**CLOUD COMPUTING:**

**OUTLINE:**

 **Introduction to Cloud Computing**:

* Definition and characteristics of cloud computing
* Cloud service models (IaaS, PaaS, SaaS)
* Cloud deployment models (public, private, hybrid, community)

 **Microsoft Azure Fundamentals**:

* Overview of Microsoft Azure
* Azure portal and management tools
* Azure subscription management
* Resource groups and resource management

 **Azure Compute Services**:

* Virtual Machines (VMs)
  + VM creation, configuration, and management
  + VM scalability and availability
* Serverless Computing
  + Azure Functions
  + Azure Logic Apps
* Container Services
  + Azure Container Instances (ACI)
  + Azure Kubernetes Service (AKS)

 **Azure Storage Services**:

* Azure Blob Storage
* Azure Files
* Azure Disk Storage
* Azure Data Lake Storage

 **Azure Networking Services**:

* Virtual Networks (VNet)
* Network Security Groups (NSGs)
* Azure Load Balancer
* Azure DNS

 **Azure Database Services**:

* Azure SQL Database
* Azure Cosmos DB
* Azure Database for MySQL/PostgreSQL
* Azure Cache for Redis

 **Azure Identity and Access Management**:

* Azure Active Directory (Azure AD)
* Role-based Access Control (RBAC)
* Managed Identities

 **Azure Monitoring and Management**:

* Azure Monitor
* Azure Advisor
* Azure Resource Manager (ARM) templates

 **Azure Security and Compliance**:

* Azure Security Center
* Azure Sentinel
* Azure Key Vault
* Azure Policies

 **Azure DevOps and CI/CD**:

* Azure DevOps Services
* Azure Pipelines
* Azure Repos
* Azure Artifacts

 **Azure Pricing and Cost Management**:

* Azure pricing models
* Cost optimization strategies
* Azure Cost Management

**Web Server:**

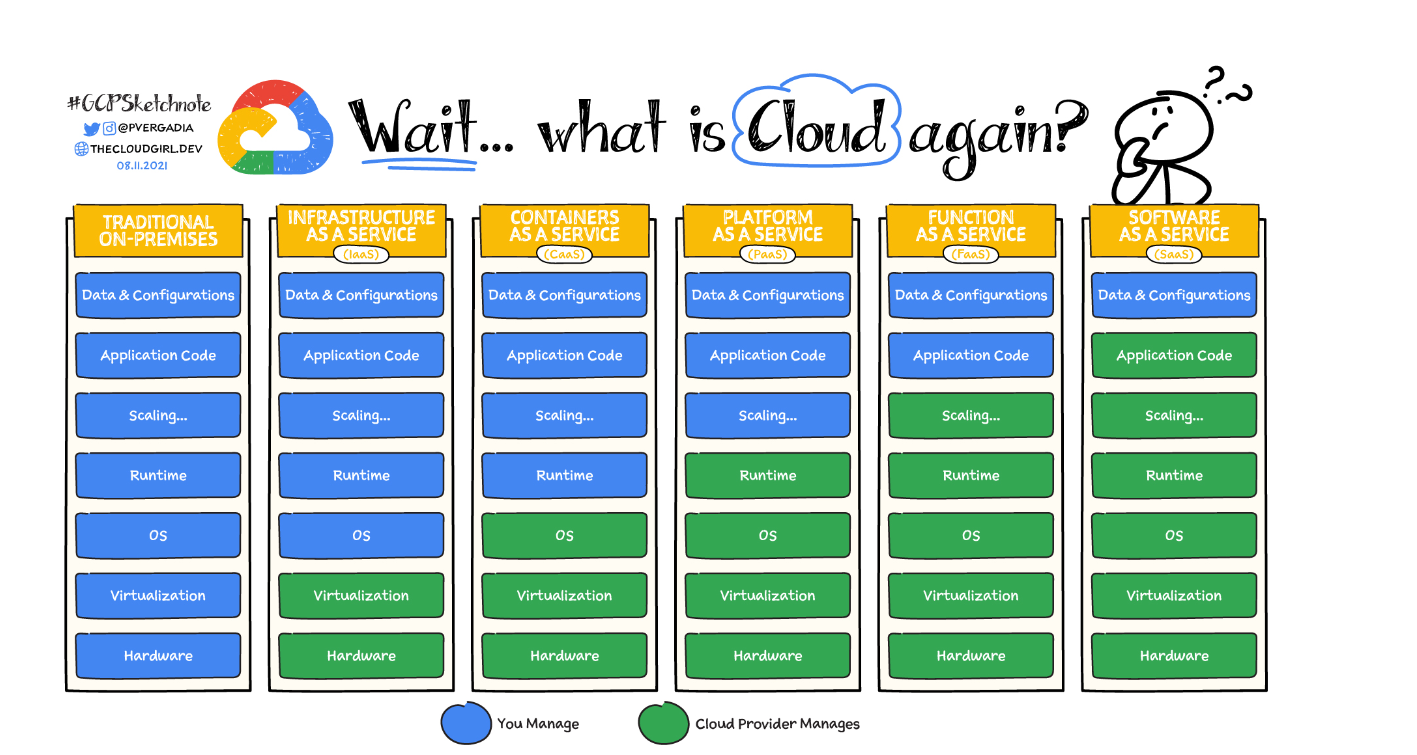
A computer that hosts a website on the Internet.

