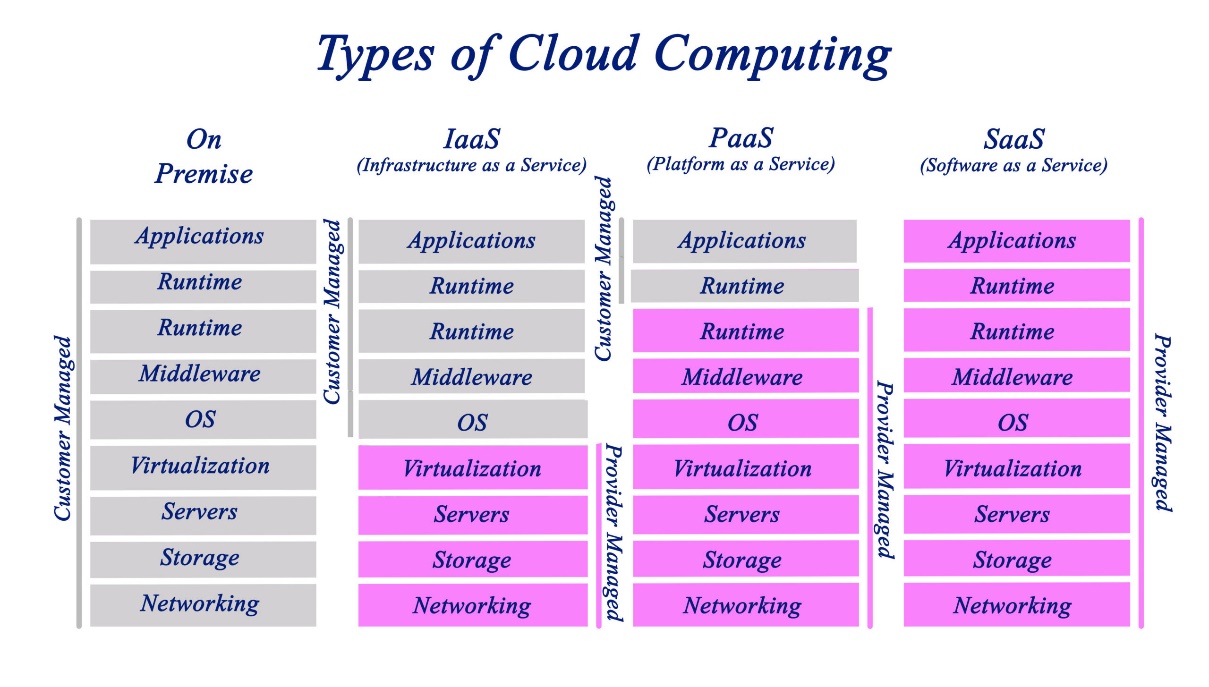
**Microsoft Certified: Azure Fundamentals**

* **What is cloud computing**
* Cloud computing is the on-demand delivery of computing resources over the internet.
* This includes services like data storage, servers, databases, networking, software, and more.
* [Instead of owning and maintaining physical data centers and servers, companies can rent access to these resources from a cloud service provider (CSP) on a pay-as-you-go basis](https://www.ibm.com/topics/cloud-computing)
* Here are some key benefits of cloud computing:
* Cost-Effectiveness: Reduces the need for large capital expenditures on hardware and software. Pay for what you consume
* Scalability: Easily scale resources up or down based on demand.
* Flexibility: Access resources from anywhere with an internet connection.
* Speed: Quickly deploy and manage applications and services.
* [Security: Many CSPs offer robust security measures to protect data](https://www.ibm.com/topics/cloud-computing)
* Global: Launch workloads anywhere in the world, just choose a region.
* Reliable: Data backup, Disaster recovery, fault tolerance
* **Types of cloud computing services**
* **Infrastructure as a service (IaaS)**
* Provides virtualized computing resources over the internet, such as virtual machines, storage, and networks.
* It simply provides the underlying operating systems, security, networking, and servers for developing such applications, and services, and deploying development tools, databases, etc.
* Example: Azure, AWS, Oracle Cloud.
* Amazon Web Services (AWS): Netflix uses AWS to host its streaming service, allowing it to scale resources up or down based on viewer demand.
* Example: Azure Virtual Machine
* **Platform as a Service:**
* Offers a platform allowing customers to develop, run, and manage applications without dealing with the underlying infrastructure.
* PaaS services are hosted in the cloud and accessed by users simply via their web browser.
* A PaaS provider hosts the hardware and software on its own infrastructure. As a result, PaaS frees users from having to install in-house hardware and software to develop or run a new application
* Example: Google App Engine. Snapchat uses Google App Engine to manage its backend services, enabling it to focus on developing new features rather than managing servers.
* Example: Azure SQL Database
* **Software as a Service:**
* Delivers software applications over the internet on a subscription basis.
* Instead of installing and maintaining software, we simply access it via the Internet, freeing ourselves from the complex software and hardware management.
* It removes the need to install and run applications on our own computers or in the data centres eliminating the expenses of hardware as well as software maintenance.
* Example: Salesforce, Gmail, Microsoft Office365
* Example: Salesforce. Coca-Cola uses Salesforce for customer relationship management (CRM) to streamline their sales processes and improve customer interactions.
* Example: Microsoft 365



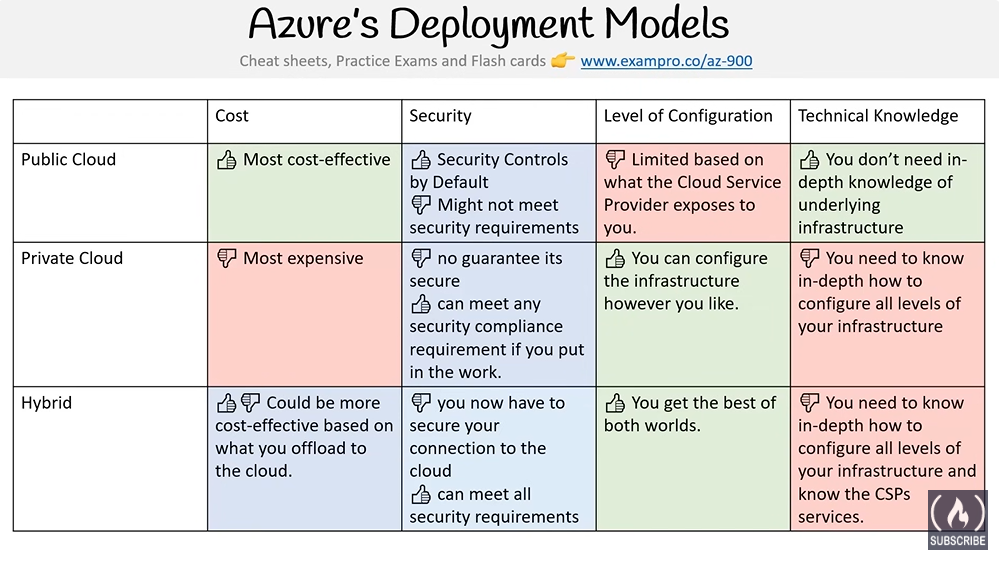
A close-up of a diagram

Description automatically generated

* **Deployment Models**

Cloud deployment models describe how cloud services are delivered and accessed. Here are the main types:

* **Public Cloud**:
* Services are delivered over the internet and shared across multiple organizations. The infrastructure is owned and managed by a third-party cloud provider.
* **Example**: **Google Cloud Platform (GCP)**. [Companies like Spotify use GCP to host their music streaming services, benefiting from scalability and cost-efficiency](https://www.bing.com/aclick?ld=e8W6jsaWdYf9ahppsU7LkhgjVUCUzkOfVZkblAJX9Dooj2Y005SAIqlxBdQg9MHr-GcKZlYGdcpUUlLEuhlLGr1VqN4uZTRsEKiSpA6e6YxLqbgjN5nIkhGVRfzLSN9BOLZ6eH1baLrBDwIAJ6Iih6YI2i4QMn-K3Vt7jgJ5MgeFKDsLM5&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa183M2M5NzlkY2Q4ZjMxNTBkZGM0MTFmNmQ3ZjhjZTVlZl9rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfNzNjOTc5ZGNkOGYzMTUwZGRjNDExZjZkN2Y4Y2U1ZWZfa18lMjZtc2Nsa2lkJTNkNzNjOTc5ZGNkOGYzMTUwZGRjNDExZjZkN2Y4Y2U1ZWY&rlid=73c979dcd8f3150ddc411f6d7f8ce5ef)[1](https://www.cloudwards.net/cloud-deployment-models/).
* **Private Cloud**:
* **Description**: Services are maintained on a private network and used exclusively by a single organization. This model offers greater control and security.
* **Example**: **IBM Cloud Private**. [Financial institutions often use private clouds to ensure compliance with regulatory requirements and to protect sensitive data2](https://www.geeksforgeeks.org/cloud-deployment-models/).
* **Hybrid Cloud**:
* **Description**: Combines public and private clouds, allowing data and applications to be shared between them. This model offers flexibility and optimized workload management.
* **Example**: **Microsoft Azure**. [Businesses use Azure’s hybrid capabilities to run applications in a combination of on-premises, private, and public cloud environment](https://cloud.google.com/discover/types-of-cloud-computing). Vaas, data sovereignty, hypervisor, virtualization concepts, data replication, cost orchestration, serverless computing, multi-tenancy, basic azure services, cloud architect.



* **Virtualization:** Virtualization is the process of creating a virtual version of something, such as an operating system, server, storage, or network resources.
* **CAPEX vs OPEX**
* **Capital Expenditure (CapEx):**
* **Description**: CapEx involves upfront investments in physical infrastructure. These are long-term investments.
* This includes server costs, storage costs, network costs, backup and archive costs, disater recovery costs, datacenter costs, technical people costs
* **With capital expenses you have to guess upfront what you plan to spend**
* **Operational Expenditure (OpEx):**
* **Description**: OpEx involves ongoing costs for services and products. In cloud computing, this is typically a pay-as-you-go model.
* **OpEx** refers to the ongoing costs for running a business. These are regular expenses that are necessary for day-to-day operations. **Examples**: Subscription fees for cloud services, such as AWS, Azure, or Google Cloud.
* **With operational expenses you can try a product or service without investing in equipment**

**Payment Structure:**

**CapEx:** Large, upfront payments.

**OpEx:** Smaller, recurring payments.

* **Cloud Computing features**
* **High Availability:** In cloud computing, High Availability (HA) ensures that a system or application is operational and accessible for a high percentage of time, typically 99.9% or higher. Load balancers and auto-scaling groups are commonly used to maintain high availability.
* **High Scalability: High scalability** in the cloud allows you to easily increase or decrease resources based on demand. Cloud platforms provide services like auto-scaling, which automatically adjusts the number of running instances based on current traffic. This ensures that your application can handle increased loads without manual intervention.
* **High Elasticity:** Elasticity in cloud computing refers to the ability to dynamically scale resources up or down based on demand.For example, during peak traffic times, additional virtual machines can be spun up to handle the load, and they can be terminated when the traffic decreases. This flexibility helps optimize costs and performance.
* **Fault Tolerance:** Fault Tolerance is the ability of a system to continue operating without interruption in the presence of hardware or software failures. Cloud providers offer fault tolerance through redundancy and failover mechanisms.
* **High Durability: High durability** in the cloud is achieved through data replication and robust storage solutions. Cloud storage services like Amazon S3, Google Cloud Storage, and Azure Blob Storage replicate data across multiple locations to ensure that it remains intact and accessible even in the event of hardware failures.
* **Disaster Recovery**: Disaster Recovery involves the planning and processes for restoring and recovering data and systems after a natural or human-induced disaster
* **Microsoft Azure**

**Microsoft Azure** is a comprehensive cloud computing platform developed by Microsoft. It offers a wide range of services and products designed to help businesses and developers build, deploy, and manage applications through Microsoft’s global network of data centers

**Key Services:** Microsoft Azure is a comprehensive cloud computing platform that offers a wide range of services to help businesses and developers build, deploy, and manage applications through Microsoft’s global network of data centers. Here’s a detailed overview of some key Azure services:

**1. Compute Services**

* **Azure Virtual Machines (VMs):** Provides scalable, on-demand computing resources for various workloads. Think of these as computers in the cloud that you can use whenever you need extra computing power.
* **Azure Kubernetes Service (AKS):** Simplifies the deployment, management, and operations of Kubernetes. This helps you manage and run lots of containers (small, fast applications) easily.
* **Azure App Services:** Enables you to build and host web apps, mobile backends, and RESTful APIs in the programming language of your choice.  Allows you to create and host websites and apps without worrying about the underlying hardware.

**2. Storage Services**

* **Azure Blob Storage:** Object storage solution for the cloud, optimized for storing massive amounts of unstructured data. A place to store large amounts of unstructured data like videos, images, and backups.
* **Azure File Storage:** Fully managed file shares in the cloud that are accessible via the SMB protocol. Cloud-based file storage that works like a network drive.
* **Azure Disk Storage:** High-performance, durable block storage for Azure VMs. High-speed storage for your cloud computers (VMs).

**3. Networking Services**

* **Azure Virtual Network (VNet):** Enables secure communication between Azure resources. Connects your cloud resources securely, like setting up a private network.
* **Azure Load Balancer:** Distributes incoming network traffic across multiple targets to ensure high availability. Distributes incoming traffic to multiple servers to ensure no single server gets overwhelmed.
* **Azure VPN Gateway:** Establishes secure, cross-premises connectivity. Creates secure connections between your on-premises networks and Azure.

**4. Database Services**

* **Azure SQL Database:** Fully managed relational database with built-in intelligence.  A managed database service for storing and managing relational data.
* **Azure Cosmos DB:** Globally distributed, multi-model database service. A globally distributed database service that supports multiple data models.
* **Azure Database for MySQL/PostgreSQL:** Managed database services for MySQL and PostgreSQL. Managed versions of popular open-source databases.

**5. AI and Machine Learning**

* **Azure Machine Learning:** Provides tools and services to build, train, and deploy machine learning models. Tools to build, train, and deploy machine learning models.
* **Azure Cognitive Services:** Offers pre-built APIs for vision, speech, language, and decision-making capabilities. Pre-built APIs for adding AI capabilities like vision, speech, and language to your apps.
* **Azure Bot Service:** Develops intelligent, enterprise-grade bots. Helps you create intelligent chatbots.

**6. Analytics Services**

* **Azure Synapse Analytics:** Integrates big data and data warehousing. Combines big data and data warehousing for powerful analytics.
* **Azure Data Lake Storage:** Scalable data lake solution for high-performance analytics. Stores large amounts of data for analysis.
* **Azure Stream Analytics:** Real-time analytics service for processing fast-moving streams of data.  Processes real-time data streams for immediate insights.

**7. DevOps Services**

* **Azure DevOps:** Provides development collaboration tools including Azure Repos, Azure Pipelines, and Azure Boards. Tools for collaboration, code repositories, and automated testing and deployment.
* **Azure DevTest Labs:** Enables quick creation of environments for testing and development. Quickly create test environments for development and testing.

**8. Security and Identity**

* **Azure Active Directory (AD):** Identity and access management service. Manages user identities and access to resources.
* **Azure Security Center:** Provides unified security management and advanced threat protection. Monitors and protects your cloud resources.
* **Azure Key Vault:** Safeguards cryptographic keys and secrets used by cloud applications and services. Stores and manages sensitive information like passwords and keys.

**9. Internet of Things (IoT)**

* **Azure IoT Hub:** Central message hub for bi-directional communication between IoT applications and devices. Connects and manages IoT devices.
* **Azure IoT Central:** Fully managed IoT app platform that makes it easy to connect, monitor, and manage IoT assets. Simplifies the setup and management of IoT solutions.

**10. Management and Governance**

* **Azure Policy:** Enforces organizational standards and assesses compliance at scale. Ensures your resources comply with company policies.
* **Azure Cost Management:** Helps you monitor and control Azure spending and optimize resource use. Helps you track and control your spending on Azure.
* **Azure Monitor:** Provides full-stack monitoring, advanced analytics, and intelligent insights. Provides insights and monitoring for your applications and infrastructure.

**Describe Azure architecture and services - Azure compute**

* **Azure Virtual Machines**
* Azure Virtual Machines (VMs) are one of the core services offered by Microsoft Azure, providing scalable, on-demand computing resources.
* An **Azure Virtual Machine (VM)** is like a computer that you can create and run in the cloud. It allows you to use the computing power of Azure without needing to buy and maintain physical hardware. You can run different operating systems (like Windows or Linux) and install software on these virtual machines, just like you would on a physical computer.
* Key Features
* [**Wide Range of VM Sizes**:](https://azure.microsoft.com/en-us/products/virtual-machines/) Azure offers different sizes and types of virtual machines (VMs) to suit various needs. Whether you need a general-purpose VM for everyday tasks, a compute-optimized VM for heavy processing, a memory-optimized VM for applications that need a lot of RAM, or a GPU-enabled VM for graphics-intensive tasks, Azure has you covered.
* [**Integration with Other Azure Services**: Easily integrate VMs with other Azure services like Azure Storage, Azure Backup, and Azure Monitor for enhanced functionality](https://azure.microsoft.com/en-us/products/virtual-machines/)
* **Series of virtual machines:** Azure offers a variety of Virtual Machine (VM) series, each designed to meet different needs and workloads.

**1. General Purpose VMs:** These VMs are balanced in terms of CPU and memory, making them suitable for a wide range of applications. These are versatile and can handle a variety of tasks.

* **A-series:** Entry-level VMs for development and testing.
* **B-series:** Economical VMs that provide burstable performance. Cost-effective VMs that can handle occasional spikes in workload.
* **D-series:** General-purpose VMs with a good balance of CPU and memory. Balanced VMs for general use.
* **Dv2-series:** An enhanced version of the D-series with more powerful CPUs.

**2. Compute Optimized VMs:** These VMs are great for tasks that need a lot of processing power.

* **F-series:** High CPU-to-memory ratio, ideal for batch processing, web servers, and analytics. Good for tasks like web servers and data analysis.
* **Fs-series:** Similar to F-series but with SSD storage for faster data access.

**3. Memory Optimized VMs:** These VMs offer a high memory-to-CPU ratio, suitable for memory-intensive applications. These VMs have lots of memory, ideal for heavy-duty applications.

* **E-series:** Great for big applications like SAP.
* **Ev3-series:** Enhanced version of E-series with more powerful CPUs.
* **M-series:** High memory VMs for large databases. Best for very large databases.

**4. Storage Optimized VMs:** These VMs are designed for high disk throughput and IO. These VMs are designed for tasks that need fast and large storage.

* **Lsv2-series:** Perfect for big data and databases.
* **Ls-series:** Similar to Lsv2 but with different setups.

**5. GPU VMs:** These VMs are equipped with powerful GPUs for intensive graphics and compute tasks. These VMs have powerful graphics processors for tasks like gaming and AI.

* **NC-series:** Designed for high-performance computing and AI workloads.  For high-performance computing and AI.
* **ND-series:** For deep learning and AI training.
* **NV-series:** Ideal for remote visualization, streaming, gaming, and encoding. For graphics-heavy tasks like gaming and video rendering.

**6. High Performance Compute (HPC) VMs:** These VMs are designed for the most demanding compute-intensive workloads. These VMs are for the most demanding tasks.

* **H-series:** High-performance VMs for scientific simulations and engineering applications.  For scientific and engineering simulations.
* **HB-series:** For high-bandwidth applications.
* **HC-series:** For complex computations like molecular modeling.
* **Costing for resources in VM**

When you’re using cloud services like Azure, you don’t just pay for one thing, like a virtual machine (VM). Instead, you need to think about the cost of everything you’re using. This includes:

* **Public IP addresses**: These let your VM talk to the internet and might have a small cost.
* **OS-level disks**: These are the hard drives your VM uses to store data.
* **Virtual networks**: These connect your VMs to each other and the internet. Azure doesn’t charge for this.
* **Network interfaces**: These are like the VM’s network card, and there’s no cost for them.
* **Network security groups**: These are like security guards for your network, controlling traffic to and from your VMs, and they’re also free.

For VMs, you’re billed by the minute. So if you use a VM for 20 minutes, you only pay for those 20 minutes. There’s no big fee just to start using a VM.

