenant environment)**.



- Customers access these resources on a **pay-as-you-go basis** without owning the underlying hardware.

Advantages

- **Low cost** (no need to buy servers or maintain infrastructure).
- **Easy scalability** to handle increased workload.
- **High reliability** with backup and disaster recovery handled by the provider.
- Accessible from **anywhere via the internet**.

Limitations

- **Less control** over infrastructure compared to private cloud.
- **Security concerns**, since resources are shared among multiple customers.
- Possible **compliance issues** for industries with strict regulations (like healthcare or defense).

Examples of Public Cloud Providers

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- IBM Cloud

Real-Time Examples of Public Cloud Model

1. Google Drive / Google Photos

- Millions of users store files, photos, and documents.
- All resources (storage, servers, security) are owned and managed by Google.
- Users just pay or use free storage.
- **Real-time example:** *Google Drive is a public cloud service where users store and access files over the internet using Google's shared cloud infrastructure.*

2. Amazon Web Services (AWS)

- Services like EC2, S3, Lambda used by companies like Netflix.
- AWS manages all infrastructure; customers only rent resources.

3. Microsoft Azure

- Provides virtual machines, databases, analytics, and AI services.
- Used by startups and enterprises worldwide.

4. Dropbox

- Cloud storage service where the infrastructure is owned by Dropbox.
- Shared among millions of users.

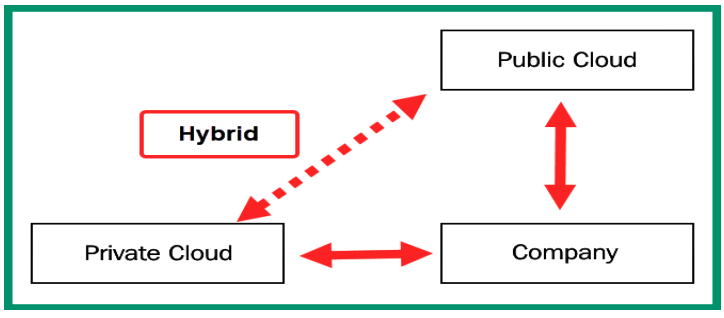
5. Gmail / Outlook Email Services

- Email services hosted on public cloud infrastructure.
- Users access it over the internet; nothing runs on local servers.

6. Netflix Streaming

- Netflix uses AWS public cloud to stream movies and shows to millions of users globally.

Hybrid Cloud Model



- A **Hybrid Cloud** is a combination of **private cloud** and **public cloud** infrastructures.

- It allows an organization to use **on-premises (private)** resources for sensitive workloads and at the same time take advantage of the **public cloud** for scalability, backup, or less critical applications.
- The two environments are integrated so that data and applications can move between them as needed.

Advantages

- **Best of both worlds** (security of private cloud + scalability of public cloud).
- **Improved reliability** with disaster recovery and backup options.
- **Cost-effective** (critical data stays private, while less critical workloads run on cheaper public cloud).
- **Business continuity** during failures or peak demand.

Limitations

- **High complexity** to integrate private and public environments.
- **Costly** compared to only private or only public cloud.
- Needs **skilled IT staff** for management and monitoring.

Example Use Cases

1. **Healthcare Sector**
    - A hospital stores **patient records** in its **private cloud** (for compliance and data privacy).
    - At the same time, it uses a **public cloud** to run data analytics on non-sensitive research data.
2. **E-Commerce Company**
    - The company runs its **payment system** on a **private cloud** for security.
    - During **festival sales**, it shifts additional website traffic to the **public cloud** to handle spikes in demand.

4. Community Cloud Model

- A **Community Cloud** is a cloud infrastructure that is **shared by several organizations** with **similar needs, interests, or requirements**.
- It can be **managed internally** by one of the organizations or **externally** by a third-party service provider.
- Resources, costs, and responsibilities are **shared among the participating organizations**.