* On the hardware side, a web server is a computer that stores web server software and a website's component files (for example, HTML documents, images, CSS stylesheets, and JavaScript files). A web server connects to the Internet and supports physical data interchange with other devices connected to the web.
* On the software side, a web server includes several parts that control how web users access hosted files. At a minimum, this is an HTTP server. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.

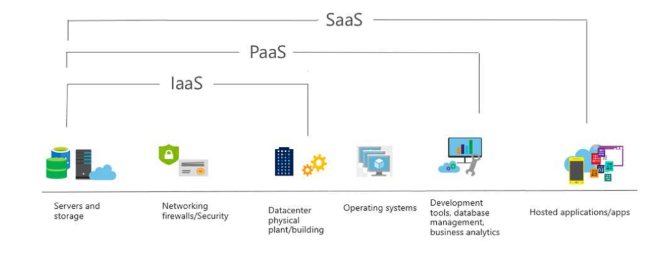
At the most basic level, whenever a browser needs a file that is hosted on a web server, the browser requests the file via HTTP. When the request reaches the correct (hardware) web server, the (software) HTTP server accepts the request, finds the requested document, and sends it back to the browser, also through HTTP. (If the server doesn't find the requested document, it returns a 404 response instead.)

* A static web server, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files as-is to your browser.
* A dynamic web server consists of a static web server plus extra software, most commonly an application server and a database. We call it "dynamic" because the application server updates the hosted files before sending content to your browser via the HTTP server.

For example, to produce the final webpages you see in the browser, the application server might fill an HTML template with content from a database. Sites like MDN or Wikipedia have thousands of webpages. Typically, these kinds of sites are composed of only a few HTML templates and a giant database, rather than thousands of static HTML documents. This setup makes it easier to maintain and deliver the content.



* **On-premises:** If you decide to build your house from scratch, you do everything yourself. You’ll need to source the raw materials and tools, put everything together, and run to the store every time you need anything. This is similar to running an application on-premises, where you own everything from the hardware to your applications and scaling.
* **Infrastructure as a service:**If you are busy, you might consider hiring a contractor to do the work. You tell them how you want the house to look and how many rooms you want, and they take the instructions and build your home. IaaS works in a similar way for your applications. You rent the hardware to run your application on, but you are responsible for managing the OS, runtime, scale, and all the data. **Example:**[Compute Engine](https://cloud.google.com/compute)
* **Containers as a service:**If buying a home is just too much work due to the maintenance it comes with, you can choose to rent instead. The basic utilities are included, but you bring your own furniture and make the space yours. With containers, you can bring a containerized application, so you don't have to worry about the underlying operating system but still have control over scale and runtime. **Example:** [Google Kubernetes Engine (GKE)](https://cloud.google.com/kubernetes-engine)
* **Platform as a service:**If you don’t want to worry about furnishing your living space, you can rent a furnished house. PaaS lets you bring your own code and deploy it but leaves the server management and scaling up to the cloud provider. **Examples:** [App Engine](https://cloud.google.com/appengine), [Cloud Run](https://cloud.google.com/run)
* **Function as a service:**If you just need a small dedicated place to work away from your home, you can rent a desk in a coworking workspace. Similarly, FaaS allows you to build and deploy a small piece of code, or a function, that performs a specific task. The cloud provider adds scale if needed when a function executes. **Example:** [Cloud Functions](https://cloud.google.com/functions)
* **Software as a service:**Now, imagine you move into a finished house (rented or purchased), but you have to pay for upkeep, such as cleaning or lawn care. SaaS is the same—you pay to use a complete application for a specific purpose that is managed, maintained, and secured by the cloud provider, but you are responsible for taking care of your own data. **Example:** [Google Workspace](https://workspace.google.com/)