* **Resource Groups**

**Azure resources: Azure resources** are the building blocks of your cloud infrastructure in Microsoft Azure. These resources can be virtual machines, databases, storage accounts, or any other service offered by Azure. Each resource is a manageable item in Azure, and they are provisioned and managed individually. Diagram from abhishel veermalla azure course

A diagram of a flowchart

Description automatically generated

**Azure Resource Manager (ARM)**

**Azure Resource Manager** is like the control center for all your Azure resources. It helps you manage everything in a consistent way. Here’s what it does:

* **Central Control**: Every time you create, update, or delete something in Azure, it goes through ARM. This ensures everything is managed the same way, no matter how you do it (through the portal, PowerShell, or CLI).
* **Templates**: You can use JSON files (called ARM templates) to describe what you want to create. This makes it easy to set up the same resources again and again.
* **Permissions**: ARM works with Azure's Role-Based Access Control (RBAC) to decide who can do what with your resources.
* **Organization**: You can tag resources to keep things organized and track costs better.
* **Azure Regions**
* Azure regions are specific geographic locations where Microsoft has established datacenters to provide cloud services. These regions are designed to offer high availability and redundancy
* **Geographic Distribution:** Azure regions are spread across the globe, allowing users to deploy resources close to their users for better performance and compliance. Some examples include: **West US, North Europe,** [**Southeast Asia**](https://learn.microsoft.com/en-us/azure/virtual-machines/regions)
* **Region Pairs:** Each Azure region is paired with another region within the same geography. This pairing helps with disaster recovery and data redundancy. For example:

**West US** is paired with **East US**

[**North Europe** is paired with **West Europe**](https://learn.microsoft.com/en-us/azure/virtual-machines/regions)

* **Availability Zones:** Within each region, there are multiple datacenters known as Availability Zones. These zones provide additional redundancy and high availability by being physically separated within the region.
* **Shutting down the virtual machine**

**Using the VM’s Operating System**

1. **Access the VM**: Open the VM window or console.
2. **Initiate Shutdown**:
   * **Windows**: Click on the Start menu, select the Power button, and choose “Shut down”.
   * **Linux**: Use the command sudo shutdown -h now in the terminal.
   * **macOS**: Click on the Apple menu and select “Shut Down”.
3. **Using Azure Portal**

* **Navigate to the VM**: Go to the Azure Portal and select the VM you want to shut down.
* **Stop the VM**: Click on the “Stop” button in the VM’s overview page. This will deallocate the VM and stop billing for compute resources.
* **Creating a Linux Virtual Machine**
* **Step-by-Step Procedure**
* **Sign in to Azure Portal**: Go to the Azure portal and sign in with your Azure account.
* **Create a Virtual Machine**: In the Azure portal, search for “Virtual Machines” in the search bar. Click on “Create” and then select “Azure virtual machine.”
* **Configure Basic Settings: Subscription**, **Resource Group**, **Virtual Machine Name**, **Region**, **Image**: Choose the Linux distribution you want (e.g., Ubuntu Server 22.04 LTS), **Size**, **Authentication Type**: Choose SSH public key or password for authentication, **Username**: Enter a username for the VM, **SSH Public Key**: If using SSH, provide the public key.
* **Configure Disk Options**

**OS Disk Type**: Choose the type of disk (e.g., Standard SSD).

**Data Disks**: Add any additional data disks if needed.

* **Configure Networking**: **Virtual Network**, **Subnet**, **Public IP**, **Inbound Ports**: Allow SSH (port 22) and any other ports you need.
* **Configure Management Options**: Leave the default options or configure as needed for monitoring, backup, etc.
* **Review and Create**
* **Access the VM**: Once the VM is deployed, go to the resource and copy the public IP address.

Use an SSH client (like PuTTY or terminal) to connect to the VM:

ssh username@public\_ip\_address

* **Post-Deployment**: Update and upgrade the packages on your VM: sudo apt update && sudo apt upgrade
* **Clean Up Resources**: When you’re done, remember to delete the resource group to avoid any charges.
* **Deleting resources in Azure**

Deleting resources in Azure is an important step to manage costs and keep your environment clean. Here’s a step-by-step guide on how to delete resources in Azure:

* **Deleting Individual Resources**
* **Sign in to Azure Portal**

Go to the Azure portal and sign in with your Azure account.

* **Navigate to the Resource**

In the left-hand menu, select “All resources.”

Find and click on the resource you want to delete.

* **Delete the Resource**

In the resource’s overview page, click on the “Delete” button.

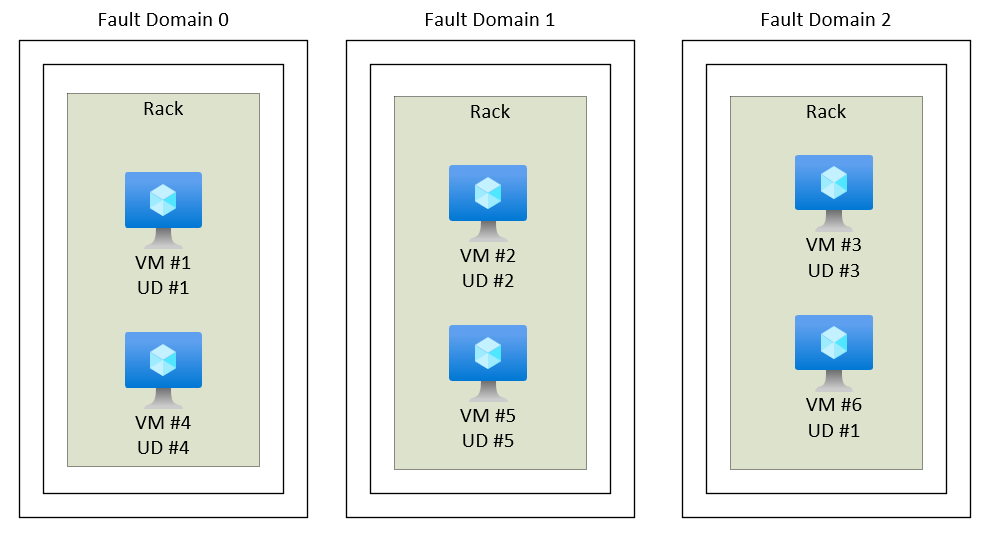
Type the name of the resource to confirm the deletion.

Click “Delete” to remove the resource.

* **Azure Marketplace**

Azure Marketplace is a digital catalog that provides thousands of software applications and services from Microsoft and its partners. These solutions are designed to integrate with Azure and cover various categories such as virtual machines, databases, application development, security, and more. Features:

* **Wide Range of Solutions:** From virtual machines and databases to AI and machine learning tools, you can find a vast array of solutions to meet your needs.
* **Certified and Trusted:** All applications and services in the Marketplace are certified to run on Azure, ensuring compatibility and reliability.
* **Easy Deployment:** Solutions can be deployed directly from the Marketplace to your Azure environment with just a few clicks.
* **Flexible Pricing:** Many solutions offer flexible pricing models, including pay-as-you-go, bring-your-own-license (BYOL), and free trials.
* **Availability Sets**



An **Availability Set** in Azure is a logical grouping of Virtual Machines (VMs) that helps ensure your application remains available during planned or unplanned maintenance events.

An **Availability Set** is a feature in Azure that helps keep your applications running smoothly by grouping your virtual machines (VMs) in a way that protects them from hardware failures and maintenance events.

Here’s a detailed overview:

**Key Features of Availability Sets**

1. **Fault Domains**: These are like separate physical racks in the data center. Each fault domain has its own power supply and network switch. If one fault domain fails (e.g., power outage), only the VMs in that domain are affected. VMs in other fault domains keep running.
2. **Update Domains**: Update domains ensure that not all VMs in your availability set are rebooted at the same time during planned maintenance. VMs are grouped into update domains, and only one update domain is rebooted at a time, so your application stays available.
3. **High Availability**: By placing your VMs in an availability set, you can achieve a higher level of availability for your applications. [Azure guarantees a 99.95% uptime SLA when you use availability sets](https://www.bing.com/aclick?ld=e8CgDtHx8jdNhQ7mUMTU-bXjVUCUxJa2s0vOWpHQxOQnQ3WPPspLrsAUKfAzarPo9ugr6rcnYTquHrNWK-mV5AAX4WFvaF4LiNZLFs_KueGSymC7w-viJbvP2_tb_22CSGZHKWAg2bhT4ljp7sMdC0Moqg4C5so36sf8pc83ARV8UB3Q4h&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa184NGQ4NDQ0YTFjZWExOTM5MjA4ZDIyZjhlMjYyMTVkY19rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfODRkODQ0NGExY2VhMTkzOTIwOGQyMmY4ZTI2MjE1ZGNfa18lMjZtc2Nsa2lkJTNkODRkODQ0NGExY2VhMTkzOTIwOGQyMmY4ZTI2MjE1ZGM&rlid=84d8444a1cea1939208d22f8e26215dc)[1](https://learn.microsoft.com/en-us/azure/virtual-machines/availability-set-overview).
4. **No Extra Cost**: There is no additional cost for using availability sets. [You only pay for the VMs you create](https://learn.microsoft.com/en-us/azure/virtual-machines/availability)

**Example Scenario:** Imagine you have a web application running on three VMs. By placing these VMs in an Availability Set with two Fault Domains and three Update Domains:

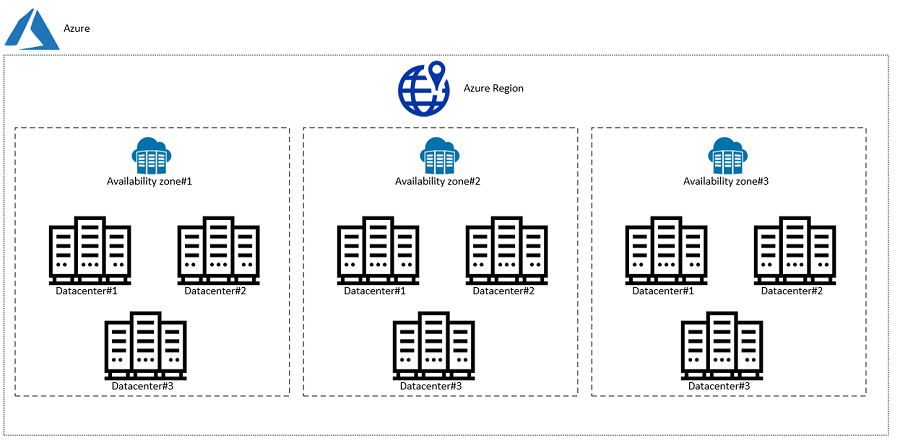
* Two VMs will be placed in different Fault Domains, ensuring that if one rack fails, the other VM continues to run.
* Each VM will be placed in a different Update Domain, ensuring that during maintenance, only one VM is updated at a time, keeping your application available.
* **Stepwise procedure:**

Creating an Availability Set in Azure involves several steps. Here’s a step-by-step guide to help you through the process:

1. **Sign in to the Azure Portal**:
2. **Create a Virtual Machine**:
3. **Configure the Basics**: and then Under “Availability options,” select “Availability set.”
4. **Create a New Availability Set**:
   * Click on “Create new” next to the Availability set option.
   * Enter a name for the Availability set.
   * Select the number of fault domains and update domains.
   * Click “OK” to create the Availability set.
5. **Complete the VM Creation**: Continue configuring the virtual machine settings as needed.

* **Availability Zones (In India, West India (Mumbai), South India (Chennai), Central India (Pune)**

Availability Zones are physically separate locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. By deploying your resources across multiple Availability Zones, you can ensure that your applications remain available even if one zone experiences an outage.



**Example Scenario:** Imagine you have a critical application that needs to be highly available. Instead of running it on VMs in a single datacenter, you can spread your VMs across multiple Availability Zones. This way, if one datacenter (zone) goes down, your application will still be up and running in the other zones.

**Why Not Always Use Availability Zones Instead of Availability Sets?**

* While Availability Zones offer higher availability, they can incur additional costs for data transfer between zones.
* Availability Sets are sufficient for many applications and do not have these additional data transfer costs

**Steps to Create Resources in Availability Zones:**

1. **Sign in to the Azure Portal**:
2. **Create a Virtual Machine**:
3. **Configure the Basics**:
4. **Select the Availability Zone**: Choose the desired availability zone (e.g., Zone 1, Zone 2, Zone 3) for your virtual machine.
5. **Complete the VM Creation**:
6. **Add Additional VMs to Different Zones**

* **Azure Dedicated Hosts**

Azure Dedicated Hosts are physical servers dedicated to your organization. They allow you to run your VMs on hardware that is not shared with other customers, providing more control over the server’s configuration. Azure Dedicated Hosts are physical servers that are reserved just for your organization. This means you don’t share the hardware with anyone else.Here are some key features and benefits of using Azure Dedicated Host:

**Key Features:**

1. [**Dedicated Physical Server**:](https://www.bing.com/aclick?ld=e8r4V2JI2dYLv-xXsywrMWBDVUCUzmklqKx5z_vRYHbBe0PhTPMEoXMvijs1npiOrn-XH_L8Zx1DUF_eTYxEAXHd79iIReXhLUXym_lnZIC1e5JpK8cr2GtIPO-AagKTOu9iaRcak2kj86WtrZEcbjp6-2dVlXLgty20OTqyir_01cbt2E&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmdmlydHVhbC1uZXR3b3JrJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa184ZDI2MzcwZjIzNTMxNWNjMjgxMzUxYmMyYmY0NGI5Y19rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfOGQyNjM3MGYyMzUzMTVjYzI4MTM1MWJjMmJmNDRiOWNfa18lMjZtc2Nsa2lkJTNkOGQyNjM3MGYyMzUzMTVjYzI4MTM1MWJjMmJmNDRiOWM&rlid=8d26370f235315cc281351bc2bf44b9c)  The physical server is dedicated to your organization, ensuring that no other customers share the same hardware. The server is only for your use, so no other customers are using the same hardware. This helps with security and compliance needs.
2. **Control and Customization:** You have control over the host’s maintenance policies, allowing you to schedule maintenance windows that suit your needs. You can choose the type of hardware and the number of VMs to run on the host.

**Setting Up Azure Dedicated Hosts**

1. **Create a Host Group:** In the Azure portal, create a group of dedicated hosts by choosing the region and availability zone.
2. **Allocate Dedicated Hosts:** Add dedicated hosts to your group. Choose the type of hardware and the number of hosts.
3. **Deploy VMs:** Set up your VMs on these dedicated hosts. Manage them like any other Azure VMs.

**Example Scenario:** Imagine you have a financial application that needs to follow strict rules. By using Azure Dedicated Hosts:

* Your VMs run on isolated hardware, meeting compliance needs.
* You control when maintenance happens, reducing disruptions.
* Your application gets consistent performance from dedicated resources.
* **Azure Spot VMs**
* **Azure Spot Virtual Machines** allow you to take advantage of unused Azure compute capacity at significantly reduced costs.
* Spot VMs are a type of virtual machine in Azure that you can use at a much lower cost compared to regular VMs. However, they come with the condition that Azure can take them back (evict them) at any time if it needs the capacity for other tasks.
* These VMs are ideal for workloads that can handle interruptions, such as batch processing jobs, development and testing environments, and large compute workloads. Here are some key points about Azure Spot VMs:

**Key Features:**

1. [**Cost Savings**: Spot VMs offer deep discounts, up to 90% compared to pay-as-you-go prices1](https://azure.microsoft.com/en-us/products/virtual-machines/spot/)[2](https://learn.microsoft.com/en-us/azure/virtual-machines/spot-vms).
2. [**Scalability**: You can scale your workloads efficiently by leveraging Azure’s excess capacity1](https://azure.microsoft.com/en-us/products/virtual-machines/spot/).
3. [**Eviction Policy**:](https://learn.microsoft.com/en-us/azure/virtual-machines/spot-vms)  Azure can evict Spot VMs when it needs the capacity for other higher-priority tasks. This means your Spot VMs might be stopped or deleted with little notice.

**Example Scenario:** Imagine you have a large data processing job that doesn’t need to be completed immediately. By using Spot VMs:

* You can run your job at a much lower cost.
* If the VM gets evicted, you can simply restart the job later without significant impact.
* **Azure Virtual Machine Scale Sets (VMSS)**

Azure Virtual Machine Scale Sets (VMSS) are a service that allows you to create and manage a group of identical, load-balanced VMs. They are designed to support large-scale applications by automatically scaling the number of VMs up or down based on demand. Here’s a simple breakdown:

1. **Group of VMs**: VMSS allows you to create and manage a collection of identical VMs. This is useful when you need multiple VMs to run the same application.
2. **Automatic Scaling**: VMSS can automatically adjust the number of VMs based on the demand. For example, if your application gets more traffic, VMSS can add more VMs to handle the load. [Conversely, it can reduce the number of VMs when the demand decreases, saving costs](https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview)
3. **Uniform Configuration:** All VMs in a scale set are created from the same template, so they are identical. This makes it easier to manage and update them.
4. **Load Balancing**: VMSS works with Azure Load Balancer to spread incoming traffic evenly across all VMs. This ensures your application runs smoothly and efficiently.
5. **High Availability**: VMSS can place VMs in different physical locations (Availability Zones or Fault Domains) to protect against hardware failures and ensure your application stays up and running.

**Steps to Create a Virtual Machine Scale Set:**

1. **Sign in to the Azure Portal**:
2. **Create a Scale Set**:
   * In the search bar, type “Virtual Machine Scale Sets” and select it.
   * Click on the “Add” button to create a new scale set.
3. **Configure the Basics**:
   * Fill in the necessary details such as the scale set name, subscription, resource group, and region.
   * Choose the image for your VMs (e.g., Windows or Linux).
4. **Set Up Scaling Options**:
   * Under the “Scaling” tab, configure the initial instance count and set up autoscaling rules based on metrics like CPU usage, memory usage, or custom metrics.  For example, if CPU usage exceeds 70% for 10 minutes, add an instance. If it drops below 25%, remove an instance.
5. **Configure Networking**: Set up the virtual network, subnet, and load balancer for your scale set.
6. **Review and Create**: Review your configuration settings and click “Create” to deploy the scale set.

**Example Scenario:** Imagine you have an e-commerce website that experiences fluctuating traffic. By using VMSS:

* During peak hours, VMSS can automatically add more VMs to handle the increased traffic.
* During off-peak hours, VMSS can reduce the number of VMs to save costs.
* All VMs are load-balanced to ensure even distribution of traffic and high availability.
* **Azure web app**

Azure Web Apps is a part of the Azure App Service, Azure Web Apps is a service that lets you host your websites and web applications without worrying about the underlying hardware and infrastructure. It supports many programming languages and frameworks, making it easy to deploy and manage your web apps. Features:

* **Fully Managed Platform:** Azure takes care of all the servers, storage, and networking for you. You just focus on your app.It automatically updates and patches the environment to keep it secure and up-to-date.
* **Multiple Language Support:** You can use various programming languages like .NET, Java, Node.js, Python, PHP, and Ruby. You can deploy your app using different methods like Git, FTP, or tools like Visual Studio.
* **Scalability:** You can easily increase or decrease the resources for your app based on the traffic. This means your app can handle more users when needed and save costs when the traffic is low. Auto-scaling features allow your app to adjust automatically to changes in traffic.
* **High Availability:** Built-in load balancing ensures your app remains available and responsive by distributing traffic evenly. Azure guarantees a 99.95% uptime, meaning your app will be available almost all the time.
* **Security:** Provides built-in security features like SSL/TLS certificates, authentication, and integration with Azure Active Directory. Regular security updates and compliance certifications help protect your app and data.
* **Continuous Integration and Deployment (CI/CD):** Integrates with tools like Azure DevOps, GitHub Actions, and Jenkins to automate your build, test, and deployment processes. Supports deployment slots, allowing you to deploy new versions of your app without any downtime.
* **Easy Deployment:** You can quickly deploy and update your web app using the App Service Editor.

**Example:**

Imagine you have a blog website. With Azure Web Apps, you can:

* Develop your blog using a language like Python.
* Deploy it to Azure Web Apps with just a few clicks.
* Azure will handle the server setup, scaling, and security.
* You can focus on writing and publishing your blog posts without worrying about the technical details.

**Step-by-Step Procedure:**

1. **Sign in to the Azure Portal**:
2. **Create a Resource Group**:
3. **Create a Web App**:
   * In the search bar, type “App Services” and select it.
   * Click on “Add” to create a new web app.
   * Fill in the necessary details on the “Basics” tab:
     + **Subscription**: Select your Azure subscription.
     + **Resource Group**: Select the resource group you created.
     + **Name**: Enter a unique name for your web app.
     + **Publish**: Choose “Code” or “Docker Container” based on your deployment preference.
     + **Runtime Stack**: Select the runtime stack (e.g., .NET, Node.js, Python).
     + **Region**: Choose the region where you want to host your web app.
4. **Configure Deployment Settings**:
   * Under the “Deployment” tab, you can configure continuous deployment from a source control system like GitHub, Azure Repos, or Bitbucket.
   * Enable continuous deployment if you want your web app to automatically update with changes from your repository.
5. **Configure Networking and Monitoring**:
6. **Review and Create**:
7. **Deploy Your Code**:
   * Once the web app is created, you can deploy your code. If you set up continuous deployment, your code will be automatically deployed from your repository.
8. **Access Your Web App**:
   * After deployment, you can access your web app using the URL provided in the Azure portal. This URL will be in the format https://<your-web-app-name>.azurewebsites.net.