Key Characteristics

1. **Shared Infrastructure** → Used by a group of organizations with common goals (e.g., security, compliance, policies).
2. **Cost Sharing** → Cheaper than private cloud since the cost is divided among multiple organizations.
3. **Controlled Access** → Only members of the community can access it (not open to the public).

4. **Customizable** → Tailored to the community's business or regulatory requirements.
5. **Security & Compliance** → Stronger security than public cloud but not as exclusive as private cloud.

Advantages

- **Cost-effective** compared to private cloud.
- **Collaboration** among organizations with shared interests.
- **Better security and compliance** than public cloud.

- **Resource pooling** helps optimize infrastructure use.

Limitations

- **Costlier** than public cloud (since it is not open to everyone).
- **Limited scalability** compared to large public cloud providers.
- **Complex governance** (multiple organizations sharing and managing).

Examples of Community Cloud

1. **Government Agencies**
   - Multiple government departments (e.g., health, education, taxation) share a **community cloud** to collaborate while keeping data secure and compliant with government regulations.
2. **Healthcare Sector**
   - Hospitals and research institutions share a **community cloud** to store and access medical records, research data, and collaborate on healthcare projects.
3. **Educational Institutions**
   - Several universities and colleges create a **community cloud** for sharing research projects, e-learning platforms, and digital libraries.

Cloud Service Models

Cloud service models define how services, applications, and resources are delivered from a cloud service provider's data center to end users.

The main service models are: **IaaS, PaaS, SaaS, DaaS, and IaC.**

1. Infrastructure-as-a-Service (IaaS)

- **Definition**: Provides the **basic building blocks** for cloud IT—networking, storage, and virtual machines.
- **Control Level**: Highest flexibility and control over resources (like in traditional IT infrastructure).
- **Management**: The user manages applications, data, and runtime, while the provider manages hardware.
- **Advantages**: Flexible, scalable, pay-as-you-go model.
- **Example**:

    - **Amazon EC2 (AWS)** → Users can launch virtual servers and install any OS or software.
    - **Microsoft Azure VM** → Create and manage Windows/Linux virtual machines.

Infrastructure-as-a-Service (IaaS)

IaaS is a cloud service model where cloud providers offer virtualized computing resources over the internet, such as:

- Virtual machines (VMs)
- Storage
- Networks
- Firewalls & load balancers
- Backup and disaster recovery

Users can rent these resources on a **pay-as-you-go** basis and manage their own applications and operating systems.

Examples of IaaS Providers

- **Amazon EC2** (AWS)
- **Microsoft Azure Virtual Machines**
- **Google Compute Engine (GCE)**
- **IBM Cloud**
- **DigitalOcean Droplets**

2. Platform-as-a-Service (PaaS)

- **Definition**: Provides a **platform** where developers can build, deploy, and manage applications **without managing the underlying infrastructure**.
- **Focus**: Application deployment and management only.
- **Advantages**: No need to handle OS updates, patching, or server maintenance.
- **Example**:
    - **Google App Engine** → Developers deploy apps without worrying about hardware.
    - **Heroku** → A platform for building and running applications in the cloud.

PaaS is a **cloud service model** that provides a **ready-to-use platform** with tools, runtime, and infrastructure so developers can build, test, and deploy applications **without worrying about managing servers, storage, or operating systems**.

Platform-as-a-Service (PaaS)

PaaS is a cloud service model that provides a complete platform for developers to build, run, and manage applications—without worrying about the underlying infrastructure.

It includes:

- Operating system
- Runtime environment
- Databases
- Development tools
- Middleware
- Web servers

Developers only focus on **writing and deploying code**, while the cloud provider manages everything else.

🔹 Example

- A developer wants to create a **chat application**.
- Instead of setting up servers, installing OS, databases, and security patches, they use **Google App Engine** or **Microsoft Azure App Service**.
- The cloud provider gives them:
    - A **runtime environment** (Java, Python, .NET, Node.js, etc.)
    - **Databases** (like Cloud SQL)
    - **Scalability** (automatically handles more users if traffic increases)
- The developer only uploads their app code → and it's deployed.