**USE CASES:**

**Infrastructure as a Service (IaaS)**

* Virtual Data Centers: Organizations can use IaaS to create virtual data centers for hosting applications and services.
* Disaster Recovery: IaaS can be used to set up backup and recovery systems to ensure business continuity.
* Scalable Web Applications: IaaS allows for the scaling of web applications based on demand, providing flexibility and cost savings.
* Big Data Analysis: IaaS can provide the computational power and storage needed for big data processing and analysis.
* Testing and Development: Developers can use IaaS to quickly set up and tear down environments for testing and development purposes.

**Platform as a Service (PaaS)**

* Application Development: PaaS provides a platform for developers to build, test, and deploy applications quickly.
* API Development and Management: PaaS can be used to develop and manage APIs, enabling integration with other applications and services.
* DevOps: PaaS supports DevOps practices by providing tools for continuous integration and continuous deployment (CI/CD).
* Microservices Architecture: PaaS is ideal for developing and managing microservices, allowing for scalable and modular application design.
* Database Management: PaaS can be used to manage databases, providing scalability, security, and backup services.

**Software as a Service (SaaS)**

* Customer Relationship Management (CRM): SaaS solutions like Salesforce are commonly used for managing customer relationships.
* Enterprise Resource Planning (ERP): SaaS ERP systems like SAP or Oracle Cloud help manage business processes and resources.
* Collaboration Tools: SaaS applications like Microsoft 365 and Google Workspace are used for collaboration and productivity.
* Human Resource Management (HRM): SaaS solutions like Workday and BambooHR are used for managing HR functions.
* E-commerce Platforms: SaaS platforms like Shopify and BigCommerce are used to set up and manage online stores.
* **Office Productivity**: Using tools like email, document editing, and collaboration without managing software installations.

**DIFFERENT DEPLOYMENT MODELS:**

 **Public Cloud**:

* The public cloud is owned and operated by a third-party cloud service provider, such as Microsoft, Amazon, or Google.
* Resources (e.g., storage, computing power, and applications) are shared among multiple tenants (organizations or individuals) over the internet.
* Public clouds offer high scalability, elasticity, and cost-effectiveness, as the infrastructure is managed by the cloud provider.
* Examples include Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP).

 **Private Cloud**:

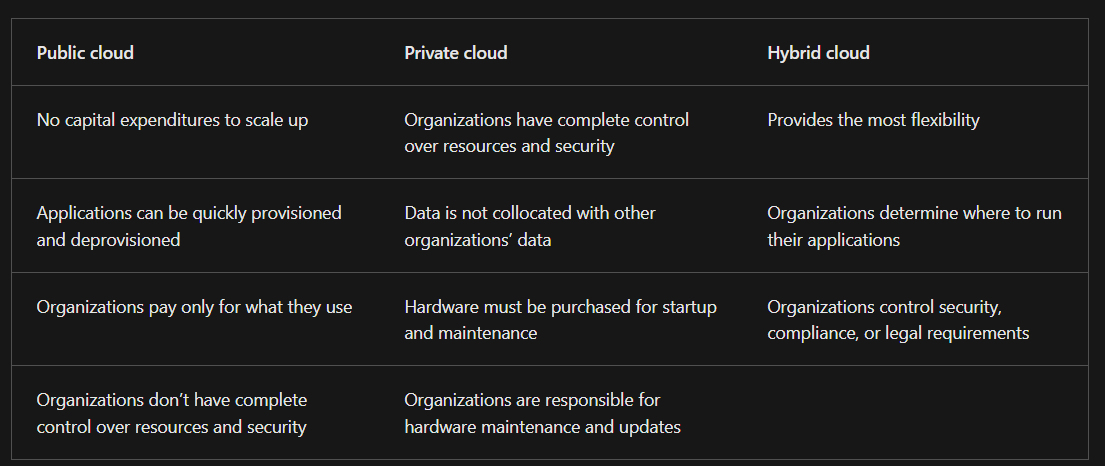
* The private cloud is owned and operated by a single organization, either on-premises or hosted by a third-party provider.
* Resources are dedicated to and controlled by the organization, providing a higher degree of security and customization.
* Private clouds are often used by organizations with strict compliance requirements or those that need more control over their infrastructure.
* Examples include on-premises private clouds and hosted private clouds.