* **Azure Functions**

**Azure Functions** is a serverless computing service that allows you to run small pieces of code, called functions, without having to manage the underlying infrastructure It’s a serverless platform, meaning you don’t have to worry about provisioning or managing servers. You only pay for the compute resources your functions use when they are running.

**Key Features:**

1. **[Serverless Architecture](https://www.bing.com/aclick?ld=e8QbM8Kq1ylLB3RcIXaQx4rzVUCUyGdQbNqig5SJM0HzKzv55Lx2wRlyzY7JNKyT3ZwtuDoyVDGDRRDN5c9kGwK42zWNSIezg3604HrFlAW-bOzutCGi1yNmSU4763FVa0JQwog93fUOCIDVCiVfXPP9nw-2khJIH6jz2dAASilnTYaQtG&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmZ1bmN0aW9ucyUyZiUzZmVmX2lkJTNkX2tfMmM3ZmI3OGQxNzI3MTc4M2I1NDZkMTNlMDAxNTg4MTlfa18lMjZPQ0lEJTNkQUlEY21tZjFlbGo5djVfU0VNX19rXzJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5X2tfJTI2bXNjbGtpZCUzZDJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5&rlid=2c7fb78d17271783b546d13e00158819" \t "_blank)**[: You don’t have to worry about managing servers. Azure automatically provides the resources needed to run your functions.](https://www.bing.com/aclick?ld=e8QbM8Kq1ylLB3RcIXaQx4rzVUCUyGdQbNqig5SJM0HzKzv55Lx2wRlyzY7JNKyT3ZwtuDoyVDGDRRDN5c9kGwK42zWNSIezg3604HrFlAW-bOzutCGi1yNmSU4763FVa0JQwog93fUOCIDVCiVfXPP9nw-2khJIH6jz2dAASilnTYaQtG&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmZ1bmN0aW9ucyUyZiUzZmVmX2lkJTNkX2tfMmM3ZmI3OGQxNzI3MTc4M2I1NDZkMTNlMDAxNTg4MTlfa18lMjZPQ0lEJTNkQUlEY21tZjFlbGo5djVfU0VNX19rXzJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5X2tfJTI2bXNjbGtpZCUzZDJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5&rlid=2c7fb78d17271783b546d13e00158819" \t "_blank) **[You only pay for the time your functions are running, which can save money.](https://www.bing.com/aclick?ld=e8QbM8Kq1ylLB3RcIXaQx4rzVUCUyGdQbNqig5SJM0HzKzv55Lx2wRlyzY7JNKyT3ZwtuDoyVDGDRRDN5c9kGwK42zWNSIezg3604HrFlAW-bOzutCGi1yNmSU4763FVa0JQwog93fUOCIDVCiVfXPP9nw-2khJIH6jz2dAASilnTYaQtG&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmZ1bmN0aW9ucyUyZiUzZmVmX2lkJTNkX2tfMmM3ZmI3OGQxNzI3MTc4M2I1NDZkMTNlMDAxNTg4MTlfa18lMjZPQ0lEJTNkQUlEY21tZjFlbGo5djVfU0VNX19rXzJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5X2tfJTI2bXNjbGtpZCUzZDJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5&rlid=2c7fb78d17271783b546d13e00158819" \t "_blank)**
2. **[Event-Driven: Azure Functions can be triggered by different events, such as a web request, a timer, a database change or a new message in a queue. This makes it great for building applications that react to events.](https://azure.microsoft.com/en-us/products/functions/" \t "_blank)**
3. [**Multiple Languages**: Supports various programming languages including C#, Java, JavaScript, Python, and PowerShell](https://www.bing.com/aclick?ld=e8QbM8Kq1ylLB3RcIXaQx4rzVUCUyGdQbNqig5SJM0HzKzv55Lx2wRlyzY7JNKyT3ZwtuDoyVDGDRRDN5c9kGwK42zWNSIezg3604HrFlAW-bOzutCGi1yNmSU4763FVa0JQwog93fUOCIDVCiVfXPP9nw-2khJIH6jz2dAASilnTYaQtG&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmZ1bmN0aW9ucyUyZiUzZmVmX2lkJTNkX2tfMmM3ZmI3OGQxNzI3MTc4M2I1NDZkMTNlMDAxNTg4MTlfa18lMjZPQ0lEJTNkQUlEY21tZjFlbGo5djVfU0VNX19rXzJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5X2tfJTI2bXNjbGtpZCUzZDJjN2ZiNzhkMTcyNzE3ODNiNTQ2ZDEzZTAwMTU4ODE5&rlid=2c7fb78d17271783b546d13e00158819)[1](https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview).
4. [**Scalability**: Automatically scales based on demand, ensuring efficient resource utilization2](https://azure.microsoft.com/en-us/products/functions/).
5. [**Integration**: Easily integrates with other Azure services like Azure Blob Storage, Azure Cosmos DB, and Azure Event Hubs](https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview)

**Steps to Create an Azure Function:**

1. **Sign in to the Azure Portal**:
2. **Create a Function App**:
3. **Configure Hosting**:
4. **Set Up Monitoring**:
5. **Review and Create**:
6. **Create a Function**:
   * Once the Function App is created, go to the Function App in the Azure portal.
   * Click on “Functions” in the left-hand menu and then “Add” to create a new function.
   * Choose a template based on the trigger type (e.g., HTTP trigger, Timer trigger, Blob trigger).
   * Configure the function settings and write your code in the provided editor.
7. **Test and Deploy**:

**Example Scenario:** Imagine you have an online store and want to send a confirmation email whenever a new order is placed. By using Azure Functions:

* You can create a function that is triggered by a new message in a queue (representing a new order).
* The function reads the order details from the message and sends a confirmation email to the customer.
* This function scales automatically to handle any number of orders without you having to manage the infrastructure.
* **Azure Containers and Kubernetes**
* **Containers:** Containers are a form of virtualization that allows you to run applications in isolated environments. Containers are like small, portable boxes that hold everything an application needs to run. This includes the code, libraries, and settings. They make sure the application runs the same way no matter where you deploy it. Unlike traditional virtual machines (VMs), containers share the host system’s operating system (OS) kernel but run in their own isolated user space.
* **Features:**
* Isolation: Each container runs separately from others, so they don’t interfere with each other.
* Portability: Containers package everything an application needs, making it easy to move and run the app anywhere, like on different computers or in the cloud.

**Example Scenario**

* Imagine you have a web application with different parts (like frontend, backend, and database). By using containers:
* Each part can run in its own container, ensuring they don’t interfere with each other.
* You can easily move the application from your development environment to testing and production without compatibility issues.
* Orchestration tools like Kubernetes can manage the deployment and scaling of your containers, ensuring high availability and performance.
* **Azure Container Instances (ACI):** Azure Container Instances is a service that lets you run containers without worrying about the underlying servers. It’s a quick and easy way to deploy containers in the cloud. **Features**
* Simplicity: ACI is straightforward to use. You can deploy containers directly without managing servers. You can use the Azure portal, command line, or templates to set it up.
* Fast Start-Up: Containers in ACI start in seconds, making it great for tasks that need quick, on-demand compute power.
* Cost-Effective: You only pay for the resources your containers use, billed per second. This is cost-effective for short tasks.
* Isolation: Each container runs in its own isolated environment, ensuring security and performance.

**Example Scenario: Processing Orders:** You have a task that processes orders whenever a customer makes a purchase. This task needs to run quickly and doesn’t require constant uptime.

1. Deploying with ACI:
   * You create a container image that includes the code for processing orders.
   * You use ACI to deploy this container. Since ACI starts containers quickly, it can handle the order processing task on-demand.
   * When a new order comes in, an event triggers the ACI container to start, process the order, and then stop.

* **Azure Kubernetes Service (AKS)**

AKS is a managed Kubernetes service that simplifies deploying, managing, and scaling containerized applications using Kubernetes. Kubernetes is a system for managing containerized applications.

**Key Features of AKS:**

1. [**Managed Kubernetes**:](https://learn.microsoft.com/en-us/azure/aks/what-is-aks) AKS handles the complex parts of running Kubernetes, like upgrades and scaling. You get a fully managed Kubernetes control plane, so you can focus on your applications.
2. [**Scalability**:](https://learn.microsoft.com/en-us/azure/aks/what-is-aks) AKS can automatically add or remove resources based on demand, ensuring your applications run smoothly. Supports both adding more containers (horizontal scaling) and adding more resources to existing containers (vertical scaling).

**Scenario: Running the Recommendation Engine:** Your recommendation engine needs to be highly available and scalable to handle varying loads, especially during peak shopping times.

1. **Deploying with AKS:**
   * You create a set of container images for the recommendation engine.
   * You use AKS to deploy these containers across a Kubernetes cluster. AKS manages the underlying infrastructure, including scaling and load balancing.
   * AKS automatically scales the number of containers based on the load, ensuring the recommendation engine can handle high traffic during peak times.

**Step-by-Step Procedure to Create an AKS Cluster:**

* **Step 1: Sign in to the Azure Portal**
* **Step 2: Create a Resource Group**
* **Step 3: Create an AKS Cluster**

In the search bar, type “Kubernetes services” and select it.

Click on “Add” to create a new Kubernetes cluster.

Fill in the necessary details on the “Basics” tab:

* 1. **Subscription**: Select your Azure subscription.
  2. **Resource Group**: Select the resource group you created.
  3. **Kubernetes Cluster Name**: Enter a unique name for your cluster.
  4. **Region**: Choose the region where you want to host your cluster.
  5. **Kubernetes Version**: Select the version of Kubernetes you want to use.
  6. **Node Size**: Choose the size of the VMs for your nodes.
  7. **Node Count**: Specify the initial number of nodes.
* **Step 4: Configure Node Pools:** Under the “Node pools” tab, configure the node pool settings such as node size, count, and scaling options.
* **Step 5: Configure Authentication and Networking**

Under the “Authentication” tab, configure the authentication method (e.g., Azure Active Directory).

Under the “Networking” tab, configure the network settings such as virtual network, subnet, and network policies.

* **Step 6: Review and Create**
* **Step 7: Connect to the AKS Cluster**

Once the cluster is created, you can connect to it using the Azure CLI.

Install the Azure CLI and Kubernetes CLI (kubectl) if you haven’t already.

* **What is Azure Virtual Desktop - Creating a host pool**

**Azure Virtual Desktop (AVD)** is a cloud-based service from Microsoft that allows you to create and manage virtual desktops and applications. Think of it as having your computer’s desktop and apps available on the internet, so you can access them from anywhere, using any device like a laptop, tablet, or even a smartphone.

**Key Components of Azure Virtual Desktop**

1. **Host Pools**: A host pool is a collection of virtual machines (VMs) that provide the resources for your virtual desktops. Each VM in the host pool runs a version of Windows that users can connect to.
2. **Session Hosts**: These are the VMs within the host pool that users connect to. They handle the user sessions and run the applications.
3. **Workspaces**: A workspace is a collection of application groups. It provides users with access to their virtual desktops and applications.
4. **Application Groups**: These are groups of applications that you can assign to users. You can have multiple application groups within a workspace.
5. **User Profiles**: User profiles store user-specific settings and data. They ensure that users have a consistent experience across sessions.

**How Does Azure Virtual Desktop Work?**

1. **Setting Up**: You start by setting up your Azure environment, which includes creating a resource group to organize your resources.
2. **Creating a Host Pool**: You create a host pool by specifying the number of VMs, their size, and the operating system. You configure network settings and assign users to the host pool.
3. **Deploying Virtual Desktops**: Once the host pool is set up, you deploy virtual desktops. Users can then connect to these desktops from their devices.
4. **Accessing Virtual Desktops**: Users access their virtual desktops through a web browser or a dedicated client application. They log in with their credentials and can use their desktop and applications just like they would on a physical computer.

**Example Scenario:** Tech Solutions uses AVD for remote work. Employees like Sarah and John access their virtual desktops from home or on personal devices, ensuring flexibility, security, and efficient management.

This setup helps Tech Solutions provide a flexible, secure, and efficient working environment for their employees.

**Describe Azure architecture and services - Networking**

* **Azure Virtual Network (While currently working, we have free access so the VNet is provided by azure is free, but when in company we need to create our own VNet and Subnets)**

An Azure Virtual Network (VNet) is like your own private network in the cloud. It helps your cloud resources, like virtual machines (VMs), talk to each other securely. It allows you to securely communicate between Azure resources, the internet, and on-premises networks.

1. **Isolation:** VNets provide isolation within the Azure environment. You can create multiple VNets within your Azure subscription, and each VNet is isolated from others unless you explicitly connect them. Think of VNets as separate rooms in a house. Each room (VNet) is private and secure unless you open a door (connect them).
2. **Public and private IP address**: Public IP addresses are used for communication between Azure resources and the internet or other public-facing services. **Private IP addresses** are used for communication within an Azure Virtual Network (VNet) and between connected VNets or on-premises networks. **Public IP**: If you need to expose a web application to users on the internet, you would assign a public IP to the VM or load balancer hosting the application. **Private IP**: For backend services like databases that only need to communicate with other internal services, a private IP is used to ensure they are not exposed to the internet.
3. **Subnets**: Within a VNet, you can create subnets to segment your network further. Subnets help organize and secure your resources by dividing the VNet into smaller, manageable sections. Inside each room (VNet), you can create smaller sections (subnets) to organize your stuff better.
4. **Classless Inter Domain Routing (CIDR):** CIDR is a way to organize IP addresses into groups, making it easier to manage and use them. Instead of using the old class-based system (Class A, B, C), CIDR allows for more flexible and efficient allocation of IP addresses. **Example:** Let's say you have an IP address: **192.168.1.0/24**.

**192.168.1.0** is the network address.

**/24** means that the first 24 bits are used for the network part, and the remaining 8 bits are used for host addresses within that network.

This means the network can have IP addresses ranging from **192.168.1.0** to **192.168.1.255**.

1. **Network Security Groups (NSGs)**: NSGs are used to control inbound and outbound traffic to network interfaces (NICs), VMs, and subnets. They contain security rules that allow or deny traffic based on source and destination IP addresses, ports, and protocols. NSGs are like security guards. They control who can enter or leave each section (subnet) based on rules you set.
2. **Virtual Network Peering**: VNet peering allows you to connect VNets, enabling resources in different VNets to communicate with each other. Peered VNets appear as a single network for connectivity purposes. VNet peering is like creating a doorway between two rooms (VNets) so they can share information.
3. **Azure VPN Gateway**: VPN Gateway provides secure cross-premises connectivity between your Azure VNet and your on-premises network. It supports both site-to-site and point-to-site VPN connections. This is like a secure tunnel that connects your cloud network to your home or office network.
4. **Azure ExpressRoute**: ExpressRoute allows you to extend your on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. This provides higher security, reliability, and speed compared to typical internet connections. ExpressRoute is a super-fast, private connection between your on-premises network and Azure, like a dedicated highway.
5. **DNS and Custom DNS**: Azure provides default DNS services for name resolution within VNets. You can also configure custom DNS servers for more control over name resolution. DNS is like a phonebook that helps your resources find each other by name instead of IP addresses.

Here’s a step-by-step guide to creating a virtual network in Azure:

**Step-by-Step Guide to Create an Azure Virtual Network**

1. **Sign in to Azure Portal**:
2. **Navigate to Virtual Networks**: In the search bar at the top, type “Virtual Networks” and select it from the search results.
3. **Create a Virtual Network**: Click on “Create” to start the process.
4. **Basics Tab**:
   * **Subscription**: Select your Azure subscription.
   * **Resource Group**: Choose an existing resource group or create a new one.
   * **Name**: Enter a name for your virtual network.
   * **Region**: Select the Azure region where you want to deploy the virtual network.
5. **IP Addresses**:
   * **IPv4 Address Space**: Define the address space using CIDR notation (e.g., 10.0.0.0/16).
   * **Subnets**: Add one or more subnets within the address space. Each subnet should have a unique address range (e.g., 10.0.1.0/24).
6. **Security**:
   * **DDoS Protection**: Choose whether to enable DDoS protection.
   * **Firewall**: Optionally, you can configure Azure Firewall for additional security.
7. **Tags**:
   * Add tags to organize your resources (optional).
8. **Review + Create**:
   * Review all the settings you have configured.
   * Click on “Create” to deploy the virtual network.
9. **Monitor Deployment**:
   * Monitor the deployment process in the Azure portal. Once completed, your virtual network will be ready for use.

A drawing of a computer screen

Description automatically generated with medium confidence

Diagram: Abhishek Veermalla azure course

* **Communication across virtual machines within a virtual network**

Communication across virtual machines (VMs) within a virtual network (VNet) in Azure is straightforward and secure. Here’s how it works and how you can set it up:

**How Communication Works**

1. **Same VNet**: When VMs are in the same subnet, they can communicate directly with each other using their private IP addresses. It’s like having computers connected to the same local network in an office.
2. **Different Subnets**: If VMs are in different subnets within the same VNet, they can still communicate with each other using their private IP addresses. Azure automatically routes the traffic between subnets.
3. **Network Security Groups (NSGs)**: NSGs control the traffic flow between VMs. You can set rules to allow or deny traffic based on IP addresses, ports, and protocols. Think of NSGs as security checkpoints that ensure only authorized traffic passes through.

**Example Scenario:** Imagine you have two VMs, VM1 and VM2, in the same VNet but different subnets:

* **VM1** is in Subnet1 with IP address 10.0.1.4.
* **VM2** is in Subnet2 with IP address 10.0.2.5.

To communicate, VM1 can directly send a request to 10.0.2.5, and Azure will route the traffic to VM2. If you have an NSG applied to Subnet2, you need to ensure it allows traffic from Subnet1.

* **Network Security Groups**

Network Security Groups (NSGs) in Azure are used to filter network traffic to and from Azure resources. A NSG is like a virtual firewall. It contains a list of security rules that allow or deny traffic based on various criteria such as source and destination IP addresses, ports, and protocols. Here’s a detailed step-by-step guide to creating and managing NSGs:

**Key Components of NSGs**

1. **Security Rules**:
   * **Inbound Rules**: Control incoming traffic to your resources.
   * **Outbound Rules**: Control outgoing traffic from your resources.
   * Each rule specifies:
     + **Priority**: Determines the order in which rules are applied. Lower numbers have higher priority.
     + **Source/Destination**: IP address or range.
     + **Port**: Specific port or range of ports.
     + **Protocol**: TCP, UDP, or Any.
     + **Action**: Allow or Deny.
2. **Default Rules**: NSGs come with default rules that allow basic connectivity and block all other traffic. These rules can be overridden by custom rules.

**How NSGs Work**

* **Association: NSGs can be associated with subnets or individual network interfaces (NICs) attached to VMs. NSGs cannot be associated directly with the virtual networks**
  + **Subnet Association**: Rules apply to all resources within the subnet.
  + **NIC Association**: Rules apply only to the specific VM’s NIC.
* **Evaluation**: When traffic reaches a resource, Azure evaluates the NSG rules in order of priority. The first rule that matches the traffic criteria is applied.

**Example Scenario**: Imagine you have a web server VM that needs to allow HTTP (port 80) and HTTPS (port 443) traffic but block all other traffic:

1. **Create NSG**: Create an NSG in the Azure portal.
2. **Add Inbound Rules**:
   * Allow HTTP: Source: Any, Source Port: Any, Destination: Any, Destination Port: 80, Protocol: TCP, Action: Allow, Priority: 100.
   * Allow HTTPS: Source: Any, Source Port: Any, Destination: Any, Destination Port: 443, Protocol: TCP, Action: Allow, Priority: 200.
3. **Associate NSG**: Associate the NSG with the subnet or NIC of the web server VM.

* **Application Security Group (Watch the udemy video for better understanding)**

An ASG is a logical grouping of virtual machine (VM) that allows you to apply network security rules to a group of VMs based on their application roles, rather than their IP addresses. This makes it easier to manage and scale your security policies. An Application Security Group (ASG) is like a label you can put on your virtual machines (VMs) to group them based on their roles, such as web servers or database servers. This makes it easier to manage security rules for these groups instead of handling each VM individually.

**Key Features of Application Security Groups**

1. **Simplified Management**: ASGs allow you to group VMs dynamically based on their application roles. You can group VMs by their roles (e.g., all web servers together) and apply security rules to the whole group.
2. **Dynamic Membership:** VMs can be added or removed from an ASG easily. When a VM joins an ASG, it automatically gets the security rules for that group.

**How ASGs Work**

* **Create ASGs**: You create ASGs in the Azure portal or using command-line tools.
* **Assign VMs**: You assign VMs to ASGs by linking their network interfaces (NICs) to the ASG.
* **Set Rules**: You create security rules in NSGs that reference ASGs to control traffic.

**Example Scenario**: Imagine you have three types of servers: web servers, application servers, and database servers.

1. **Create ASGs**:
   * WebServersASG
   * AppServersASG
   * DBServersASG
2. **Assign VMs**:
   * Put your web server VMs in WebServersASG.
   * Put your application server VMs in AppServersASG.
   * Put your database server VMs in DBServersASG.
3. **Set Rules**:
   * Allow web servers to talk to application servers on port 80 (HTTP).
   * Allow application servers to talk to database servers on port 1433 (SQL).
   * Block all other traffic.