Software-as-a-Service (SaaS)

- **Definition**: Provides **ready-to-use software applications** over the internet.
- **Management**: The vendor manages everything (infrastructure, platform, updates, security).
- **User Task**: Just use the software.
- **Advantages**: No installation, no maintenance; accessible via web or mobile.
- **Example**:
    - **Gmail / Outlook.com** → Web-based email services.
    - **Google Workspace / Microsoft 365** → Online productivity apps (Docs, Sheets, Teams).
    - Netflix is a SaaS platform that provides movies and shows to users over the internet without requiring installation or maintenance.

    - **WhatsApp Web / WhatsApp Business Cloud**

✓ Users access messaging services through cloud-managed servers.
✓ No installation of backend systems.

Desktop-as-a-Service (DaaS)

DaaS is a cloud service model where virtual desktops are delivered to users over the internet. Instead of using a physical PC, users access a **cloud-hosted desktop** that contains:

- Operating system (Windows/Linux)
- Applications
- Storage
- User settings

All processing happens in the **cloud**, not on the local device.

Types:

  - **Persistent Desktop** → User's data and settings remain saved after logout.
  - **Non-Persistent Desktop** → Resets after logout, always starts fresh.
- **Management**: Provider manages backend (hardware, updates, backups). Security may be **shared responsibility**.
- **Advantages**: Access your desktop from anywhere; reduces hardware dependency.
- **Example**:
  - **Amazon WorkSpaces** → Virtual desktop service.
  - **VMware Horizon Cloud** → Cloud-hosted virtual desktops.

Benefits

- **Cost savings** (no need for powerful local PCs)
- **High security** (data stored in cloud, not on device)
- **Easy for IT to manage users and applications**
- **Scalable** (add/remove users quickly)
- Great for **work-from-home** or **remote work**

Infrastructure as Code (IaC)

- **Infrastructure as Code (IaC) means managing and setting up IT infrastructure (servers, networks, databases) using code instead of doing it manually.**
  You write configuration files or scripts, and tools automatically create the infrastructure for you.
- **Approach**: Instead of manual setup, configuration is automated with scripts.
- **Advantages**:
  - Faster deployment
  - Consistency across environments
  - Easier rollback with version control
- **Example**:
  - **Terraform** (by HashiCorp) → Automates cloud infrastructure setup.
  - **AWS CloudFormation** → Defines infrastructure in templates (code).

Example

Imagine you need **5 virtual servers** in AWS.

Traditional Way

- You log into AWS console
- Click "Create server"
- Select settings
- Repeat 5 times → slow and error-prone

Using IaC (e.g., Terraform)

You write a small code file and run the command.

The tool **automatically creates all 5 servers** with the exact same configuration.

Cloud Connectivity Solutions

When organizations use cloud services, it's important to make sure that **users connect securely** to applications, servers, and resources hosted in the cloud. Without secure access, attackers could intercept communication and steal sensitive data.

There are two common ways to connect securely: **VPN** and **Private Direct Connection**.

1. VPN (Virtual Private Network)

- **How it works**: Creates a **secure, encrypted tunnel** between your organization's network and the cloud provider's data center over the internet.
- **Advantages**:
  - Protects data-in-motion from hackers.
  - Low cost compared to private leased lines.
- **Limitation**: Organization must manage the VPN (setup, maintenance, user access).
- **Example**:
  - A company sets up a **VPN to AWS (Amazon Web Services)** so that employees working from home can securely access cloud-hosted applications.

2. Private Direct Connection

- **How it works**: Uses a **dedicated line** provided by an ISP to connect directly to the cloud provider's data center.
- **Advantages**:
  - More secure than using the internet.
  - Better performance (low latency, high speed).
- **Limitation**: More expensive than VPN.
- **Example**:
  - A financial firm uses **Azure ExpressRoute** (Microsoft's private connection) to link its office network directly to Microsoft's cloud data center for high-security transactions.