 **Hybrid Cloud**:

* The hybrid cloud is a combination of public and private cloud environments.
* Organizations can leverage the benefits of both public and private clouds, using the public cloud for scalable, on-demand resources and the private cloud for sensitive or mission-critical workloads.
* Hybrid clouds enable organizations to maintain control over critical applications and data while still benefiting from the flexibility and cost-effectiveness of public cloud services.
* Hybrid cloud deployment often involves integrating on-premises infrastructure with public cloud services.

 **Community Cloud**:

* The community cloud is a cloud infrastructure shared by several organizations that have shared concerns (e.g., security, compliance, or mission).
* It may be managed by the organizations themselves or by a third-party provider.
* Community clouds are useful for organizations within the same industry or with similar regulatory requirements, allowing them to share resources and costs.
* Examples include government agencies or healthcare organizations that collaborate on a community cloud.



**Consumption-based model**

• Cloud service providers operate on a consumption-based model, which means that end users only pay for the resources that they use. Whatever they use is what they pay for.

• Better cost prediction

• Prices for individual resources and services are provided

• Billing is based on actual usage

When comparing IT infrastructure models, there are two types of expenses to consider. **Capital expenditure (CapEx) and operational expenditure (OpEx).**

CapEx is typically a one-time, up-front expenditure to purchase or secure tangible resources. A new building, repaving the parking lot, building a datacenter, or buying a company vehicle are examples of CapEx.

In contrast, OpEx is spending money on services or products over time. Renting a convention center, leasing a company vehicle, or signing up for cloud services are all examples of OpEx.

Cloud computing falls under OpEx because cloud computing operates on a consumption-based model. With cloud computing, you don’t pay for the physical infrastructure, the electricity, the security, or anything else associated with maintaining a datacenter. Instead, you pay for the IT resources you use. If you don’t use any IT resources this month, you don’t pay for any IT resources.

This consumption-based model has many benefits, including:

* No upfront costs.
* No need to purchase and manage costly infrastructure that users might not use to its fullest potential.
* The ability to pay for more resources when they're needed.
* The ability to stop paying for resources that are no longer needed.

**Benefits of cloud:**

* **High availability:** When you’re deploying an application, a service, or any IT resources, it’s important the resources are available when needed. High availability focuses on ensuring maximum availability, regardless of disruptions or events that may occur.

When you’re architecting your solution, you’ll need to account for service availability guarantees. Azure is a highly available cloud environment with uptime guarantees depending on the service. These guarantees are part of the service-level agreements (SLAs).

* **Scalability:** Another major benefit of cloud computing is the scalability of cloud resources. Scalability refers to the ability to adjust resources to meet demand. If you suddenly experience peak traffic and your systems are overwhelmed, the ability to scale means you can add more resources to better handle the increased demand.

The other benefit of scalability is that you aren't overpaying for services. Because the cloud is a consumption-based model, you only pay for what you use. If demand drops off, you can reduce your resources and thereby reduce your costs.

Scaling generally comes in two varieties: vertical and horizontal. Vertical scaling is focused on increasing or decreasing the capabilities of resources. Horizontal scaling is adding or subtracting the number of resources.

**Vertical scaling**

With vertical scaling, if you were developing an app and you needed more processing power, you could vertically scale up to add more CPUs or RAM to the virtual machine. Conversely, if you realized you had over-specified the needs, you could vertically scale down by lowering the CPU or RAM specifications.

**Horizontal scaling**

With horizontal scaling, if you suddenly experienced a steep jump in demand, your deployed resources could be scaled out (either automatically or manually). For example, you could add additional virtual machines or containers, scaling out. In the same manner, if there was a significant drop in demand, deployed resources could be scaled in (either automatically or manually), scaling in.

* **Reliability:** The ability of a system to recover from failures and continue to function. It's also one of the pillars of the Microsoft Azure Well-Architected Framework.