* **Azure Virtual Network Peering (Udemy video)**

**A drawing of a face

Description automatically generated**

Azure Virtual Network Peering allows you to seamlessly connect two or more virtual networks in Azure. This connection enables resources in different virtual networks to communicate with each other as if they were part of the same network. Here’s a detailed overview and step-by-step guide:

**Key Features of Virtual Network Peering**

1. **Low Latency and High Bandwidth**: The connection between peered VNets is fast and reliable because it uses the Microsoft’s internal network, ensuring low latency and high bandwidth.
2. **Private Communication**: Traffic between peered virtual networks is private and does not go through the public internet. Resources in peered VNets can communicate using private IP addresses, just like they would within a single network.
3. **Cross-Region Peering**: You can peer virtual networks across different Azure regions (Global VNet Peering).
4. **No Downtime**: Creating or deleting a peering connection does not cause any downtime to resources in either virtual network. You can set up peering connections without any downtime, meaning your resources stay available.

* **VPN Connections to Azure**

VPN connections to Azure allow you to securely connect your on-premises network and individual devices to your Azure Virtual Network (VNet) over the internet. Here’s a detailed overview and step-by-step guide for setting up different types of VPN connections:

**Types of VPN Connections**

1. **Site-to-Site (S2S) VPN**: This type of VPN connects your on-premises network to an Azure VNet. It’s like extending your local network to the cloud. Uses IPsec/IKE (IKEv1 or IKEv2) for secure communication. Requires a VPN device or a compatible router on your on-premises network. Imagine a company with offices in New York and Los Angeles. A site-to-site VPN can connect the local network in New York to the local network in Los Angeles. [This setup allows employees in both offices to access shared resources like file servers and databases as if they were on the same local network](https://www.paloaltonetworks.com/cyberpedia/what-is-a-site-to-site-vpn).
2. **Point-to-Site (P2S) VPN**: This type of VPN allows individual devices (like laptops) to connect to an Azure VNet. Uses SSTP (Secure Socket Tunneling Protocol) or OpenVPN for secure communication. Ideal for remote workers who need to access resources in Azure.
3. **VNet-to-VNet VPN**: This type of VPN connects two Azure VNets. It’s useful for connecting VNets in different regions or subscriptions. Uses IPsec/IKE for secure communication.

**Components of a VPN Connection**

1. **Virtual Network Gateway**: A gateway that handles the VPN connection. You need to create a VPN gateway in your Azure VNet. There are two types of gateways: VPN Gateway (for VPN connections) and ExpressRoute Gateway (for ExpressRoute connections).
2. **Local Network Gateway**: Represents your on-premises network. You need to define the IP address of your on-premises VPN device and the address space of your on-premises network.
3. **Connection**: The connection between the VPN gateway and the local network gateway. This defines the type of VPN (S2S, P2S, or VNet-to-VNet) and the shared key for encryption.

* **Azure ExpressRoute**

Azure ExpressRoute is a service that lets you create a private, dedicated connection between your on-premises network and Microsoft Azure. This connection doesn’t use the public internet, making it more secure and reliable. This service offers several key benefits, including enhanced security, reliability, and performance compared to typical internet connections.

**Key Features of Azure ExpressRoute**

1. **Private Connectivity**: Establishes a private connection between your on-premises network and Azure, bypassing the public internet.
2. **High Performance**: ExpressRoute offers high-speed connections, which are great for transferring large amounts of data quickly.
3. **Consistent Performance:** Because the connection is private, you get more consistent and reliable performance compared to regular internet connections.
4. **Enhanced Security**: Offers a more secure connection by avoiding the public internet.
5. **Global Reach**: You can connect to Microsoft services around the world, including Azure, Office 365, and Dynamics 365, using one ExpressRoute connection.

**Difference between Azure VPN and Azure expressroute:** In short, **Azure VPN** is good for smaller, less critical projects and is cheaper, while **Azure ExpressRoute** is better for important projects that need high performance and security, but it costs more.

**Example**: Imagine you have a company with data centers in different locations. You want a secure and reliable connection to Azure:

1. **Set Up ExpressRoute**:
   * Work with a provider to establish the physical connection.
   * Create an ExpressRoute circuit in Azure.
2. **Configure Routing**:
   * Set up private peering to connect your network to Azure VNets.
   * Optionally, set up Microsoft peering to access services like Office 365.
3. **Test and Monitor**: Make sure the connection works and monitor its performance.

* **Azure Load Balancer**

Azure Load Balancer is a service that helps distribute incoming network traffic across multiple healthy resources, such as virtual machines (VMs) or instances in a Virtual Machine Scale Set. This ensures that no single VM gets overwhelmed, making your application more reliable and available. [It operates at Layer 4 of the OSI model, handling TCP and UDP traffic](https://www.bing.com/aclick?ld=e8BT9XwrRCL0KmlypjvGMbajVUCUyrXBcOlNXhkObq3TI5ydc0-0ga9D31vkfPDfLt-WYn99Y-Rf7nHXuErQyYs8BdbqD95FEMeWWhqsQ7UcUR3M8PGK4jlII-Jl3JDDtFPYNTeLfVCBHxqtXUxjxOVmUqzQIbxT8vw1TEG0VFeqr3HVCQ&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmxvYWQtYmFsYW5jZXIlMmYlM2ZlZl9pZCUzZF9rXzdhMjNmMjgwN2Y3MzExNzZjYzY3OTQzNDNiYjI0OThmX2tfJTI2T0NJRCUzZEFJRGNtbWYxZWxqOXY1X1NFTV9fa183YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zl9rXyUyNm1zY2xraWQlM2Q3YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zg&rlid=7a23f2807f731176cc6794343bb2498f)[1](https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-overview).

Here are some key features of Azure Load Balancer:

* [**Scalability**: It can handle millions of flows for all TCP and UDP applications, ensuring high throughput and low latency](https://www.bing.com/aclick?ld=e8BT9XwrRCL0KmlypjvGMbajVUCUyrXBcOlNXhkObq3TI5ydc0-0ga9D31vkfPDfLt-WYn99Y-Rf7nHXuErQyYs8BdbqD95FEMeWWhqsQ7UcUR3M8PGK4jlII-Jl3JDDtFPYNTeLfVCBHxqtXUxjxOVmUqzQIbxT8vw1TEG0VFeqr3HVCQ&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmxvYWQtYmFsYW5jZXIlMmYlM2ZlZl9pZCUzZF9rXzdhMjNmMjgwN2Y3MzExNzZjYzY3OTQzNDNiYjI0OThmX2tfJTI2T0NJRCUzZEFJRGNtbWYxZWxqOXY1X1NFTV9fa183YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zl9rXyUyNm1zY2xraWQlM2Q3YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zg&rlid=7a23f2807f731176cc6794343bb2498f)[1](https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-overview).
* [**High Availability**:](https://www.bing.com/aclick?ld=e8BT9XwrRCL0KmlypjvGMbajVUCUyrXBcOlNXhkObq3TI5ydc0-0ga9D31vkfPDfLt-WYn99Y-Rf7nHXuErQyYs8BdbqD95FEMeWWhqsQ7UcUR3M8PGK4jlII-Jl3JDDtFPYNTeLfVCBHxqtXUxjxOVmUqzQIbxT8vw1TEG0VFeqr3HVCQ&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmxvYWQtYmFsYW5jZXIlMmYlM2ZlZl9pZCUzZF9rXzdhMjNmMjgwN2Y3MzExNzZjYzY3OTQzNDNiYjI0OThmX2tfJTI2T0NJRCUzZEFJRGNtbWYxZWxqOXY1X1NFTV9fa183YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zl9rXyUyNm1zY2xraWQlM2Q3YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zg&rlid=7a23f2807f731176cc6794343bb2498f)  It spreads the traffic across multiple VMs, so if one VM fails, others can still handle the load.
* [**Health Probes**: These monitor the health of your backend resources and ensure traffic is only sent to healthy instances](https://www.bing.com/aclick?ld=e8BT9XwrRCL0KmlypjvGMbajVUCUyrXBcOlNXhkObq3TI5ydc0-0ga9D31vkfPDfLt-WYn99Y-Rf7nHXuErQyYs8BdbqD95FEMeWWhqsQ7UcUR3M8PGK4jlII-Jl3JDDtFPYNTeLfVCBHxqtXUxjxOVmUqzQIbxT8vw1TEG0VFeqr3HVCQ&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmxvYWQtYmFsYW5jZXIlMmYlM2ZlZl9pZCUzZF9rXzdhMjNmMjgwN2Y3MzExNzZjYzY3OTQzNDNiYjI0OThmX2tfJTI2T0NJRCUzZEFJRGNtbWYxZWxqOXY1X1NFTV9fa183YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zl9rXyUyNm1zY2xraWQlM2Q3YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zg&rlid=7a23f2807f731176cc6794343bb2498f). It checks the health of your VMs. If a VM is not working properly, it stops sending traffic to it until it recovers.
* [**Outbound Connections**:](https://www.bing.com/aclick?ld=e8BT9XwrRCL0KmlypjvGMbajVUCUyrXBcOlNXhkObq3TI5ydc0-0ga9D31vkfPDfLt-WYn99Y-Rf7nHXuErQyYs8BdbqD95FEMeWWhqsQ7UcUR3M8PGK4jlII-Jl3JDDtFPYNTeLfVCBHxqtXUxjxOVmUqzQIbxT8vw1TEG0VFeqr3HVCQ&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZmxvYWQtYmFsYW5jZXIlMmYlM2ZlZl9pZCUzZF9rXzdhMjNmMjgwN2Y3MzExNzZjYzY3OTQzNDNiYjI0OThmX2tfJTI2T0NJRCUzZEFJRGNtbWYxZWxqOXY1X1NFTV9fa183YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zl9rXyUyNm1zY2xraWQlM2Q3YTIzZjI4MDdmNzMxMTc2Y2M2Nzk0MzQzYmIyNDk4Zg&rlid=7a23f2807f731176cc6794343bb2498f) It manages outgoing connections from your VMs, ensuring they use a consistent IP address.
* **Load Balancing Rules:** You can set rules to decide how the traffic should be distributed based on different criteria like port numbers.

**Types of Azure Load Balancer**

1. **Public Load Balancer**: Distributes traffic from the internet to your VMs. It provides a public IP address for your application.
2. **Internal Load Balancer**: Distributes traffic within your private network in Azure. It is used for internal applications that are not exposed to the internet.

**Example:** Imagine you have a web application running on multiple VMs:

1. **Create a Public Load Balancer**: Set up a public load balancer in the Azure portal.
2. **Define Backend Pool**: Add your web server VMs to the backend pool.
3. **Configure Health Probes**: Set up health checks to monitor your web servers.
4. **Set Load Balancing Rules**: Create a rule to distribute web traffic on port 80 to your web servers.
5. **Test and Monitor**: Ensure the load balancer is working correctly and monitor the health of your web servers.

* **Azure DNS**

Azure DNS is a cloud-based service that allows you to manage and host DNS domains, providing name resolution for your applications and services. It uses the same infrastructure as other Microsoft services, ensuring high availability and performance. **Key Features**

1. **Global Reach**: Azure DNS uses a network of servers around the world to quickly respond to requests for your domain name.
2. **High Availability**: Your DNS records are always accessible, ensuring your website or service is always reachable.
3. **Scalability**: It can handle a large number of requests, so it works well even if your site gets a lot of traffic.
4. **Custom Domain Names**: You can use your own domain names for your applications, giving them a professional look.

**How It Works**

1. **DNS Zones**: A DNS zone is like a container for all the DNS records for a domain. For example, the zone for example.com would include records for www.example.com, mail.example.com, etc.
2. **DNS Records**: These are entries that map your domain names to IP addresses or other resources. Common types include:
   * + **A Record**: Links a domain name to an IPv4 address.
     + **AAAA Record**: Links a domain name to an IPv6 address.
     + **CNAME Record**: Links a domain name to another domain name.
     + **MX Record**: Specifies mail servers for your domain.
     + **TXT Record**: Provides text information for your domain.
     + **SRV Record**: Specifies services available for your domain.
3. **Name Servers**: Azure DNS uses a network of servers to respond to DNS queries. When you create a DNS zone, Azure gives you a set of name servers to use.

**Example:** Imagine you have a website called example.com:

1. **Create a DNS Zone**: In the Azure portal, create a DNS zone for example.com.
2. **Add DNS Records**:
   * Add an A record to link www.example.com to your web server’s IP address.
   * Add an MX record to specify your mail servers.
3. **Delegate Your Domain**: Update your domain registrar to use the Azure DNS name servers provided when you created the DNS zone.
4. **Test and Monitor**: Use tools to check that your domain name resolves correctly and monitor its performance.

**Describe Azure architecture and services – Storage**

A diagram of a system

Description automatically generated with medium confidence

* **Azure Storage**

Azure Storage is a comprehensive cloud storage solution provided by Microsoft Azure, designed to handle a wide range of data storage needs. Azure Storage provide a scalable, durable, and secure platform for storing data in the cloud. [They support various types of data storage, including blobs, files, queues, and tables](https://www.bing.com/aclick?ld=e8lidA1zAmY9rMTGdW3xCYETVUCUyYgC2gvtRORzfnbKfQs6hg7w8QyH1sY-jMQ0foNiowCJejf2a6hUw2xxdufEy4q2_r-uydmlEJQ5l2Oq8ImQSkbTWalVe1KVaJclrn3PfJSfxXDPnSGebA5EZXDI2IoPug67Zz6PxiMt9856wFdhz0&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc3RvcmFnZSUyZnNlYXJjaCUyZiUzZmVmX2lkJTNkX2tfYTUzZTk4ZDBkMDRkMTJkZTJkMDRmZTY0ODI2YzgwZTdfa18lMjZPQ0lEJTNkQUlEY21tZjFlbGo5djVfU0VNX19rX2E1M2U5OGQwZDA0ZDEyZGUyZDA0ZmU2NDgyNmM4MGU3X2tfJTI2bXNjbGtpZCUzZGE1M2U5OGQwZDA0ZDEyZGUyZDA0ZmU2NDgyNmM4MGU3&rlid=a53e98d0d04d12de2d04fe64826c80e7)[1](https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview).

**Types of Azure Storage**

1. **Blob Storage**: It is used to store large amounts of unstructured data like text or images.
   * **Types**:
     + **Block Blobs**: Good for storing files like documents or videos.
     + **Append Blobs**: Great for adding new data to existing files, like logs.
     + **Page Blobs**: Used for data that needs to be read and written quickly, like virtual hard drives.
2. **File Storage**: Azure File Share is like having a file server in the cloud. You can store files there and access them from anywhere, just like you would with a regular file server in your office.
   * **Features**: Can be used by multiple computers at the same time and can sync with on-premises servers.
3. **Queue Storage**: Designed for storing large numbers of messages that can be accessed from anywhere via authenticated calls. It is a service for storing messages that can be processed later.
   * **Use Case**: Useful for building applications where different parts need to communicate without being directly connected.
4. **Table Storage**: It is a NoSQL database for storing large amounts of structured data.
   * **Features**: Flexible and can handle lots of data quickly.

**Step-by-Step Guide to Create an Azure Storage Account:**

**Step 1: Sign in to Azure Portal**

**Step 2: Create a Storage Account**

1. **Navigate to Storage Accounts**: In the left-hand menu, select **Storage accounts**.
2. **Click on + Create**: This will open the storage account creation wizard.
3. **Enter Basic Information**:
   * **Subscription**: Select your Azure subscription.
   * **Resource Group**: Choose an existing resource group or create a new one.
   * **Storage Account Name**: Enter a unique name for your storage account.
   * **Region**: Select the region where you want to create the storage account.
   * **Performance**: Choose between Standard and Premium based on your needs.
   * **Replication**: Select the replication option (LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS) based on your redundancy requirements.
4. **Review and Create**: Click **Review + create** and then **Create**.

**Step 3: Configure Storage Account**

1. **Access Storage Account**: Once created, navigate to your storage account.
2. **Configure Settings**: You can configure various settings such as networking, access keys, and encryption.

**Step 4: Create and Manage Data**

1. **Blobs**: Store unstructured data like text and binary data.
2. **Files**: Create file shares accessible via SMB protocol.
3. **Queues**: Store and retrieve messages.
4. **Tables**: Store structured NoSQL data.

* **Working with the Blob service**

Azure Blob Storage is designed to store massive amounts of unstructured data. This can include documents, images, videos, backups, and logs. It’s highly scalable, durable, and accessible from anywhere in the world.

**Types of Blobs**

**Block Blob**

**Purpose**: Block blobs are designed for storing large amounts of text or binary data, such as documents, media files, or backups.

**Structure**: They are made up of blocks, each identified by a unique block ID. You can upload blocks individually and then commit them to form the final blob.

**Example**: Imagine you are uploading a large video file. You can break the video into smaller chunks (blocks), upload each chunk separately, and then commit all the chunks to create the final video file in Azure Storage.

**Append Blob**

**Purpose**: Append blobs are optimized for scenarios where data needs to be appended frequently, such as logging.

**Structure**: Similar to block blobs, append blobs are made up of blocks. However, you can only add (append) new blocks to the end of an append blob. You cannot modify or delete existing blocks.

**Example**: Consider a logging system for a web application. Each time an event occurs, a new log entry is appended to the end of the append blob. This way, you maintain a continuous log file without modifying previous entries.

**Page Blob**

**Purpose**: Page blobs are designed for random read/write operations and are used for scenarios like virtual hard drives.

**Structure**: Page blobs consist of 512-byte pages. You can read from or write to any page within the blob, making it suitable for applications that require frequent updates to specific parts of the data.

**Example**: Think of a virtual machine’s hard disk. The operating system and applications running on the VM need to read and write data to the disk frequently. Page blobs are used to store this virtual hard disk, allowing efficient random access to the data.

Here’s a step-by-step guide to help you get started:

**Step 1: Create an Azure Storage Account**

**Step 2: Create a Blob Container**

**Step 3: Upload Blobs to the Container**

**Step 4: Access Blobs Programmatically:**

**Step 5: Manage Blob Access**

**Step 6: Monitor and Optimize**

* **Azure Storage Accounts - Access tiers**

Azure Storage Accounts offer different access tiers to help you manage costs based on how frequently you access your data. Here are the main access tiers:

**1. Hot Tier**

* **Purpose**: For data that is accessed or modified frequently.
* **Cost**: Highest storage cost, but lowest access cost.
* **Use Cases**: Active data, frequently accessed files, and data staging for processing.

**2. Cool Tier: 30 Days**

* **Purpose**: For data that is infrequently accessed or modified. Stores data for 30 days
* **Cost**: Lower storage cost than the hot tier, but higher access cost.
* **Use Cases**: Short-term backups, disaster recovery, and older data that needs to be readily available.

**3. Cold Tier: 90 Days**

* **Purpose**: For data that is rarely accessed but still requires fast retrieval.
* **Cost**: Lower storage cost than the cool tier, but higher access cost.
* **Use Cases**: Long-term backups, archival data that needs quick access.

**4. Archive Tier: 180 Days**

* **Purpose**: For data that is rarely accessed and has flexible latency requirements.
* **Cost**: Lowest storage cost, but highest access cost.
* **Use Cases**: Long-term archival, compliance data, and data that can tolerate retrieval times in hours.

**Setting the Access Tier**

You can set the access tier at the time of blob upload or change it later. Here’s how you can set the access tier using the Azure portal:

1. **Navigate to your storage account** in the Azure portal.
2. **Select “Containers”** and choose the container where your blob is stored.
3. **Select the blob** you want to change.
4. **Click on “Change tier”** and select the desired access tier (Hot, Cool, or Archive).

* **Azure Storage Accounts - Data Redundancy**

Azure Storage Accounts offer several data redundancy options to ensure your data is protected against various types of failures, including hardware issues, network outages, and natural disasters. Here are the main redundancy options:

**1. Locally Redundant Storage (LRS): 3 copies**

* **Replication**: Data is replicated three times within a single datacenter in the primary region.
* **Use Case**: Suitable for scenarios where data loss can be tolerated and cost is a concern.
* **Cost**: Least expensive option.

**2. Zone-Redundant Storage (ZRS): 3 copies**

* **Replication**: Data is replicated synchronously across three Azure availability zones in the same region.
* **Use Case**: Ideal for applications requiring high availability and durability within a single region.
* **Cost**: More expensive than LRS but provides higher availability.

**3. Geo-Redundant Storage (GRS): 6 copies**

* **Replication**: Data is replicated three times within the primary region and asynchronously to a secondary region.
* **Use Case**: Suitable for disaster recovery scenarios where data needs to be available even if the primary region fails.
* **Cost**: Higher than ZRS due to cross-region replication.

**5. Geo-Zone-Redundant Storage (GZRS): 6 copies**

* **Replication**: Combines the benefits of ZRS and GRS by replicating data across availability zones in the primary region and asynchronously to a secondary region.
* **Use Case**: Best for mission-critical applications requiring high availability and disaster recovery.
* **Cost**: Higher than GRS due to the combination of zone and geo-redundancy.

**How to Set Redundancy:** You can set the redundancy option when creating a storage account or change it later through the Azure portal:

1. **Navigate to your storage account** in the Azure portal.
2. **Select “Configuration”** under the Settings section.
3. **Choose the desired redundancy option** from the “Replication” dropdown.
4. **Save the changes**.

* **Azure File Shares**

Working with Azure File Shares allows you to create and manage file shares in the cloud, which can be accessed from various operating systems. Azure File Share is like having a file server in the cloud. You can store files there and access them from anywhere, just like you would with a regular file server in your office.

**Key Features**

1. **Uses Common Protocols**: It uses SMB (a common file-sharing protocol) and NFS (another protocol used in Unix/Linux systems), so you can easily connect to it from Windows, Linux, or macOS computers.
2. **Access from Anywhere**: You can access your files from any location with an internet connection, making it easy to share files with team members in different places.
3. **Works with Other Azure Services**: You can use Azure File Share with other Azure services, like virtual machines, to store data that needs to be accessed by multiple services.

**Types of File Shares**

1. **Standard File Shares**: These are good for general use and are stored on regular hard drives. They offer different levels of redundancy to protect your data.
2. **Premium File Shares**: These are faster because they use SSDs (solid-state drives) and are great for applications that need high performance.

**When to Use It**

1. **Moving Applications to the Cloud**: If you have applications that use file shares, you can move them to Azure without changing much.
2. **Replacing File Servers**: You can replace your on-premises file servers with Azure File Shares for easier management and scalability.
3. **Development and Testing**: Share files easily between different environments without setting up complex infrastructure.
4. **Disaster Recovery**: Use snapshots and geo-redundant storage to protect your data and recover it if something goes wrong.

**How It Works**

1. **Creating a File Share**: You can create a file share using the Azure portal, Azure CLI, or PowerShell.
2. **Mounting the File Share**: You can mount the file share on your computer or server using the SMB protocol, just like mapping a network drive.
3. **Managing Access**: Control who can access the file share using Azure Active Directory (AAD) or shared access signatures (SAS).
4. **Syncing Files**: Use Azure File Sync to keep files in sync between your on-premises servers and Azure.

* **Azure Queue Storage (See working)**

**Azure Queue Storage** is a service for storing large numbers of messages that can be accessed from anywhere in the world. It’s designed for scenarios where you need to decouple different parts of your application and ensure reliable communication between them. Think of it as a mailbox where messages wait until they are picked up and processed.

Here are some key points:

* **Message Size**: You can store millions of messages, each up to 64 KB in size. Messages stay in the queue until they are read and processed.
* **Decoupling Components:** Helps different parts of your application work independently by sending messages to each other. Useful for handling varying loads by queuing messages.
* **FIFO(First in First out):** Ensures messages are processed in the order they were added.
* **Visibility Timeout:** When a message is read, it becomes invisible to others for a while, so only one part of your application processes it at a time.

**When to Use It**

1. **Task Scheduling**: Queue tasks for background processing, like image processing or data transformation.
2. **Load Leveling**: Smooth out traffic spikes by queuing requests and processing them steadily.
3. **Decoupling Microservices**: Allow different microservices to communicate without being directly connected, improving scalability and reliability.
4. **Asynchronous Processing**: Handle tasks that don’t need immediate processing, like sending emails or generating reports.

**How It Works**

1. **Creating a Queue**: You can create a queue using the Azure portal, Azure CLI, or PowerShell.
2. **Adding Messages**: Add messages to the queue using Azure tools or APIs.
3. **Reading Messages**: Read messages from the queue, process them, and then delete them once done.
4. **Managing Visibility Timeout**: Set a time period to ensure messages aren’t processed by multiple parts of your application at the same time.

* **Azure Table Storage**

Azure Table Storage is a service that allows you to store large amounts of structured data. It’s a NoSQL key-value store, which means it’s designed for high availability, scalability, and fast access to data.It provides a key/attribute store with a schemaless design, making it easy to adapt your data as your application evolves. Features:

1. **Schema-less Design**: You don’t need to define a schema for your data, making it flexible and easy to adapt as your data needs change.
2. **Key-Value Store**: Data is stored as key-value pairs, where each entity (row) has a unique key.
3. **Partitioning**: Data is automatically partitioned to improve scalability and performance. Each partition can be stored on different servers.
4. **High Availability**: Built on Azure’s robust infrastructure, ensuring your data is always available.
5. **Cost-Effective**: Pay only for what you use, making it a cost-effective solution for storing large amounts of data.

**Use Cases**

1. **IoT Data**: Store large volumes of sensor data from IoT devices.
2. **User Data**: Store user profiles, settings, and other related data.
3. **Application Logs**: Store logs from applications for monitoring and analysis.
4. **Metadata Storage**: Store metadata for other Azure services or applications.

**How It Works**

1. **Creating a Table**: You can create a table using the Azure portal, Azure CLI, or PowerShell.
2. **Adding Entities**: Add entities (rows) to the table using the Azure Storage SDKs or REST API. Each entity has a unique key composed of a partition key and a row key.
3. **Querying Data**: Query data using the OData protocol, which allows for flexible querying based on keys and properties.
4. **Managing Partitions**: Azure automatically manages partitions to optimize performance and scalability.

* **Azure Premium Storage accounts**

**Azure Premium Storage** is designed to support high-performance and low-latency applications. Azure Premium Storage is a type of storage in the cloud that’s super fast and reliable. It’s designed for applications that need to handle a lot of data quickly, like databases and big data applications. Here are some key features and steps to create and use Azure Premium Storage accounts:

**Key Features**

1. **High Performance**: It’s really fast and can handle a lot of data operations per second. Perfect for things like databases and big data.
2. SSD- Based Storage: Uses solid-state drives (SSDs), which are faster than traditional hard drives.

**When to Use It**

1. **Databases**: Great for SQL Server, Oracle, MySQL, and other databases that need high performance.
2. **Big Data and Analytics**: Ideal for applications that need to process large datasets quickly.
3. **Virtual Machines**: Provides fast storage for virtual machines, ensuring they run smoothly.
4. **Enterprise Applications**: Supports business applications that need consistent and reliable performance.

**How It Works**

1. **Creating a Premium Storage Account**: You can set up a premium storage account using Azure’s tools.
2. **Provisioning Managed Disks**: Use managed disks to make storage management easier. Azure takes care of the setup.
3. **Attaching Disks to VMs**: Attach these fast disks to your virtual machines to boost their performance.
4. **Monitoring Performance**: Use Azure Monitor to keep an eye on how well your storage is performing.

**Types of Premium Storage**:

* + [**Premium Block Blob Storage**: Optimized for workloads with high transaction rates or that use smaller objects1](https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-block-blob-premium).
  + [**Premium File Shares**: Suitable for enterprise or high-performance scale applications2](https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview).
  + [**Premium Page Blobs**: Ideal for I/O-intensive applications like databases](https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-block-blob-premium)[3](https://learn.microsoft.com/en-us/azure/virtual-machines/premium-storage-performance).
* **AzCopy tool**

AzCopy is a command-line tool provided by Microsoft that allows you to efficiently copy data to and from Azure Storage. It supports copying data between different storage accounts and even between on-premises storage and Azure.It’s particularly useful for copying blobs, files, and tables between storage accounts or from on-premises to Azure. Here’s a quick guide on how to use AzCopy:

**Key Features**

1. **Fast**: It’s designed to transfer data very quickly, even if you have a lot of it.
2. **Versatile**: Works with different types of Azure Storage, like Blob Storage, File Storage, and Table Storage.
3. **Cross-Platform**: You can use it on Windows, Linux, and macOS.
4. **Resumable Transfers**: If your transfer gets interrupted, it can pick up where it left off.
5. **Secure**: Supports secure ways to log in and transfer your data.

* **Azure Databox**

**Azure Data Box** is a physical device provided by Microsoft to help you securely transfer large amounts of data to and from Azure. It’s particularly useful when network bandwidth is limited or when transferring data over the network is not feasible. Here’s an overview of its features and how to get started:

**Key Features**

1. **Physical Data Transfer**: Azure Data Box provides physical devices that you can use to transfer large amounts of data to Azure. This is useful when transferring data over the internet would take too long or be too costly.
2. **Variety of Devices**:
   * **Data Box**: A rugged device with up to 100 TB of storage.
   * **Data Box Disk**: Smaller, portable disks with up to 8 TB of storage each, and you can use up to five disks at a time.
   * **Data Box Heavy**: A larger device with up to 1 PB (petabyte) of storage for very large data transfers.
3. **Secure Transfer**: Data is encrypted during transfer and can only be accessed by you. The devices are tamper-resistant and have secure handling procedures.
4. **Integration with Azure Services**: Once the data is transferred to Azure, it can be easily integrated with various Azure services like Azure Blob Storage, Azure Files, and Azure Data Lake Storage.
5. **Data Import and Export**: You can use Data Box to import data to Azure or export data from Azure to your on-premises environment.

**How It Works**

1. **Order a Data Box**: Request a Data Box device from the Azure portal.
2. **Receive and Load Data**: Receive the device, connect it to your network, and load your data onto it.
3. **Ship the Device Back**: Once the data is loaded, ship the device back to the Azure data center.
4. **Data Upload**: Azure uploads the data from the device to your specified Azure Storage account.
5. **Data Access**: Once the data is uploaded, you can access it through Azure services.

* **Azure Migrate**

**Azure Migrate** is a comprehensive service designed to help you plan and execute the migration of your on-premises workloads to Azure. Azure Migrate helps you move your existing data, applications, and servers to Azure. Think of it as a tool that guides you through the process of moving your stuff to the cloud.

**Key Features**

1. **Assessment Tools**:
   * **Server Assessment**: Checks if your servers are ready to move to Azure.
   * **Database Assessment**: Checks if your databases can be moved to Azure.
   * **Application Assessment**: Checks if your applications can run in Azure.
2. **Migration Tools**:
   * **Server Migration**: Helps you move your servers to Azure.
   * **Database Migration**: Helps you move your databases to Azure.
   * **Application Migration**: Helps you move your web applications to Azure.
3. **Centralized Hub**: A single place to manage everything related to your migration.
4. **Integration with Other Azure Services**: Works well with other Azure tools to make the migration process smooth.

**Example Scenarios**

**Azure Migrate:** Imagine you have a data center with servers and databases you want to move to Azure:

1. **Assessment**: Use Azure Migrate to check if your servers and databases are ready.
2. **Planning**: Plan how you will move everything based on the assessment.
3. **Migration**: Use Azure Migrate tools to move your servers and databases to Azure.

* **Azure Endpoint**

Azure Endpoint, or Azure Private Endpoint, allows you to connect to Azure services privately and securely using a private IP address within your virtual network (VNet).

**Key Features**

1. **Private Connectivity**: Connects your network to Azure services without using the public internet.
2. **Enhanced Security**: Uses private IP addresses to keep your data secure.
3. **Simplified Network Architecture**: Makes network setup easier by avoiding complex configurations.
4. **Integration with Azure Services**: Works with services like Azure Storage, Azure SQL Database, and Azure Cosmos DB.

**Use Cases**

1. **Secure Data Access**: Access Azure services securely from your network.
2. **Compliance Requirements**: Ensure your data stays private to meet regulatory requirements.
3. **Simplified Network Management**: Easier network setup compared to using VPNs or ExpressRoute.

**Example Scenario:**

**Azure Endpoint:** Suppose you have an application in Azure that needs to access an Azure SQL Database securely:

1. **Create a Private Endpoint**: Set up a private connection for your Azure SQL Database.
2. **Configure Network**: Make sure your network uses this private connection.
3. **Access Securely**: Access the database securely using the private IP address.

**Describe Azure architecture and services – Databases**

* **Azure SQL Database**

**Azure SQL Database** is a fully managed relational database service provided by Microsoft Azure. It offers a range of features designed to support modern cloud applications with high availability, scalability, and security. It’s like having a powerful database server, but you don’t have to worry about setting it up or maintaining it. Here’s an overview of its key features and how to get started:

**Key Features**

1. [**Managed Service**: **Azure SQL Database is a Platform-as-a-Service (PaaS)** that handles most database management tasks such as patching, backups, and monitoring without user intervention](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql)[1](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql).
2. [**High Availability**: Built-in high availability with a 99.99% uptime SLA](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql)[1](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql).
3. **Scalability**: Supports both vertical and horizontal scaling. [You can scale up or down based on your workload needs](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql)[2](https://azure.microsoft.com/en-us/products/azure-sql/database/).
4. [**Security**: Provides advanced security features like data encryption at rest and in transit, advanced threat protection, and vulnerability assessments](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql)[1](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql).
5. [**Performance**: Offers intelligent query processing and in-memory technologies to enhance performance](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql)[1](https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql).

**Pricing:** Azure SQL Database offers a flexible pricing model based on the chosen service tier and deployment option. You can choose between the DTU-based model (Database Transaction Units) and the vCore-based model (virtual cores), depending on your performance and cost requirements.

**Getting Started with Azure SQL Database**

**1. Create an Azure SQL Database**

* **Azure Portal**: Go to the Azure Portal.
* **Create a Resource**: Click on “Create a resource” > “Databases” > “SQL Database”.
* **Configure Database**: Fill in the required details such as subscription, resource group, database name, and server. Choose the appropriate pricing tier and performance level. Click “Review + create” and then “Create”.

**2. Configure the Database**

* **Server Configuration**: Set up the SQL server, including the server name, admin login, and password.
* **Networking**: Configure the networking settings to allow access to the database. You can set firewall rules to restrict access to specific IP addresses.

**3. Connect to the Database**

* **Connection String**: Obtain the connection string from the Azure Portal. This string is used to connect your application to the database.
* **SQL Server Management Studio (SSMS)**: Use SSMS or other database management tools to connect to your Azure SQL Database using the connection string.
* **Adding Data**

Adding data to an Azure SQL Database involves several steps, including creating tables, inserting data, and querying the database. Here’s a step-by-step guide to help you get started:

**Step-by-Step Procedure to Add Data to Azure SQL Database**

**1. Create a Table**

* **Connect to the Database**: Use SQL Server Management Studio (SSMS), Azure Data Studio, or the Azure Portal’s Query Editor to connect to your Azure SQL Database.
* **Create a Table with T-SQL**: Use the following T-SQL command to create a table:

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

HireDate DATE

);

**2. Insert Data into the Table**

**Insert Data with T-SQL**: Use the following T-SQL command to insert data into the table:

INSERT INTO Employees (EmployeeID, FirstName, LastName, HireDate)

VALUES (1, 'John', 'Doe', '2023-01-15'),

(2, 'Jane', 'Smith', '2023-02-20');

**3. Query the Data**

**Select Data with T-SQL**: Use the following T-SQL command to query the data:

SELECT \* FROM Employees;

**Tools and Methods for Adding Data**

**Using SQL Server Management Studio (SSMS)**

1. **Connect to Azure SQL Database**: Open SSMS and connect to your Azure SQL Database using the server name, database name, and authentication details.
2. **Execute T-SQL Commands**: Use the Query Editor to execute the T-SQL commands for creating tables, inserting data, and querying data.

**Using Azure Data Studio**

1. **Connect to Azure SQL Database**: Open Azure Data Studio and connect to your Azure SQL Database.
2. **Execute T-SQL Commands**: Use the Query Editor to run the T-SQL commands.

**Using Azure Portal’s Query Editor**

1. **Navigate to Your Database**: Go to the Azure Portal, navigate to your SQL Database, and open the Query Editor.
2. **Execute T-SQL Commands**: Use the Query Editor to run the T-SQL commands.

* **Azure Cosmos DB**

Azure Cosmos DB is a fully managed NoSQL database system provided by Microsoft Azure. It’s designed to be fast, reliable, and flexible for different types of applications. **How Azure Cosmos DB Works**

1. **Global Distribution**:
   * **Replication**: Azure Cosmos DB can replicate your data across multiple Azure regions. This ensures that your data is close to your users, providing low-latency access and high availability.
   * **Multi-Region Writes**: You can configure your database to allow writes in multiple regions, which helps in scenarios requiring high availability and disaster recovery.
2. **Data Models and APIs**:
   * **Multi-Model Support**: Azure Cosmos DB supports various data models, including document (JSON), key-value, graph, and column-family. This flexibility allows you to use the same database service for different types of applications.
   * **APIs**: It provides APIs for SQL, MongoDB, Cassandra, Gremlin, and Table storage, making it easy to integrate with existing applications.
3. **Consistency Levels**:
   * **Five Consistency Levels**: Azure Cosmos DB offers five consistency levels: Strong, Bounded Staleness, Session, Consistent Prefix, and Eventual. This allows you to choose the right balance between consistency and performance based on your application’s needs.
4. **High Availability**:
   * **SLA**: Azure Cosmos DB guarantees 99.999% availability for multi-region deployments and 99.99% for single-region deployments.
   * **Resilience**: The service is designed to handle regional failures, ensuring your data is always available.

**Example: E-commerce Application:**

You are building an e-commerce platform that needs to store product catalogs, user profiles, and order histories. The platform serves users globally, so low-latency access to data is crucial.

**Implementation**

1. **Global Distribution**: You configure Azure Cosmos DB to replicate data across multiple regions (e.g., North America, Europe, Asia) to ensure fast access for users worldwide.
2. **Data Models and APIs**:
   * **Product Catalog**: You use the document model (JSON) to store product details. Each product is a document with attributes like name, price, description, and inventory count.
   * **User Profiles**: User profiles are also stored as documents, containing information like username, email, address, and order history.
   * **Order History**: Each order is a document linked to a user profile, containing details like order date, items purchased, and total amount.
3. **Consistency Levels**: For critical data like order processing, you choose the **Strong** consistency level to ensure that all users see the most recent data. For less critical data like product reviews, you might choose the **Session** consistency level to balance performance and consistency.
4. **Performance and Scalability**: You set up autoscaling for throughput to handle traffic spikes during sales events, ensuring that the database can handle increased load without performance degradation.
5. **High Availability**: By enabling multi-region writes, you ensure that users can place orders even if one region experiences an outage. The data is automatically replicated to other regions, maintaining high availability.
   * + **Using SQL API**

Azure Cosmos DB’s SQL API is a way to store and manage data in the cloud using JSON documents. You can use SQL-like queries to work with this data, making it easy to use if you’re familiar with SQL

**Key Features**

1. **Stores Data as JSON**: Data is stored in JSON format, which is flexible and easy to read. Each piece of data (document) can have a different structure.
2. **SQL-like Queries**: You can use SQL commands to query your data, like SELECT, FROM, WHERE, and JOIN.
3. **Global Availability**: Your data can be copied to different parts of the world to make sure it’s always available and fast to access. If one region goes down, Cosmos DB can switch to another region automatically.
4. **Consistency Levels**: You can choose how consistent your data needs to be, balancing between speed and accuracy.
5. **Automatic Indexing**: Cosmos DB automatically indexes your data, so you don’t have to manage indexes yourself. You can customize indexing to improve query performance.

* **Enterprise Data Warehouse**

An Enterprise Data Warehouse (EDW) is a big storage system that collects and stores data from different parts of a company**. It helps in making better decisions by providing a complete view of the data.**

**Key Components of an EDW**

1. [**Data Sources**:](https://www.snowflake.com/guides/what-enterprise-data-warehouse)  These are the various systems and databases from which data is collected. They can include transactional databases, CRM systems, ERP systems, flat files, and external data sources.
2. **ETL Process (Extract, Transform, Load):** Extract: Taking data from different sources. Transform: Cleaning and organizing the data. Load: Putting the cleaned data into the data warehouse.
3. **Staging Area**: This is an intermediate space where data is aggregated, cleaned, and prepared before being loaded into the EDW. A temporary place where data is held and cleaned before moving to the main storage.
4. **Data Warehouse:** The main storage area where all the cleaned and organized data is kept. It’s designed for easy querying and analysis. The main part of the data warehouse where data is stored. It includes:

* **Fact Tables**: Store numerical data for analysis.
* **Dimension Tables**: Store descriptive data that gives context to the numbers.