The cloud, by virtue of its decentralized design, naturally supports a reliable and resilient infrastructure. With a decentralized design, the cloud enables you to have resources deployed in regions around the world. With this global scale, even if one region has a catastrophic event other regions are still up and running. You can design your applications to automatically take advantage of this increased reliability. In some cases, your cloud environment itself will automatically shift to a different region for you, with no action needed on your part. You’ll learn more about how Azure leverages global scale to provide reliability later in this series.

* **Predictability**: In the cloud lets you move forward with confidence. Predictability can be focused on performance predictability or cost predictability. Both performance and cost predictability are heavily influenced by the Microsoft Azure Well-Architected Framework. Deploy a solution built around this framework and you have a solution whose cost and performance are predictable

1. Performance predictability focuses on predicting the resources needed to deliver a positive experience for your customers. Autoscaling, load balancing, and high availability are just some of the cloud concepts that support performance predictability. If you suddenly need more resources, autoscaling can deploy additional resources to meet the demand, and then scale back when the demand drops. Or if the traffic is heavily focused on one area, load balancing will help redirect some of the overload to less stressed areas.
2. Cost predictability is focused on predicting or forecasting the cost of the cloud spend. With the cloud, you can track your resource use in real time, monitor resources to ensure that you’re using them in the most efficient way, and apply data analytics to find patterns and trends that help better plan resource deployments. By operating in the cloud and using cloud analytics and information, you can predict future costs and adjust your resources as needed. You can even use tools like the Total Cost of Ownership (TCO) or Pricing Calculator to get an estimate of potential cloud spend.

* **Security:** You can find a cloud solution that matches your security needs. If you want maximum control of security, infrastructure as a service provides you with physical resources but lets you manage the operating systems and installed software, including patches and maintenance. If you want patches and maintenance taken care of automatically, platform as a service or software as a service deployment may be the best cloud strategies for you. And because the cloud is intended as an over-the-internet delivery of IT resources, cloud providers are typically well suited to handle things like distributed denial of service (DDoS) attacks, making your network more robust and secure.
* **Governance:** By establishing a good governance footprint early, you can keep your cloud footprint updated, secure, and well managed. Whether you’re deploying infrastructure as a service or software as a service, cloud features support governance and compliance. Things like set templates help ensure that all your deployed resources meet corporate standards and government regulatory requirements. Plus, you can update all your deployed resources to new standards as standards change. Cloud-based auditing helps flag any resource that’s out of compliance with your corporate standards and provides mitigation strategies. Depending on your operating model, software patches and updates may also automatically be applied, which helps with both governance and security.
* Manageability: Management of the cloud speaks to managing your cloud resources. In the cloud, you can:
* Automatically scale resource deployment based on need.
* Deploy resources based on a preconfigured template, removing the need for manual configuration.
* Monitor the health of resources and automatically replace failing resources.
* Receive automatic alerts based on configured metrics, so you’re aware of performance in real time.

Management in the cloud

Management in the cloud speaks to how you’re able to manage your cloud environment and resources. You can manage these:

* Through a web portal.
* Using a command line interface.
* Using APIs.
* Using PowerShell.

**Clustering in cloud computing** refers to the technique of grouping multiple servers or resources together to work as a single system, providing benefits like improved performance, redundancy, high availability, and scalability. In a cloud environment, clustering allows multiple instances of servers or virtual machines to work together to handle tasks more efficiently, manage workloads, and ensure that if one server fails, another can take over without interrupting services.

**What is a failover cluster?**

In computing, a failover cluster refers to a group of independent servers that work together to maintain high availability of applications and services. If one of the servers fails, another node in the cluster can take over its workload with little or no downtime. This process is known as failover.

In a failover cluster, the clustered servers can collectively improve the availability and scalability of applications and services. The servers work together to provide either continuous availability or at least high availability (HA) through the failover process. Failover clusters can comprise either physical servers or virtual machines (VMs).