1. **Data Marts:** Smaller sections of the data warehouse tailored for specific departments or purposes, like sales or marketing.
2. **Presentation Layer**: This provides an interface for users to access and interact with the data. [It includes tools for querying, reporting, and data visualization](https://www.snowflake.com/guides/what-enterprise-data-warehouse)

* **Creating an Azure Synapse workspace**

**Azure Synapse:** Azure Synapse Analytics is a service from Microsoft that combines big data and data warehousing. **It helps you collect, store, process, and analyze large amounts of data all in one place.**

1. **All-in-One Platform**: Combines tools for big data and data warehousing. Lets you use both on-demand (serverless) and dedicated resources.
2. **Data Lake Integration**: Works seamlessly with Azure Data Lake Storage to handle large volumes of data.
3. **SQL Analytics**: Allows you to use SQL queries to analyze data. Supports both serverless and dedicated SQL pools.
4. **Apache Spark**: Includes a managed Apache Spark environment for big data processing and machine learning.
5. **Data Integration**: Built-in tools to create data pipelines and move data around.

**How It Works**

1. **Data Ingestion**: You can bring in data from various sources like databases, cloud services, and real-time streams. Use Azure Data Factory to create pipelines that move data into Synapse.
2. **Data Storage**: Data is stored in Azure Data Lake Storage, which can handle both structured and unstructured data.
3. **Data Processing**: **SQL Pools**: Use SQL queries to process and analyze data. **Spark Pools**: Use Spark for big data processing and machine learning.
4. **Data Integration**: Combine data from different sources using Azure Data Factory. Clean and prepare data for analysis.
5. **Data Analysis**: Use SQL queries or Spark to analyze your data. Perform advanced analytics and machine learning.
6. **Data Visualization**: Create interactive reports and dashboards with Power BI. Use Synapse Studio to manage and visualize data.

* **Using pipelines to transfer data**

Using pipelines to transfer data in an Enterprise Data Warehouse (EDW) involves moving data from various sources into the data warehouse. This process is typically managed using ETL (Extract, Transform, Load) or ELT (Extract, Load, Transform) pipelines.

**Tools for Creating Pipelines**

1. **Azure Data Factory**:
   * A cloud-based data integration service that allows you to create, schedule, and orchestrate data pipelines.
   * Supports both ETL and ELT processes.
   * Provides a visual interface for designing pipelines.
2. **SSIS (SQL Server Integration Services)**:
   * A data integration tool that can be used to create ETL pipelines.
   * Supports complex data transformations and workflows.
3. **Apache NiFi**:
   * An open-source data integration tool that supports data flow automation.
   * Provides a web-based interface for designing data pipelines.

* **Azure Databricks**

**Azure Databricks** is a unified analytics platform that combines the best of data engineering, data science, and machine learning. It is built on Apache Spark and provides a collaborative environment for working with big data.

**Key Features of Azure Databricks**

* **All in One:** Combines tools for data engineering, data science, and data analytics. Allows teams to work together in shared workspaces.
* **Data Engineering**: Build and manage data pipelines for ETL (Extract, Transform, Load) processes.
* **Data Science**: Develop and train machine learning models using large datasets.
* **Machine Learning**: Deploy and manage machine learning models at scale.
* **Business Intelligence**: Create interactive dashboards and reports for data visualization and analysis.

**Getting Started**

1. **Create a Databricks Workspace**: Go to the Azure Portal. Create a new Databricks workspace.
2. **Launch the Workspace**: Open the Databricks interface from the Azure Portal.
3. **Create a Cluster**: Set up a cluster to run your data processing tasks.
4. **Create a Notebook**: Write and run code in a notebook to process and analyze data.
5. **Ingest and Process Data**: Bring in data from various sources and process it using Spark.
6. **Visualize Data**: Create reports and dashboards with Power BI or use built-in tools.

**Describe Azure architecture and services - Other services**

* **Azure Traffic Manager**

Azure Traffic Manager is a DNS-based traffic load balancer that helps you distribute traffic to your public-facing applications across global Azure regions. Here’s a detailed look at its features and functionalities:

**Key Features of Azure Traffic Manager**

1. **Traffic Routing Methods**:
   * **Priority**: Routes traffic to the primary endpoint unless it’s unavailable, then it fails over to secondary endpoints. Sends users to the main server first. If that server is down, it sends them to a backup.
   * **Weighted**: Distributes traffic based on assigned weights, allowing you to control the proportion of traffic sent to each endpoint. Distributes users based on assigned weights, like giving 70% of users to one server and 30% to another.
   * **Performance**: Directs traffic to the endpoint with the lowest latency, improving user experience by reducing response times. Sends users to the server that will respond the fastest.
   * **Geographic**: Routes traffic based on the geographic location of the user, ensuring compliance with data sovereignty regulations and improving performance. **Directs users to the nearest server based on their location.**
2. **High Availability**:

* Health Monitoring: Regularly checks if servers are working and redirects users if a server is down.
* Automatic Failover: Automatically sends users to a working server if one fails.

**How It Works:** Azure Traffic Manager operates at the DNS level, directing client requests to the appropriate service endpoint based on the configured routing method. [It does not act as a proxy or gateway, meaning it does not see the traffic passing between the client and the service](https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-how-it-works)[1](https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-how-it-works).

* **Azure content delivery network (Udemy Video)**

Azure Content Delivery Network (CDN) is a service that helps deliver web content quickly and efficiently to users around the world.

Imagine you have a website or an app with lots of images, videos, or other big files. When someone from another part of the world tries to access your site, it might take a while for those files to load because they have to travel a long distance. Azure CDN helps by storing copies of your files at various locations around the globe, called edge servers.

**Edge Servers**: Azure CDN stores copies of your content on edge servers located in various geographic locations. [When a user requests content, it’s delivered from the nearest edge server, reducing load times1](https://learn.microsoft.com/en-us/azure/cdn/cdn-overview)

[**Reduced Latency**: By serving content from locations closer to users, Azure CDN minimizes the time it takes for data to travel, resulting in faster load times](https://azure.microsoft.com/en-us/products/cdn/)

**How Does It Work?**

1. **User Request**: When someone visits your site, their request is sent to the nearest edge server instead of your main server.
2. **Fast Delivery**: The edge server quickly delivers the files because it’s closer to the user.
3. **Caching**: If the edge server doesn’t have the file, it fetches it from your main server, stores a copy, and then delivers it to the user. Next time, it can deliver the file even faster.

* **Azure Devops**

**A whiteboard with writing on it

Description automatically generated**

**A diagram of a project

Description automatically generated**

Azure DevOps is a comprehensive suite of tools and services designed to support the entire software development lifecycle. Azure DevOps is a set of tools from Microsoft that helps software development teams work together more effectively. It covers everything from planning and writing code to testing and deploying software.

**Key Components of Azure DevOps**

1. **Azure Boards**: Helps in planning, tracking, and discussing work across teams. **Features:**
   * **Kanban Boards**: Visual boards to track tasks.
   * **Backlogs**: Lists of tasks to be done.
   * **Dashboards**: Visual summaries of project progress.
2. **Azure Repos**:
   * **Purpose**: Provides version control for your code. Stores and manages your code. **Features:**
   * **Git Repositories**: Places to keep your code.
   * **Pull Requests**: Ways to review and approve code changes.
   * **Usage**: Developers can store their code, collaborate through pull requests, and manage different versions of their codebase. It allows multiple people to work on the same codebase.
3. **Azure Pipelines**:
   * **Purpose**: Automates the process of building, testing, and deploying code.
   * **Features**: Continuous Integration (CI) and Continuous Deployment (CD) for any language and platform. Automatically test and deploy your code.
   * **Usage**:  Ensure your code is always ready to be deployed.
4. **Azure Test Plans**:
   * **Purpose**: Provides tools for testing your software.
   * **Features**: Includes manual and automated testing to ensure your software works correctly.
   * **Usage**: Teams can create and run tests, track bugs, and ensure that the software meets quality standards.
5. **Azure Artifacts**:
   * **Purpose**: Manages and shares code packages.
   * **Features**: Store and share code libraries.
   * **Usage**: Manage dependencies in your projects.

**How It Works**

1. Planning and Tracking: Use Azure Boards to plan tasks and track progress.
2. Source Control: Store and manage your code in Azure Repos.
3. CI/CD Pipelines: Use Azure Pipelines to automate building, testing, and deploying your code.
4. Testing: Use Azure Test Plans to run tests and ensure your software is working properly.
5. Package Management: Share and manage code packages with Azure Artifacts**.**

* **Machine Learning in Azure**

Azure Machine Learning is a platform that supports the entire machine learning lifecycle. It helps data scientists, machine learning engineers, and developers to create, train, deploy, and manage machine learning models efficiently.

**Key Features**

1. **Model Training**:
   * [**Automated Machine Learning (AutoML)**: Automatically selects the best algorithms and hyperparameters for your data, making it easier to build high-quality models](https://www.bing.com/aclick?ld=e8qJ8HdXy8kA_fo0d-YrkzYjVUCUzmGYAe4Ki5WMcQpSwV14lrm2RQdzFHS-lKGqkxauX8pygmj97YfXlGDJCh2cY7oNpuQa3ipt-osArGj_duWY4YJqA7SMEf21RtNsBZWTZ-LnIRPkfqD8U9tnlhs2giTtSc_oPtPuWPa3qjsl_9pc9O&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmbWFjaGluZS1sZWFybmluZyUyZnNlYXJjaCUyZiUzZmVmX2lkJTNkX2tfN2Y2YTdlOWE2NjM2MTBkNmMyZWUxODViYTEyNzM3M2Nfa18lMjZPQ0lEJTNkQUlEY21tZjFlbGo5djVfU0VNX19rXzdmNmE3ZTlhNjYzNjEwZDZjMmVlMTg1YmExMjczNzNjX2tfJTI2bXNjbGtpZCUzZDdmNmE3ZTlhNjYzNjEwZDZjMmVlMTg1YmExMjczNzNj&rlid=7f6a7e9a663610d6c2ee185ba127373c)
   * [**Custom Training**: Supports popular frameworks like TensorFlow, PyTorch, and scikit-learn, allowing you to train models using your preferred tools2](https://www.datacamp.com/tutorial/azure-machine-learning-guide).
2. **Model Deployment**:
   * **Real-time Inference**: Deploy models as web services that can handle real-time predictions.
   * **Batch Inference**:You can run predictions on large sets of data all at once, which is useful for big datasets.
3. **MLOps (Machine Learning Operations)**:
   * **Model Management**: Track and manage different versions of your models.
   * [**Monitoring and Retraining**:](https://www.bing.com/aclick?ld=e8qJ8HdXy8kA_fo0d-YrkzYjVUCUzmGYAe4Ki5WMcQpSwV14lrm2RQdzFHS-lKGqkxauX8pygmj97YfXlGDJCh2cY7oNpuQa3ipt-osArGj_duWY4YJqA7SMEf21RtNsBZWTZ-LnIRPkfqD8U9tnlhs2giTtSc_oPtPuWPa3qjsl_9pc9O&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmbWFjaGluZS1sZWFybmluZyUyZnNlYXJjaCUyZiUzZmVmX2lkJTNkX2tfN2Y2YTdlOWE2NjM2MTBkNmMyZWUxODViYTEyNzM3M2Nfa18lMjZPQ0lEJTNkQUlEY21tZjFlbGo5djVfU0VNX19rXzdmNmE3ZTlhNjYzNjEwZDZjMmVlMTg1YmExMjczNzNjX2tfJTI2bXNjbGtpZCUzZDdmNmE3ZTlhNjYzNjEwZDZjMmVlMTg1YmExMjczNzNj&rlid=7f6a7e9a663610d6c2ee185ba127373c)  Watch how your models perform and set up automatic retraining to keep them up-to-date.
4. **Collaboration and Productivity**:
   * **Notebooks and IDEs**: Use Jupyter notebooks and integrated development environments (IDEs) like Visual Studio Code for development.
   * [**Pipelines**: Create and manage machine learning workflows, automating steps like data preparation, training, and deployment2](https://www.datacamp.com/tutorial/azure-machine-learning-guide).

**How It Works**

1. **Data Preparation**: Import and clean your data using Azure Machine Learning’s data preparation tools.
2. **Model Training**: Use AutoML or custom training to build your model. You can run experiments to find the best model.
3. **Model Evaluation**: Evaluate the performance of your model using built-in metrics and visualizations.
4. **Model Deployment**: Deploy your model as a web service or for batch processing.
5. **Model Management**: Monitor your model’s performance and set up automated retraining if needed.

* **Azure AI Services**

Azure AI Services offer a comprehensive suite of tools and APIs to help you build intelligent applications. Here are some of the key services available:

**Key Azure AI Services**

1. **Azure OpenAI Service**: Leverage powerful language models for tasks like text generation, summarization, and translation.
2. **Azure AI Vision**: Analyze images and videos to detect objects, faces, and text using optical character recognition (OCR).
3. **Azure AI Speech**: Convert speech to text, text to speech, and translate spoken language in real-time.
4. **Azure AI Language**: Build conversational interfaces, analyze sentiment, and extract key phrases from text.
5. **Azure AI Search**: Implement AI-powered search capabilities in your applications to retrieve the most relevant data.
6. **Azure AI Content Safety**: Monitor and filter out offensive or inappropriate content in text and images.
7. **Azure AI Translator**: Translate text and documents across more than 100 languages.

* **Azure Key Vault**

Azure Key Vault is a cloud service designed to securely store and manage sensitive information such as cryptographic keys, secrets, and certificates. Here’s a detailed overview:

**Key Features**

1. **Secrets Management**:
   * **Secure Storage**: You can store things like passwords and API keys safely.
   * **Access Control**: You decide who can access and manage these secrets.
2. **Key Management**:
   * **Encryption Keys**: Create and manage keys that encrypt your data.
   * **Key Rotation**: Automatically change keys regularly to keep them secure.
3. **Certificate Management**:
   * **Provisioning and Management**: Easily get and manage SSL/TLS certificates for secure websites.
   * **Automated Renewal**: Automatically renew certificates so they don’t expire.
4. **Integration and Compatibility**:
   * **Works with Azure Services**: Integrates smoothly with other Azure services.
   * **APIs and SDKs**: Provides tools to easily connect with your applications.

**How It Works**

1. **Centralized Storage**: Store all your secrets, keys, and certificates in one secure place.
2. **Access Control**: Use Azure Active Directory to control who can access and manage the stored information.
3. **Secure Access**: Applications can access the stored secrets and keys securely, without needing to store them in the code.
4. **Monitoring and Logging**: Keep track of who accesses and uses your secrets, keys, and certificates.

* **Azure Logic Apps (same as Azure functions)**

Azure Logic Apps is a cloud-based service that helps you create and automate workflows. Here’s a detailed overview:

**Key Features**

1. **Visual Designer**:
   * **Drag-and-Drop Interface**: You can create workflows by dragging and dropping elements, so you don’t need to write code.
   * **Prebuilt Connectors**: There are many ready-made connectors to link services like Office 365, SQL Server, and Salesforce.
2. **Triggers and Actions**:
   * **Triggers**: These are events that start your workflow, like receiving an email or a new file being uploaded.
   * **Actions**: These are the steps your workflow takes after the trigger, like sending an email or updating a database.
3. **Control Flow**:
   * **Conditions and Loops**: You can add conditions to control what happens next and use loops to repeat actions until a condition is met.
   * **Error Handling**: You can set up ways to handle errors to make sure your workflows run smoothly.

**How It Works**

1. **Create a Logic App**: Start by creating a logic app in the Azure portal.
2. **Design the Workflow**: Use the visual designer to add triggers and actions to your workflow.
3. **Configure Connectors**: Set up connections to the services and systems you want to integrate.
4. **Run and Monitor**: Run your workflow and monitor its performance using built-in tools.

**Example: Automating Email Notifications**

Let’s walk through an example where we create a logic app to send an email notification whenever a new file is uploaded to an Azure Blob Storage container.

**Step-by-Step Guide**

1. **Create a Logic App**
2. **Define the Trigger**: In the Logic Apps Designer, choose the trigger “When a blob is added or modified”. Configure the trigger to monitor a specific container in your Azure Blob Storage account.
3. **Add an Action**:
   * After the trigger, add an action to send an email.
   * Choose the “Office 365 Outlook” connector and select the “Send an email” action.
   * Configure the email details (recipient, subject, body).
4. **Configure Connectors**: Authenticate and configure the Office 365 Outlook connector to use your email account.
5. **Deploy and Monitor**: Save and deploy your logic app. Test the workflow by uploading a file to the specified Blob Storage container. Monitor the execution in the Azure portal to ensure the email is sent.

* **Azure IOT Devices**

Azure IoT Hub is a cloud service that helps you manage, monitor, and communicate with Internet of Things (IoT) devices. Azure IoT Hub acts as a central hub for connecting, managing, and securing IoT devices. [It allows you to connect millions of devices and their backend solutions reliably and securel](https://www.bing.com/aclick?ld=e8l4iJPdC_mH8ZL6WAQCF9QTVUCUxV_6hEFAadOzJfeU_EUy30VGrb1Rfn6bJiT0UPjkGHxru9e8BFx63aRLfHDCHyAw24amUXWvQWwnm20uW4FYQ8bMZbkJxxYSyCd0-Xd50K0Bp1Zvo1GrWB-8LjR0NzfM5vqB_zLDA9l4O2wXJVsqSz&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa19lNmEyZWZhYjFmZmUxZTJmNTY0Mzg5NmFkNDM3Zjk5MF9rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfZTZhMmVmYWIxZmZlMWUyZjU2NDM4OTZhZDQzN2Y5OTBfa18lMjZtc2Nsa2lkJTNkZTZhMmVmYWIxZmZlMWUyZjU2NDM4OTZhZDQzN2Y5OTA&rlid=e6a2efab1ffe1e2f5643896ad437f990)y

**Key Features**

1. **Device Communication**:
   * **Device-to-Cloud Messaging**: Devices send data (like temperature readings) to the cloud.
   * **Cloud-to-Device Messaging**: The cloud sends commands (like turning on a light) to the devices.
   * **File Upload**: Devices can upload files (like logs or images) to the cloud.
2. **Device Management**:
   * **Device Twins**: Digital copies of your devices that store their state and settings.
   * **Direct Methods**: You can send direct commands to devices from the cloud.
   * **Device Provisioning**: Automatically register and set up devices in large numbers.
3. **Security**:
   * **Identity Registry**: Keeps track of devices and their security credentials.
   * **Authentication**: Ensures secure communication using tokens or certificates.
   * **Access Control**: Manage who can access and control the devices.

**How It Works**

1. **Connect Devices**: Devices connect to the IoT Hub using internet protocols like MQTT or HTTPS.
2. **Send and Receive Messages**: Devices send data to the cloud and receive commands from the cloud.
3. **Manage Devices**: Use digital copies (device twins) and direct commands to monitor and control devices.
4. **Secure Communication**: Ensure all communication is secure using authentication and access controls.

* **Azure Batch Service**

Azure Batch is a cloud computing service that enables you to run large-scale parallel and high-performance computing (HPC) applications efficiently in the cloud.

**Key Features**

1. **Job Scheduling**:
   * **Automatic Scheduling**: Azure Batch automatically decides when and where to run your tasks. You don’t need to worry about managing this.
   * **Task Management**: You can break your job into smaller tasks, and Azure Batch will distribute these tasks across the virtual computers.
2. **Compute Management**:
   * **Pool of Compute Nodes**: You create a group of virtual computers (called a pool) that will run your tasks. You can choose how many computers you need and what kind they should be.
   * **Scaling**: Azure Batch can automatically add or remove computers based on how much work there is, so you only use what you need.
3. **Data Integration**:
   * **Azure Storage Integration**: You can easily upload your data to Azure Storage, and your tasks can access this data directly.
   * **Application Packages**: You can upload the software your tasks need to run, so everything is ready on the virtual computers.
4. **Monitoring and Logging**:
   * **Job Monitoring**: You can keep an eye on how your jobs and tasks are doing using the Azure portal or other tools.
   * **Logging**: Azure Batch collects logs and diagnostic data to help you troubleshoot and improve your workflows.

**Describe Azure architecture and services - Identity and Access**

* **Microsoft Entra ID**

Microsoft Entra ID, formerly known as Azure Active Directory (Azure AD), is a comprehensive cloud-based identity and access management service. Microsoft Entra ID is a service that allows employees to access external resources like Microsoft 365, the Azure portal, and thousands of other SaaS applications

**Key Features of Microsoft Entra ID**

* **Identity Management:**
* **User Accounts:** Create and manage user accounts to ensure only authorized users can access your resources.
* **Groups**: Organize users into groups to make it easier to manage access for multiple people at once.
* **Access Management:**
* **Single Sign-On (SSO):** Users can sign in once and access multiple applications without needing to sign in again.
* **Conditional Access:** Set rules that control access based on conditions like where the user is located or what device they’re using.
* **Security:**
* **Multi-Factor Authentication (MFA):** Adds an extra layer of security by requiring users to verify their identity with a second method, like a phone call or text message.
* **Identity Protection:** Detects and responds to suspicious activities to protect user accounts.
* **Use in Azure:** Microsoft Entra ID is deeply integrated with Azure, providing identity and access management for Azure resources. Here’s how it’s used:

1. **Azure Portal Access**: Controls who can access the Azure portal and manage resources.
2. **Resource Access Management**: Manages access to Azure services like Virtual Machines, Storage Accounts, and Databases.
3. **Role-Based Access Control (RBAC)**: Assigns roles to users and groups to control their access to Azure resources.