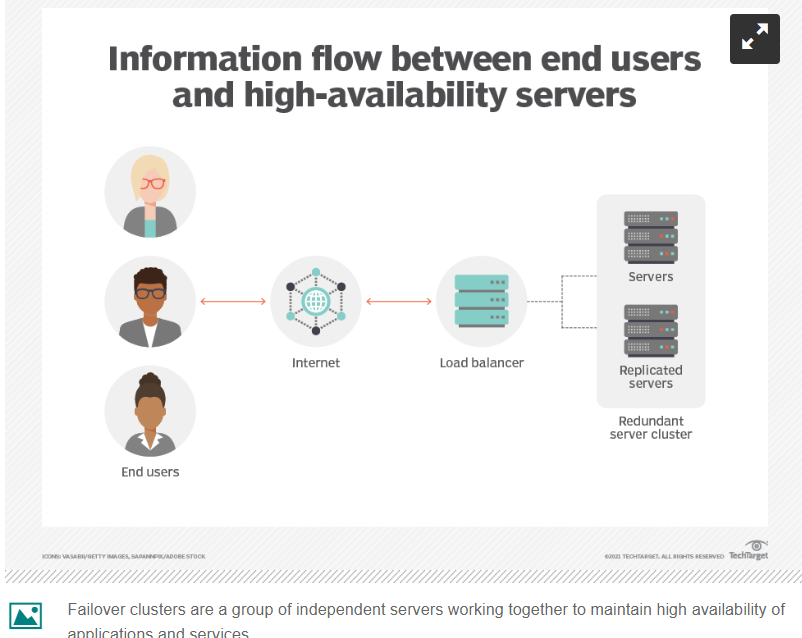
Each server is called a node and is connected to other servers in the cluster by physical cables and software. A failover cluster consists of at least two nodes to transfer data, as well as software to process the data through the cables. In addition, a cluster uses one or more technologies for load balancing, storage, parallel processing and other functions.

The applications and services in a failover cluster are sometimes known as clustered roles. The cluster keeps these roles operational if one server fails. At the same time, each role is proactively monitored to ensure that it is working properly. If it's not, it might be restarted or the role moved to another node.

**The need for failover clusters:**

Server failure can lead to application downtime, which can result in operational disruptions for users. By providing continuous availability or high availability, failover clusters enable users to keep using the applications and services they need without experiencing outages, even if a server fails. There might be a brief service interruption with high-availability clusters during the failover process. However, the system usually recovers quickly with little or no data loss and minimum downtime.

Failover clusters play a vital role to ensure the ongoing availability of mission-critical applications and systems such as online transaction processing (OLTP) systems that demand very high -- near 100% -- availability. Database replication and disaster recovery (DR) also require failover clusters. These clusters provide geographic replication so if a server in one location goes down, data will still be available on failover servers at other sites.

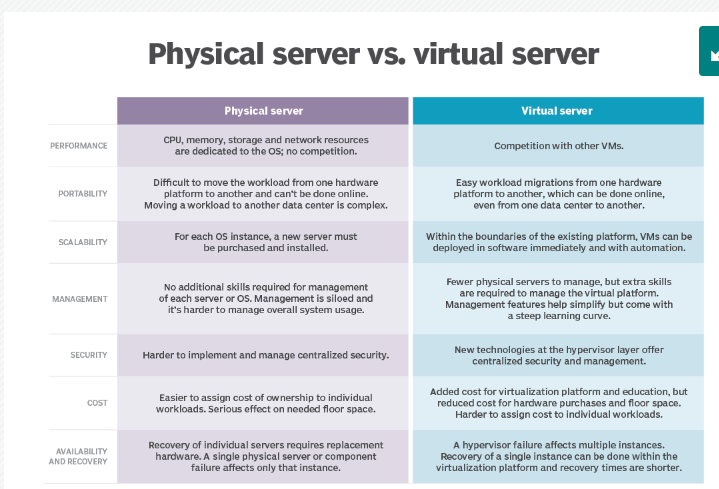


**How failover clusters work:**

**In a high-availability failover cluster**, groups of independent servers share data and resources, including storage. At any time, a least one node is active and at least one is passive. These clusters include a monitoring connection that allows each server to check the health of the other servers. In a two-node cluster (the simplest possible configuration), if a node fails the other node will recognize the failure via the monitoring connection and configure itself as the active node. Larger configurations usually use dedicated servers that determine if any nodes are failing and then direct another node to assume the load and participate in the failover process.

In high-availability failover clusters using VMs, the VMs are pooled into a cluster along with the physical servers they reside on. If there is a failure, the VMs on the failed host will be restarted on alternate hosts.

**In a continuous-availability failover** cluster -- also known as a fault-tolerant cluster -- multiple systems share one copy of a computer's operating system (OS). Thus, the commands issued by one system are simultaneously executed on the other systems as well. The cluster requires a continuously available and near-exact copy of a machine running the service (physical or virtual). This redundancy model is known as 2N. It can automatically detect failures of hard drives, power supplies, network components and CPUs. When the cluster identifies a failure point, a backup component -- or procedure -- takes its place instantaneously without service interruption.



**High-availability failover clusters vs. continuous-availability failover clusters**

High-availability clusters attempt 99.999% availability (five 9s availability). These clusters are generally adequate for most applications and services. However, services or applications that require 100% availability, continuous-availability failover clusters are required. These include mission-critical applications such as electronic stock trading, ATM banking, order management, employee time clock systems and reservation systems. Many applications in manufacturing, logistics and e-commerce also require continuous-availability failover clusters

**Types of failover clusters**

Failover clustering is a popular feature in Windows Server and [Azure Stack](https://www.techtarget.com/searchwindowsserver/definition/Microsoft-Azure-Stack) HCI. With those OSes, organizations can create highly available or continuously available file share storage for applications such as [Microsoft SQL Server](https://www.techtarget.com/searchdatamanagement/definition/SQL-Server) and Hyper-V VMs. Another approach is to create highly available clustered roles on physical servers or VMs that are installed on servers running Hyper-V.

**What is virtualization?**

Virtualization is a process that allows for more efficient use of physical computer hardware and is the foundation of cloud computing.

Virtualization uses software to create an abstraction layer over computer hardware, enabling the division of a single computer's hardware components—such as processors, memory and storage—into multiple virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware.

Today, virtualization is a standard practice in enterprise IT architecture. It is also the technology that drives cloud computing economics. Virtualization enables cloud providers to serve users with their existing physical computer hardware. It enables cloud users to purchase only the computing resources they need when they need it, and to scale those resources cost-effectively as their workloads grow.

**Benefits of virtualization:**

Virtualization brings several benefits to data center operators and service providers:

**Resource efficiency:**

Before virtualization, IT staff would allocate a dedicated physical CPU to each application server, buying and setting up a separate server for every application. This approach, favoring one application and one operating system per computer, was adopted for its reliability. Invariably, each physical server would be underused. In contrast, server virtualization enables you to run several applications—each on its own VM with its own OS—on a single physical computer (typically an x86 server) without sacrificing reliability. This enables maximum use of the physical hardware’s computing capacity.

**Easier management:**

Replacing physical computers with software-defined VMs makes it easier to use and manage policies written in software. This allows you to create automated IT service management workflows. For example, automated deployment and configuration tools enable administrators to define collections of virtual machines and applications as services, in software templates. This means that they can install those services repeatedly and consistently without cumbersome, time-consuming and error-prone manual setup. Admins can use virtualization security policies to mandate certain security configurations based on the role of the virtual machine. Policies can even increase resource efficiency by retiring unused virtual machines to save on space and computing power.

**Minimal downtime:**

OS and application crashes can cause downtime and disrupt user productivity. Admins can run multiple redundant virtual machines alongside each other and failover between them when problems arise. Running multiple redundant physical servers is more expensive.

**Faster provisioning:**

Buying, installing and configuring hardware for each application is time-consuming. If the hardware is already in place, provisioning virtual machines to run all your applications is significantly faster. You can even automate it using management software and build it into existing workflows.

**Solutions:**

Several companies offer virtualization solutions covering specific data center tasks or end user-focused, desktop virtualization scenarios. Better-known examples include VMware, which specializes in server, desktop, network and storage virtualization; Citrix, which has a niche in application virtualization but also offers server virtualization and virtual desktop solutions; and Microsoft, whose Hyper-V virtualization solution ships with Windows and focuses on virtual versions of server and desktop computers.

**Virtual machines:**

Virtual machines are virtual environments that simulate a physical computer in software form. They normally comprise several files containing the VM’s configuration, the storage for the virtual hard drive, and some snapshots of the VM that preserve its state at a particular point in time.

**Hypervisors**

A hypervisor is the software layer that coordinates VMs. It serves as an interface between the VM and the underlying physical hardware, ensuring that each has access to the physical resources it needs to execute. It also ensures that the VMs don’t interfere with each other by impinging on each other’s memory space or compute cycles.