* **Role Based Access Control**

Azure RBAC is an authorization system built on Azure Resource Manager that provides access management to Azure resources. [It allows you to segregate duties within your team and grant only the amount of access to users that they need to perform their jobs](https://learn.microsoft.com/en-us/azure/role-based-access-control/overview). Here’s a detailed explanation:

**Key Concepts of Azure RBAC**

1. **Security Principal:** A security principal is an object that represents a user, group, service principal, or managed identity that is requesting access to Azure resources. **Types**: Users (people with Azure accounts), groups (collections of users), service principals (applications), and managed identities (identities managed by Azure).
2. **Role Definition**: A role is a collection of permissions. It’s typically just called a role. A role lists the actions that can be performed, such as read, write, and delete. Azure has many built-in roles, but you can also create custom roles if needed.
3. **Scope:** Scope is the level at which permissions apply. It can be a management group, subscription, resource group, or individual resource. You can give someone permissions to manage all resources in a specific resource group.

**How It Works**

1. **Assign Roles**: Decide what roles your team members need and assign them.
2. **Define Permissions**: Use built-in roles or create custom roles to specify what each role can do.
3. **Set Scope**: Choose where the role applies, like a specific resource group or subscription.
4. **Enforce Access**: Azure RBAC makes sure users can only do what their roles allow.

* **Multi-Factor Authentication**

Multi-Factor Authentication (MFA) in Azure adds an extra layer of security to your Azure account by requiring more than one method of verification. This helps protect against unauthorized access to your account. Here’s a detailed explanation and step-by-step procedure to set up MFA in Azure:

**Key Concepts of Azure Multi-Factor Authentication**

1. **Verification Methods**
   * **Phone Call**: Receive a call and press the # key to authenticate.
   * **Text Message**: Receive a text message with a verification code.
   * **Mobile App Notification**: Receive a push notification on your mobile app and approve it.
   * **Mobile App Verification Code**: Use a verification code generated by the mobile app.
2. **Conditional Access**: Conditional Access policies can be used to enforce MFA based on specific conditions, such as user location, device state, or application being accessed.

* **Conditional Access Policies**

Conditional Access policies in Azure are a way to control access to your applications and data based on specific conditions. They help ensure that only the right people, using the right devices, under the right conditions, can access your resources. Here’s a detailed explanation:

**Key Components**

1. **Conditions**:
   * **User or Group Membership**: Decide which users or groups the policy applies to.
   * **Sign-in Risk**: Check how risky a sign-in attempt is based on things like location and behavior.
   * **Device State**: Make sure the device meets your security standards.
   * **Location**: Restrict access based on where the user is signing in from.
   * **Client Apps**: Apply policies to specific apps, like web apps or mobile apps.
2. **Controls**:
   * **Grant Controls**: Set actions required for access, like needing multi-factor authentication (MFA) or using a secure device.
   * **Session Controls**: Manage user sessions, like limiting how long a session lasts or requiring re-authentication.

**How It Works**

1. **Define Conditions**: Set up the conditions for the policy. For example, you might require MFA if someone is signing in from a new location.
2. **Set Controls**: Decide what actions to take if the conditions are met, like blocking access or requiring MFA.
3. **Apply Policies**: The policies are enforced whenever someone tries to access your resources. If the conditions are met, the specified actions are taken.

* **Microsoft Entra ID: Inviting external identities**

Inviting external identities in Microsoft Entra ID allows you to securely collaborate with users outside your organization. Microsoft Entra External ID is a feature that lets you invite external users to access your apps and resources. [These users can use their own credentials, such as work, school, or social accounts](https://learn.microsoft.com/en-us/entra/external-id/external-identities-overview) Here’s a detailed explanation and step-by-step procedure:

**Key Concepts of External Identities**

1. **External Users (Guests)**: External users, also known as guests, can be partners, vendors, or other collaborators who need access to your resources.
2. **Identity Providers:** External users can use various identity providers to authenticate, such as Microsoft Entra accounts, Microsoft accounts, or social identities like Google.

**How to Invite External Users**

1. **Sign In**: Log in to the Microsoft Entra admin center as a User Administrator.
2. **Navigate to Users**: Go to Identity > Users > All users.
3. [**Invite External User**: Select “Invite external user” from the menu2](https://learn.microsoft.com/en-us/entra/external-id/b2b-quickstart-add-guest-users-portal).
4. **Enter Details**: Provide the email address of the guest user, their display name, and an optional invitation message.
5. **Send Invitation**: Review the details and send the invitation. [The user will receive an email with a link to accept the invitation2](https://learn.microsoft.com/en-us/entra/external-id/b2b-quickstart-add-guest-users-portal).

* **Microsoft Entra Id domain services**

Microsoft Entra Domain Services is a cloud service that provides the same kinds of domain services you get with traditional on-premises Active Directory (AD), but without the need to manage the servers yourself. It’s like having a managed AD in the cloud.

**Key Features**

1. **Managed Domain Services**:
   * **Domain Join**: You can join virtual machines and other resources to the domain, just like with on-premises AD.
   * **Group Policy**: Apply settings and configurations to users and computers in the domain.
2. **Integration with Microsoft Entra ID**:
   * **Synchronization**: Syncs user accounts, groups, and passwords from Microsoft Entra ID to the managed domain.
   * **Hybrid Environments**: Works with both cloud-only and hybrid setups, integrating with on-premises AD through Microsoft Entra Connect.
3. **Simplified Management**:
   * **No Domain Controller Management**: Microsoft handles the domain controllers, so you don’t need to worry about updates, patches, or backups.
   * **High Availability**: Ensures your domain services are always available with multiple domain controllers.
4. **Security**:
   * **Encryption**: Uses encryption to protect data.
   * **Access Control**: Uses Microsoft Entra ID features like multi-factor authentication (MFA) and conditional access to secure your domain.

**How It Works**

1. **Create a Managed Domain**: You start by creating a managed domain in the Microsoft Entra admin center.
2. **Deployment**: Azure sets up two domain controllers in your chosen region.
3. **Synchronization**: User accounts, groups, and passwords are synced from Microsoft Entra ID to the managed domain.
4. **Usage**: Your applications, services, and virtual machines can now use the managed domain for things like domain join, group policy, LDAP, and secure authentication.

* **Microsoft Entra Cloud Sync**

Microsoft Entra Cloud Sync helps you keep your users, groups, and contacts in sync between your on-premises Active Directory (AD) and Microsoft Entra ID (formerly Azure Active Directory).

**Key Points**

1. **Easy Setup**: Uses lightweight agents instead of a big, complex application. These agents connect your AD to Microsoft Entra ID.
2. **Works with Multiple ADs**: Great for companies with multiple ADs, like after mergers. Can sync from several ADs to one Microsoft Entra ID.
3. **Reliable**: You can use multiple agents to ensure it keeps working even if one fails.
4. **Managed in the Cloud**: Configuration and management are done in the cloud, reducing the need for on-site servers. Agents update automatically, so less work for your IT team.
5. **Frequent Updates**: Syncs every 2 minutes, so changes in AD show up quickly in Microsoft Entra ID.
6. **Handles Large Groups**: Can manage groups with up to 50,000 members.

* **Microsoft Defender for Cloud**

Microsoft Defender for Cloud is a security tool that helps protect your cloud-based applications and infrastructure. It works with different cloud services like Azure, AWS, and Google Cloud, as well as on-premises environments. Here’s a detailed overview:

**Key Features of Microsoft Defender for Cloud**

1. **Protects Your Applications**:
   * **DevSecOps Integration**: Ensures security is part of the development process from the start.
   * **Cloud Security Posture Management (CSPM)**: Finds and fixes security risks in your cloud setup.
   * **Cloud Workload Protection Platform (CWPP)**: Protects specific parts of your cloud, like servers and databases.
2. **Unified Security Management**:
   * **Multi-Cloud and Hybrid Support**: Works across different cloud services and on-premises setups.
   * **Centralized Dashboard**: Provides a single place to manage and monitor all your cloud security.
3. **Advanced Threat Protection**:
   * **Threat Detection and Response**: Uses smart technology to find and respond to security threats.
   * **Integration with Microsoft Defender XDR**: Combines alerts from different sources for a complete security view.

* **Microsoft Sentinel**

Microsoft Sentinel is a cloud-based tool that helps keep your computer systems and data safe. Think of it as a security guard for your digital world. It watches over your systems, looks for any signs of trouble, and helps you respond quickly if something goes wrong.

**Key Features**

* **Security Monitoring:**

**Watches for Threats**: Continuously monitors your systems for any suspicious activity.

**Alerts:** Sends alerts if it detects something unusual, like a potential cyber attack.

* **Data Collection:**

**Gathers Information:** Collects data from various sources like servers, applications, and devices.

**Centralized Storage**: Stores all this data in one place for easy analysis.

* **Analysis and Detection:**

**Smart Analysis:** Uses artificial intelligence (AI) to analyze the collected data and identify potential threats.

**Incident Detection**: Helps detect incidents like unauthorized access or malware.

* **Response and Automation**:

**Automated Actions**: Can automatically take actions to mitigate threats, such as blocking a suspicious IP address.

**Playbooks:** Provides predefined response plans to handle different types of security incidents.

**Example**

Imagine you run a company with several servers and applications. You want to make sure your data is safe from hackers. Here’s how Microsoft Sentinel can help:

1. **Set Up**: You set up Microsoft Sentinel to monitor your servers and applications.
2. **Data Collection**: It starts collecting data from all your systems.
3. **Threat Detection**: One day, Sentinel detects unusual login attempts from a foreign country.
4. **Alert**: It sends you an alert about the suspicious activity.
5. **Response**: You can quickly investigate the alert and, if necessary, block the suspicious IP address to prevent unauthorized access.

* **Azure DDOS**

**Azure DDoS Protection** is a service designed to protect your applications from Distributed Denial of Service (DDoS) attacks, which aim to overwhelm your resources and make your applications unavailable to legitimate users. Here’s a detailed overview:

**Key Points**

1. **Always Watching**: Monitors your application traffic all the time to detect and stop DDoS attacks automatically.
2. **Smart Adjustments**: Learns your application’s normal traffic patterns and adjusts protection measures as needed.
3. **Detailed Reports**: Provides analytics and alerts to help you understand and respond to DDoS attacks.
4. **Multi-Layer Protection**: Protects both the network and application layers when used with a Web Application Firewall (WAF).
5. **Cost Protection**: Helps cover the costs of scaling up resources during a DDoS attack.

**How It Works**

1. **Setup**: Enable DDoS Protection on your Azure virtual network. No changes to your applications are needed.
2. **Monitoring and Detection**: Continuously monitors traffic and uses machine learning to detect unusual patterns.
3. **Mitigation**: Automatically filters out malicious traffic while allowing legitimate traffic through.

* **Azure Firewall**

Azure Firewall is a managed, cloud-based network security service that protects your Azure Virtual Network resources. **Azure Firewall** is a security service that protects your Azure Virtual Network resources. It provides both east-west and north-south traffic inspection and is designed to offer high availability and unrestricted cloud scalability. Here’s a detailed overview:

1. **Stateful Firewall**: Inspects and filters traffic based on the state of the connection.
2. **High Availability**: Built-in high availability and can span multiple Availability Zones for reliability.
3. **Scalable**: Automatically scales to meet your network traffic needs.
4. **Threat Intelligence**: Blocks traffic from known malicious IP addresses and domains using Microsoft threat intelligence.
5. **Custom Rules**: Allows you to create rules to control both inbound and outbound traffic.
6. **Logging and Analytics**: Provides detailed logs and analytics to help you monitor and manage your network security.

**How It Works**

1. **Deployment**: Deploy Azure Firewall in your virtual network and manage it centrally.
2. **Configuration**: Set up rules to control traffic flow based on your security needs.
3. **Monitoring**: Use Azure Monitor to track firewall activity and get insights into your network traffic.

**Describe Azure management and governance**

* **Pricing Calculator**

The Azure Pricing Calculator is a web-based tool that helps you estimate the costs of using various Azure services.

**Key Points**

1. **Easy to Use**: The calculator has a user-friendly interface where you can pick and configure the Azure services you need.
2. **Service Selection**: You can choose from various Azure services like Virtual Machines, Storage, and Databases. Each service has its own settings you can adjust.
3. **Cost Estimation**: After selecting and configuring the services, the calculator shows you a detailed cost estimate, including the cost of each service and the total cost.

**How to Use the Azure Pricing Calculator**

1. **Access the Calculator**: Go to the Azure Pricing Calculator website.
2. **Select Services**: Use the product picker to find and select the Azure services you need.
3. **Configure Services**: Adjust the settings for each service, like region and usage. The calculator updates the cost estimate based on your settings.
4. **Review Estimates**: The calculator shows a detailed estimate of the costs. You can change the settings to see how it affects the cost.
5. **Save and Share**: Save your estimate, export it to Excel, or share it with others using a link.

* **Total Cost of Ownership**

**The Azure Total Cost of Ownership (TCO) Calculator is a tool designed to help you estimate the cost savings you can achieve by migrating your on-premises workloads to Microsoft Azure**. The Azure TCO Calculator helps you figure out how much money you can save by moving your on-premises workloads (like servers and storage) to Microsoft Azure.

**Key Points**

1. **Compare Costs**: You enter details about your current setup, like the number of servers, storage, and networking costs. The calculator compares these costs with what it would cost to run the same workloads on Azure.
2. **Detailed Reports**: It generates reports showing potential savings and benefits of moving to Azure.

**How It Works**

1. **Input Current Infrastructure**: Enter details about your existing setup, including hardware costs, maintenance expenses, and electricity bills.
2. **Azure Cost Comparison**: The calculator shows a breakdown of costs if you move to Azure, considering things like discounts and savings on maintenance and power.
3. **Generate Reports**: It creates a report comparing your current costs with the estimated costs on Azure, highlighting potential savings.

**Advanced Features**

1. **Cost Management Integration**: Works with Azure Cost Management to help you track and manage your actual spending.
2. **Discounts and Offers**: If you have special pricing or discounts, you can log in to see estimates with your specific prices.

* **Budgets in Azure**

Creating budgets in Azure helps you manage and control your spending on Azure services. Here’s a detailed explanation:

**Steps to Create a Budget**

1. **Access Cost Management**: [Go to the Azure portal and select **Cost Management + Billing** from the left-hand menu](https://www.bing.com/aclick?ld=e8ffFW6oVnwaz9UWs7494eyDVUCUyhi0p9YFUMDPPsDiKwXZazTjUTkBG_3Wvz6f5Fuu77Q9IRk6KbL9guu-D7H8pqrgbuEIZSdPtD5wVPfZ38dHLxOoQyU4qQTHyPkQgjTjeJBremUf7O3TtKotZ_nxz_SF1hsxoawaMawPYGGtmO9ggT&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa184Mzc3ODIwZjM1ZDAxZGIwMGY3ZGU0Y2VkNTI0NDlmMl9rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjJfa18lMjZtc2Nsa2lkJTNkODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjI&rlid=8377820f35d01db00f7de4ced52449f2)[1](https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/tutorial-acm-create-budgets).
2. **Navigate to Budgets**: [In the Cost Management section, click on **Budgets**](https://www.bing.com/aclick?ld=e8ffFW6oVnwaz9UWs7494eyDVUCUyhi0p9YFUMDPPsDiKwXZazTjUTkBG_3Wvz6f5Fuu77Q9IRk6KbL9guu-D7H8pqrgbuEIZSdPtD5wVPfZ38dHLxOoQyU4qQTHyPkQgjTjeJBremUf7O3TtKotZ_nxz_SF1hsxoawaMawPYGGtmO9ggT&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa184Mzc3ODIwZjM1ZDAxZGIwMGY3ZGU0Y2VkNTI0NDlmMl9rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjJfa18lMjZtc2Nsa2lkJTNkODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjI&rlid=8377820f35d01db00f7de4ced52449f2)[1](https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/tutorial-acm-create-budgets).
3. **Create a New Budget**: [Click on the **+ Add** button to create a new budget2](https://www.codingdrills.com/tutorial/azure-tutorial/azure-budgets).
4. **Set Budget Details**:
   * **Name**: Give your budget a name.
   * **Scope**: Choose the subscription or resource group for the budget.
   * **Amount**: Specify the budget amount and currency.
   * [**Timeframe**: Set the timeframe for your budget (e.g., monthly, quarterly, annually)](https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/tutorial-acm-create-budgets)[2](https://www.codingdrills.com/tutorial/azure-tutorial/azure-budgets).
5. **Configure Alerts**: [Set up alerts to notify you when spending reaches certain thresholds (e.g., 50%, 75%, 90% of the budget)](https://www.bing.com/aclick?ld=e8ffFW6oVnwaz9UWs7494eyDVUCUyhi0p9YFUMDPPsDiKwXZazTjUTkBG_3Wvz6f5Fuu77Q9IRk6KbL9guu-D7H8pqrgbuEIZSdPtD5wVPfZ38dHLxOoQyU4qQTHyPkQgjTjeJBremUf7O3TtKotZ_nxz_SF1hsxoawaMawPYGGtmO9ggT&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa184Mzc3ODIwZjM1ZDAxZGIwMGY3ZGU0Y2VkNTI0NDlmMl9rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjJfa18lMjZtc2Nsa2lkJTNkODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjI&rlid=8377820f35d01db00f7de4ced52449f2)[1](https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/tutorial-acm-create-budgets).
6. **Save the Budget**: Review your settings and save the budget. [Azure will now monitor your spending and send alerts based on your configurations](https://www.bing.com/aclick?ld=e8ffFW6oVnwaz9UWs7494eyDVUCUyhi0p9YFUMDPPsDiKwXZazTjUTkBG_3Wvz6f5Fuu77Q9IRk6KbL9guu-D7H8pqrgbuEIZSdPtD5wVPfZ38dHLxOoQyU4qQTHyPkQgjTjeJBremUf7O3TtKotZ_nxz_SF1hsxoawaMawPYGGtmO9ggT&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZmcmVlJTJmc2VhcmNoJTJmJTNmZWZfaWQlM2Rfa184Mzc3ODIwZjM1ZDAxZGIwMGY3ZGU0Y2VkNTI0NDlmMl9rXyUyNk9DSUQlM2RBSURjbW1mMWVsajl2NV9TRU1fX2tfODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjJfa18lMjZtc2Nsa2lkJTNkODM3NzgyMGYzNWQwMWRiMDBmN2RlNGNlZDUyNDQ5ZjI&rlid=8377820f35d01db00f7de4ced52449f2)[1](https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/tutorial-acm-create-budgets).

* **Support options in Azure**

Azure offers a variety of support options to cater to different needs and budgets. Here’s an overview of the main support plans available:

* **1. Basic Support: Cost**: Free
* **Features**:
  + Access to Azure documentation and whitepapers
  + Community support through forums like Microsoft Q&A and Stack Overflow
  + Service health status and notifications
* **2. Developer Support: Cost**: $29 per month
* **Features**:
  + Business hours access to support engineers via email
  + General guidance on best practices
  + Support for non-production environments
* **3. Standard Support: Cost**: $100 per month
* **Features**:
  + 24/7 access to support engineers via email and phone
  + Faster response times for critical issues
  + Support for production workloads
  + Unlimited subscription management support (e.g., billing, quota adjustments)
* **4. Professional Direct Support: Cost**: $1,000 per month
* **Features**:
  + All features of Standard Support
  + Access to a pool of ProDirect delivery managers
  + Onboarding services and proactive guidance
  + Architecture reviews and performance tuning
* **5. Premier Support: Cost**: Custom pricing
* **Features**:
  + All features of Professional Direct Support
  + Dedicated account manager
  + 24/7 problem resolution support
  + Onsite support and training
  + Comprehensive services for large enterprises
* **Azure Service Level Agreements**

Azure Service Level Agreements (SLAs) define the performance standards that Microsoft commits to for its Azure services. These agreements specify the uptime and connectivity guarantees for each service, ensuring that customers have a clear understanding of the reliability they can expect. Here’s an overview of key aspects of Azure SLAs:

**Key Components of Azure SLAs**

1. **Uptime Guarantee**: Azure SLAs typically guarantee a certain percentage of uptime for services. [For example, many services offer a 99.9% uptime guarantee, meaning the service will be available 99.9% of the time in a given month](https://k21academy.com/microsoft-azure/az-900/az-900-azure-service-level-agreements/)
2. **Service Credits**: If Azure fails to meet the SLA uptime guarantee, customers may be eligible for service credits. [These credits are a percentage of the monthly service fees and are applied to future billing cycles](https://k21academy.com/microsoft-azure/az-900/az-900-azure-service-level-agreements/)
3. **Performance Standards**: [SLAs also define performance standards for specific services, such as response times for virtual machines or data throughput for storage services2](http://claudiasittig.com/azure-virtual-machine-sla-understanding-service-level-agreements/).
4. **Exclusions**: SLAs include exclusions where the uptime guarantee does not apply. [These can include scheduled maintenance, customer-caused outages, or issues beyond Microsoft’s control](https://k21academy.com/microsoft-azure/az-900/az-900-azure-service-level-agreements/)

* **Cloud Adoption Framework**

The Microsoft Cloud Adoption Framework (CAF) for Azure is a comprehensive guide designed to help organizations plan and implement their cloud adoption journey. It provides best practices, documentation, and tools to ensure a successful transition to the cloud. Here’s an overview of the key components and phases of the Cloud Adoption Framework:

**Key Components of the Cloud Adoption Framework**

**Key Components and Phases**

1. **Strategy**:
   * **Motivations**: Understand why your business wants to move to the cloud.
   * **Business Outcomes**: Define what you want to achieve and how you’ll measure success.
   * **Financial Considerations**: Look at the costs and benefits of moving to the cloud.
2. **Plan**:
   * **Digital Estate**: List and evaluate your current digital assets (like servers and applications).
   * **Organizational Alignment**: Make sure your team and structure are ready for cloud adoption.
   * **Skills Readiness**: Ensure your team has the skills needed for cloud adoption.
3. **Ready**:
   * **Landing Zones**: Set up a secure and scalable cloud environment.
   * **Operating Model**: Define how you will manage and operate your cloud resources.
4. **Adopt**:
   * **Migrate**: Move your workloads (applications, data) to the cloud.
   * **Innovate**: Use cloud technologies to create new value and drive innovation.
5. **Govern**:
   * **Policy and Compliance**: Set up rules to ensure security and compliance.
   * **Cost Management**: Implement practices to control and manage cloud spending.
6. **Manage**:
   * **Operations Management**: Monitor and manage your cloud resources to ensure everything runs smoothly.
   * **Security and Resilience**: Implement security measures and ensure your business can continue operating in case of issues.