There are two types of hypervisors:

**Type 1 hypervisors**

Type 1 or “bare-metal” hypervisors interact with the underlying physical resources, replacing the traditional operating system altogether. They most commonly appear in virtual server scenarios.

**Type 2 hypervisors**

Type 2 hypervisors run as an application on an existing OS. Most commonly used on endpoint devices to run alternative operating systems, they carry a performance overhead because they must use the host OS to access and coordinate the underlying hardware resources.

**Types of virtualizations:**

To this point we’ve discussed server virtualization, but many other IT infrastructure elements can be virtualized to deliver significant advantages to IT managers in particular and the enterprise as a whole. In this section, we cover the following types of virtualization:

* Desktop virtualization
* Network virtualization
* Storage virtualization
* Data virtualization
* Application virtualization
* Data center virtualization
* CPU virtualization
* GPU virtualization
* Linux virtualization
* Cloud virtualization

**Virtualization versus containerization**

Server virtualization reproduces an entire computer in hardware, which then runs an entire OS. The OS runs one application. That’s more efficient than no virtualization at all, but it still duplicates unnecessary code and services for each application you want to run.

Containers take an alternative approach. They share an underlying OS kernel, only running the application and the things it depends on, like software libraries and environment variables. This makes containers smaller and faster to deploy.

A **jump server** (also known as a jump box, jump host, or bastion host) is a special-purpose server that sits between a public network (like the internet) and a private, more secure network. It acts as a gateway, allowing users to securely connect to and access servers and resources within the private network.

**Key Features and Purpose of a Jump Server:**

* Security: It serves as a controlled entry point to the private network, reducing the attack surface by limiting direct access to the internal network.
* Access Control: Only authorized users can connect to the jump server, which then provides access to other servers within the internal network.
* Monitoring and Auditing: Activity on the jump server can be monitored and logged, providing a clear audit trail of who accessed what resources and when.
* Segregation: By isolating the internal network from direct public access, it reduces the risk of attacks on the more critical or sensitive systems within the private network.

**Common Use Cases:**

* SSH Access: In environments where SSH is used to manage servers, the jump server acts as the only SSH entry point to the internal network.
* RDP Access: For Windows environments, a jump server might provide Remote Desktop Protocol (RDP) access to internal servers.
* Administrative Access: System administrators use the jump server to manage internal resources securely.

**FUNDAMENTALS OF MICROSOFT AZURE:**

 **Overview of Microsoft Azure**:

* Microsoft Azure is a comprehensive cloud computing platform offered by Microsoft.
* Azure provides a wide range of services, including computing, storage, networking, database, analytics, and more.
* It supports a variety of operating systems, programming languages, frameworks, and tools, making it a versatile cloud platform.
* Azure follows the cloud computing service models (IaaS, PaaS, SaaS) and can be deployed in public, private, or hybrid cloud environments.

 **Azure Portal and Management Tools**:

* The Azure Portal is the web-based graphical user interface (GUI) for managing Azure resources and services.
* The portal allows users to create, configure, and monitor Azure resources, as well as access various management tools and services.
* In addition to the Azure Portal, Azure also provides other management tools, such as:
  + Azure PowerShell: A command-line interface (CLI) for managing Azure resources using PowerShell scripts.
  + Azure CLI: A cross-platform CLI tool for managing Azure resources.
  + Azure Resource Manager (ARM): A deployment and management service that provides a consistent management layer for Azure resources.

 **Azure Subscription Management**:

* An Azure subscription is a logical unit of Azure services that is linked to an Azure Active Directory tenant and an Azure account.
* Subscriptions provide a way to organize and manage access, policies, and billing for the Azure resources used within the subscription.
* Subscription management includes tasks such as creating new subscriptions, managing billing and costs, and assigning roles and permissions to users.

 **Resource Groups and Resource Management**:

* Resource groups are containers that hold related Azure resources, such as virtual machines, storage accounts, and databases.
* Resource groups provide a way to manage and organize Azure resources based on factors like deployment, management, and billing.
* Resource management involves tasks like creating, updating, and deleting resources within a resource group, as well as setting policies and permissions for those resources.
* Azure Resource Manager (ARM) templates are used to define and deploy resources in a declarative way, making it easier to manage and automate resource deployments.