* **Resource Tags**

Resource tags in Azure are metadata elements that you can apply to your Azure resources to help organize and manage them more effectively. Tags consist of key-value pairs and can be used to categorize resources based on settings that are relevant to your organization

**How to Apply Tags**

**Using the Azure Portal**

1. **Select a Resource**: Go to the Azure Portal and select the resource you want to tag.
2. **Navigate to Tags**: In the resource’s menu, find and select the **Tags** option.
3. **Add Tags**: Click on **Add Tag** and enter the key and value for your tag. For example, Environment as the key and Production as the value.
4. **Save**: [After adding the necessary tags, click **Save** to apply them to the resourc](https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/tag-resources-portal)e

* **Management Groups**

**Azure Management Groups** are a way to organize and manage multiple Azure subscriptions. Think of them as folders that help you group and control access, policies, and compliance across all your subscriptions in a structured way.

**Key Features**

1. **Hierarchical Organization**: You can create a hierarchy of management groups and subscriptions, making it easier to apply policies and manage access consistently.
2. **Governance and Compliance**: Apply policies at the management group level to ensure all subscriptions within the group follow the same rules.
3. **Access Management**: Simplify role-based access control (RBAC) by assigning roles at the management group level, which then apply to all subscriptions within the group.
4. **Policy Assignment**: Policies set at the management group level cascade down to all subscriptions and resources within the group.
5. [**Cost Management**: Provides a unified view for monitoring and managing costs across multiple subscriptions1](https://learn.microsoft.com/en-us/azure/governance/management-groups/overview)[2](https://www.theknowledgeacademy.com/blog/azure-management-groups/).

**Example Scenario:** Imagine you work for a large company with multiple departments, each using its own Azure subscription. Here’s how you could use management groups:

1. **Create Management Groups**:
   * **Root Management Group**: This is the top-level group that contains all other management groups and subscriptions.
   * **Department Management Groups**: Create separate management groups for each department, like HR, Finance, and IT.
2. **Organize Subscriptions**: Place each department’s subscriptions under their respective management groups. For example, all HR-related subscriptions go under the HR management group.
3. **Apply Policies**: Apply a policy at the HR management group level to restrict the creation of resources to specific regions. This policy will automatically apply to all subscriptions under the HR group.
4. **Manage Access**: Assign roles at the Finance management group level to give finance team members access to all finance-related subscriptions. This way, you don’t have to assign roles individually to each subscription.

* **Azure Blueprints**

**Azure Blueprints** helps you set up and manage your cloud environment in a consistent and compliant way, like following a blueprint for building a house.

**Key Points**

1. **Templates**: Blueprints use templates to define what resources and settings you need.
2. **Artifacts**: These are the parts of a blueprint, like role assignments, policies, and resource groups.
3. **Consistency and Compliance**: Ensures every deployment follows your organization’s standards and rules.
4. **Versioning**: You can create different versions of your blueprints to manage changes easily.
5. **Tracking**: Keeps track of what should be deployed and what was actually deployed.

**How It Works**

1. **Create a Blueprint**: Define the resources and settings you need.
2. **Assign the Blueprint**: Apply the blueprint to a subscription or resource group.
3. **Deploy Resources**: The blueprint sets up the resources and settings as defined.

* **Azure Policy**

Azure Policy is a service in Azure that allows you to create, assign, and manage policies to enforce organizational standards and assess compliance at scale. **Azure Policy** helps you enforce rules and ensure your resources comply with your organization’s standards.

**Key Points**

1. **Policy Definitions**: Policies are written in JSON and describe the rules and actions to take if conditions are met.
2. **Policy Assignments**: Policies are applied to a scope, like a subscription or resource group, and affect all resources within that scope.
3. **Compliance Evaluation**: Continuously checks resources to ensure they follow the policies.
4. **Remediation**: Can automatically or manually fix resources that don’t comply with the policies.
5. **Built-in Policies**: Azure provides ready-made policies for common needs, like ensuring resources are in allowed regions.

**How It Works**

1. **Create a Policy**: Define the rules and conditions for your policy.
2. **Assign the Policy**: Apply the policy to a scope, like a subscription or resource group.
3. **Evaluate Compliance**: Azure checks resources against the policy to ensure they comply.
4. **Remediate Non-Compliance**: Automatically or manually fix resources that don’t follow the policy.

**Example Use Case:** Imagine you want to ensure that all resources in a specific subscription have a tag named Environment with the value Production:

1. **Create a Policy Definition**: Define a policy that checks for the Environment tag and its value.
2. **Assign the Policy**: Assign this policy to the desired subscription.

* **Azure resource locks**

Azure resource locks are a feature that helps you protect your resources from accidental or unauthorized changes. Here’s a detailed overview of how they work and how to use them:

**Key Features of Azure Resource Locks**

1. **Types of Locks**:
   * **CanNotDelete**: This lock allows authorized users to read and modify a resource but prevents them from deleting it.
   * **ReadOnly**: This lock allows authorized users to read a resource but prevents them from deleting or modifying it. [This is similar to restricting all users to the permissions provided by the Reader role1](https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources).
2. **Scope of Locks**: Locks can be applied at different scopes, including subscriptions, resource groups, and individual resources. [When a lock is applied at a parent scope, all resources within that scope inherit the lock1](https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources).
3. **Inheritance**: Locks are inherited by all child resources within the scope. [For example, if you apply a lock to a resource group, all resources within that group will inherit the lock1](https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources).

**How to Apply Resource Locks**

**Using the Azure Portal**

1. **Select the Resource**: Go to the Azure Portal and navigate to the resource you want to lock.
2. **Navigate to Locks**: In the resource’s menu, find and select **Locks** under the **Settings** section.
3. **Add a Lock**:
   * Click on **+ Add** to create a new lock.
   * Provide a **Lock Name** and select the **Lock Type** (CanNotDelete or ReadOnly).
   * Optionally, add a **Notes** field to describe the purpose of the lock.
   * [Click **OK** to apply the lock](https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources)

* **Azure CLI**

The Azure Command-Line Interface (CLI) is a powerful tool that allows you to manage Azure resources directly from your command line. It is designed to be simple and easy to use, enabling you to automate tasks and manage your Azure environment efficiently. Here’s an overview of its key features and how to get started:

**Key Features of Azure CLI**

1. **Cross-Platform**: Azure CLI is available on Windows, macOS, and Linux. [It can also be run in a Docker container or accessed through the Azure Cloud Shell](https://learn.microsoft.com/en-us/cli/azure/)[1](https://learn.microsoft.com/en-us/cli/azure/).
2. **Resource Management**: You can create, update, delete, and manage Azure resources using simple commands. [This includes virtual machines, storage accounts, databases, and more](https://learn.microsoft.com/en-us/cli/azure/)[1](https://learn.microsoft.com/en-us/cli/azure/).
3. **Automation**: Azure CLI supports scripting, allowing you to automate repetitive tasks. [You can write scripts in Bash, PowerShell, or any other scripting language that can call command-line tools](https://learn.microsoft.com/en-us/cli/azure/)[1](https://learn.microsoft.com/en-us/cli/azure/).
4. **Interactive Mode**: [The interactive mode provides command suggestions and auto-completion, making it easier to discover and use commands1](https://learn.microsoft.com/en-us/cli/azure/).

**Basic Commands**

1. **Login**:

az login

1. **List Subscriptions**:

az account list

1. **Create a Resource Group**:

az group create --name MyResourceGroup --location eastus

1. **Create a Virtual Machine**:

az vm create --resource-group MyResourceGroup --name MyVM --image UbuntuLTS --admin-username azureuser --generate-ssh-keys

1. **Delete a Resource Group**:

az group delete --name MyResourceGroup --yes --no-wait

* **Azure Powershell**

Azure PowerShell modules are a powerful way to manage and automate your Azure resources using PowerShell. Here’s a detailed guide on how to get started with Azure PowerShell modules:

**Key Features of Azure PowerShell**

1. **Cross-Platform**: Azure PowerShell works on Windows, macOS, and Linux. [It can also be accessed through the Azure Cloud Shell](https://learn.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-0.10.0)[1](https://learn.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-0.10.0).
2. **Resource Management**: You can create, update, delete, and manage Azure resources using cmdlets. [This includes virtual machines, storage accounts, databases, and more](https://learn.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-0.10.0)[1](https://learn.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-0.10.0).
3. **Automation**: Azure PowerShell supports scripting, allowing you to automate repetitive tasks. [You can write scripts in PowerShell to manage your Azure environment efficiently](https://learn.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-0.10.0)[1](https://learn.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-0.10.0).

**Basic Commands**

1. **Login**: Sign in to your Azure account:

Connect-AzAccount

1. **List Subscriptions**: View your Azure subscriptions:

Get-AzSubscription

1. **Create a Resource Group**: Create a new resource group:

New-AzResourceGroup -Name MyResourceGroup -Location EastUS

1. **Create a Virtual Machine**: Create a new virtual machine:

New-AzVM -ResourceGroupName MyResourceGroup -Name MyVM -Image UbuntuLTS -Credential (Get-Credential)

1. **Delete a Resource Group**: Delete a resource group and all its resources:

Remove-AzResourceGroup -Name MyResourceGroup -Force

* **Azure Cloud Shell**

Azure Cloud Shell is an interactive, authenticated, browser-accessible terminal for managing Azure resources. It provides a flexible and convenient way to use either Bash or PowerShell directly from your browser.

**How to Use Azure Cloud Shell**

**Accessing Cloud Shell**

1. **Azure Portal**: Sign in to the Azure Portal and click on the Cloud Shell icon in the top navigation bar.
2. **Direct URL**: Go to shell.azure.com to access Cloud Shell directly.
3. **Azure Mobile App**: Use the Azure mobile app to access Cloud Shell on the go.
4. **Visual Studio Code**: Use the Azure Account extension in Visual Studio Code to open Cloud Shell within the editor.

**Basic Commands**

1. **Starting Cloud Shell**: When you first start Cloud Shell, you’ll be prompted to choose between Bash and PowerShell. Select your preferred shell.
2. **Creating a Resource Group**:
   * In Bash:

az group create --name MyResourceGroup --location eastus

In PowerShell:

New-AzResourceGroup -Name MyResourceGroup -Location EastUS

1. **Listing Resources**:
   * In Bash:

az resource list

* + In PowerShell:

Get-AzResource

1. **Using the Integrated Editor**:
   * Open the editor by typing code . to edit files directly within Cloud Shell.

* **ARM Template**

**ARM Templates** are a way to define and deploy your Azure infrastructure using code. They help you automate the setup of your Azure resources in a consistent and repeatable manner. They allow you to define the infrastructure and configuration for your Azure resources using a declarative JSON format. Here’s a detailed overview of ARM Templates:

1. **Declarative Syntax**: ARM templates use JSON to describe what you want to create. You tell Azure what you need, and it sets it up for you.
2. **Infrastructure as Code**: Treat your infrastructure like software. You can store your templates in version control systems like Git, update them, and deploy them just like you would with application code.
3. **Idempotent**: You can deploy the same template multiple times, and it will always produce the same result without creating duplicates.
4. **Modular and Reusable**: Break down templates into smaller parts that you can reuse. Link templates together to create complex setups.
5. **Parameterization**: Use parameters to make your templates flexible. You can pass different values to the same template to deploy resources with different settings.

**How It Works**

1. **Template File**: The template is a JSON file that defines the resources and their settings. It includes sections like parameters, variables, resources, and outputs.
2. **Deployment**: Deploy the template using tools like Azure PowerShell, Azure CLI, or the Azure portal. Azure Resource Manager (ARM) processes the template and sets up the resources.
3. **Template Sections**:
   * **Parameters**: Values you can pass during deployment to customize the resources.
   * **Variables**: Values you can reuse within the template.
   * **Resources**: The Azure resources you want to create.
   * **Outputs**: Values returned after deployment, like resource IDs or connection strings.

* **Azure advisor, azure service health, azure monitor service**
* **Azure Advisor**

**Azure Advisor** is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. [It analyzes your resource configuration and usage and provides recommendations to improve the cost-effectiveness, performance, reliability, and security of your Azure resources](https://www.bing.com/aclick?ld=e8MmCPMrZkPlx7NKip0ARG3jVUCUxfBzta5BMDHWkmkCpmobc49Np8wga0gdTMS0ly8SqlmK9U_ztAFwT8IuZHHx1FwdKzHyfUR7FdGkzNHBgx8E6mYG6slaz5zshzqQFwoyZj_vKNbpGFmSC8XNjvwmlJFgadHmgNxTDvSogiz3yALH34&u=aHR0cHMlM2ElMmYlMmZhenVyZS5taWNyb3NvZnQuY29tJTJmZW4taW4lMmZzZXJ2aWNlcyUyZm1vbml0b3IlMmYlM2ZlZl9pZCUzZF9rXzFjNWIyNzQwMDg4MjFiODRhYjRiNDFmNzUyZmE0MDljX2tfJTI2T0NJRCUzZEFJRGNtbWYxZWxqOXY1X1NFTV9fa18xYzViMjc0MDA4ODIxYjg0YWI0YjQxZjc1MmZhNDA5Y19rXyUyNm1zY2xraWQlM2QxYzViMjc0MDA4ODIxYjg0YWI0YjQxZjc1MmZhNDA5YyUyM292ZXJ2aWV3JTJm&rlid=1c5b274008821b84ab4b41f752fa409c)[1](https://azure.microsoft.com/en-us/products/advisor).

**Key Features**:

* **Actionable Recommendations**: Provides suggestions to optimize your resources for reliability, security, operational excellence, performance, and cost.
* **Advisor Score**: Helps you assess how well your workloads follow best practices and prioritize the most impactful recommendations.
* **Integration**: Accessible through the Azure portal, Azure CLI, or Advisor API. [You can also configure alerts to notify you about new recommendations](https://azure.microsoft.com/en-us/products/advisor)[2](https://learn.microsoft.com/en-us/azure/advisor/advisor-overview).
* **Azure Service Health**

**Azure Service Health** keeps you informed about the health of your Azure services and resources. [It provides personalized alerts and guidance when Azure service issues like outages and planned maintenance affect you](https://azure.microsoft.com/en-us/products/advisor)[3](https://azure.microsoft.com/en-us/get-started/azure-portal/service-health/).

**Key Features**:

* **Personalized Dashboard**: Shows service issues that affect your resources, including active incidents, planned maintenance, and health advisories.
* **Customizable Alerts**: Notifies you of service incidents, planned maintenance, and health advisories via email, SMS, push notifications, or other channels.
* [**Guidance and Support**: Offers guidance during service incidents and provides root cause analyses (RCAs) for incidents](https://learn.microsoft.com/en-us/azure/service-health/overview)
* **Azure Monitor**

**Azure Monitor** is a service that helps you keep an eye on your applications and infrastructure to make sure everything is running smoothly.

**Key Points**

1. **Collects Data**: Gathers information from different sources like applications, virtual machines, and databases.
2. **Aggregates Data**: Combines all the collected data into one place.
3. **Analyzes and Visualizes Data**: Provides tools to help you understand the data, like dashboards and reports.
4. **Alerts and Automation**: Lets you set up alerts to notify you of issues and automate responses to fix them.
5. **Integration**: Works with other Azure services and third-party tools for a seamless experience.

**How It Works**

1. **Data Sources**: Collects data from various sources like applications and virtual machines.
2. **Data Collection and Routing**: Sends the collected data to Azure Monitor’s data platform.
3. **Data Platform**: Stores and processes the data.
4. **Consumption**: Use tools like Azure Monitor Metrics and Logs to analyze the data.

* **Azure Monitor Alerts**

**Azure Monitor Alerts** help you detect problems early and notify you so you can fix them before they affect your users.

**Key Points**

**1. Types of Alerts**:

* + **Metric Alerts**: Monitor resource metrics and trigger alerts based on conditions.
  + **Log Search Alerts**: Use queries to check logs and trigger alerts based on the results.

1. **Alert Rules**: Define conditions that trigger an alert, like monitoring specific metrics or logs.
2. **Action Groups**: Define actions to take when an alert is triggered, like sending notifications or starting automated workflows.
3. **Alert Processing Rules**: Modify alerts by adding or suppressing actions, applying filters, or scheduling rules.

**How It Works**

* + **Create Alert Rules**: Set up rules to monitor specific metrics or logs and define conditions for triggering alerts.
  + **Trigger Alerts**: When conditions are met, an alert is triggered, and the associated actions are initiated.
  + **Respond to Alerts**: Alerts notify you of issues via email, SMS, or other methods, and can also trigger automated actions to resolve the issue.
* **Log Analytics**

**Azure Log Analytics** is a tool within Azure Monitor that allows you to edit and run log queries against data collected from your Azure resources. It helps you gain insights into your environment by analyzing and visualizing log data. Here’s a detailed overview of its key features and how to use it:

**Key Features of Azure Log Analytics**

1. **Data Collection**: Collects log data from various sources, including Azure resources, on-premises environments, and other cloud providers. [This data is stored in a Log Analytics workspace](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview)
2. **Querying Data**: Use the **Kusto Query Language (KQL)** to write queries that retrieve and analyze log data. [KQL is a powerful language designed for querying large datasets](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview)[1](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview).
3. **Visualization**: Create charts, graphs, and dashboards to visualize the results of your queries. [This helps in identifying trends, patterns, and anomalies in your data](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview)
4. **Alerts and Automation**: Set up alerts based on log queries to notify you of specific conditions or issues. [You can also automate actions in response to these alerts](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview)[1](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview).
5. **Integration**: [Integrates with other Azure services like Azure Monitor, Azure Security Center, and Azure Sentinel to provide a comprehensive monitoring and security solution](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview)

* **Using Application Insights**

**Azure Application Insights** is a tool that helps you keep an eye on how your applications are performing and how users are interacting with them. Here’s what it does and how you can use it:

**What It Does**

1. **Collects Data**: It gathers telemetry data about your app, like how many people are using it, how fast it’s running, and if there are any errors.
2. **Monitors Performance**:
   * **Application Map**: Shows a visual map of how different parts of your app connect and interact.
   * **Live Metrics**: Gives you real-time data on how your app is performing right now.
   * **Transaction Search**: Lets you look into specific actions or transactions to find and fix issues.
3. **Analyzes User Behavior**:
   * **Usage Analytics**: Tracks how users are using your app, like which features they use the most.
   * **Funnels and Flows**: Helps you see the paths users take through your app and where they might drop off.
4. **Proactive Monitoring**:
   * **Availability Tests**: Regularly checks if your app is up and running.
   * **Smart Detection**: Automatically alerts you if something seems off.
5. **Integrates with Other Tools**: Works well with other Azure services and supports multiple programming languages.

**Example**

Imagine you have a web app and you want to make sure it’s running smoothly and that users are having a good experience:

1. **Create an Application Insights Resource** in the Azure Portal.
2. **Add the SDK to Your App** to start collecting data.
3. **Monitor Performance** using the tools provided.
4. **Analyze User Behavior** to see how people are using your app.
5. **Set Up Alerts** to get notified if there are any issues.