# AZ 900 Study Guide

# !Describe Cloud Concepts (25-30%)

## !Describe cloud computing

### Define Cloud Computing

The delivery of computing services over the internet - includes things like:

- VMs
- Storage
- Databases
- Networking
  Cloud computing allows for rapid expansion, unlike on-prem
- compute power and storage are the main two things - both scalable

### Describe the shared responsibility model

The shared responsibility model describes what the client and what the cloud provider must be responsible for.
In an on-prem environment the IT team would be responsible for everything.
**the client is always responsible for**:

- information and data stored in the cloud
- what devices can connect to the cloud
- accounts and identities of the people/services/devices in your organization
  **the cloud provider is always responsible for:**
- physical datacenter
- physical network
- physical hosts
  **shared responsibilities (depending on service types, IaaS, PaaS, SaaS)**
- OS
- networking controls
- apps
- identity and directory infrastructure

### Define cloud models, including Public, Private, and Hybrid

**Private Cloud:** a single entity owned datacenter that provides IT services over the internet

- the organization has complete control over the resources and security
- may be hosted on prem - or by a third party company that allocates a specific datacenter to a company
  **Public Cloud:** a third party provider that allows anyone who pays wants to purchase cloud services, access and use the resources.
- *general public availability is the big difference between private and public*
  **Hybrid Cloud:** an environment that uses both public and private cloud interconnectedly
- can be used to allow a private cloud to quickly increase their resources upon a surge of workload by using public cloud
- can also be used for extra security to keep some services in their private cloud, and other not as important services in the private cloud

**Multi Cloud:** using multiple public clouds and manage resources and security in both environments.
**Azure Arc:** set of technologies that help you manage your cloud environment (public cloud, private cloud, hybrid, or even multi-cloud)
Azure VMWare: can seamlessly migrate to public/hybrid cloud

## Identify appropriate use cases for each cloud model

**Private Cloud Use Cases:**

- when you want to keep all of your data and infrastructure local but have people that need to access it that are not physically there
- laws around where data can be stored, best way of making sure your data doesn't cross borders
  **Public Cloud Use Cases:**
- when things need to be quickly provisioned and deprovisioned
- when you don't have existing infrastructure and you want to pay for only what you use
  **Hybrid Cloud Use Cases:**
- when you need more control over your cloud, but need occasional scalability

## Describe the consumption-based model

Cloud computing uses a consumption-based model - meaning you only pay for what you use.

- you don't pay for the power, security, or any other cost of maintenance, just the resources that you used.
- if you need more VMs, you pay for them and get them, if you don't need as many, remove them and stop paying for them.

*Capital Expenditures (CapEx) - upfront costs of things: buying a new building, purchasing a datacenter, etc.*
*Operational Expenditures (OpEx) - over time costs to things: renting a convention center, signing up for cloud services, etc.*

### Compare cloud pricing models

Cloud computing is the delivery of computing services over the internet. It uses a "pay-as-you-go" pricing model.

- this is essentially renting compute power and storage from someone else's datacenter, and only paying for what you use.

### Describe serverless

Allows developers to ignore the infrastructure management and get into development.

- Also allows for scalability (10s of thousands of requests)
- Only pay for what you use - only for the time your program is being run.

## !Describe the benefits of using cloud services

### Describe the benefits of high availability and scalability in the cloud

**high availability** is linked with SLA (service level agreements) to ensure that the customer gets their guaranteed amount of uptime.

- 99%, 99.9%, 99.95%, 99.99% are all different SLA %s to ensure uptime. The higher the percentage of uptime, the higher the cost.
  High availability is important because if you don't have your service up, you could be losing money.

**Scalability** is the ability to adjust how many resources you need at a particular time to meet demand. This could be increasing or decreasing number of resources.
**Vertical scaling** - increasing the capabilities of resources (think adding/removing RAM or processing power to a single VM)
**Horizontal Scaling** - increasing the number of resources you have (think deploying more VMs to load balance)

### Describe the benefits of reliability and predictability in the cloud

**Reliability** is the ability of a system to recover from failures and continue to function. (one of the pillars of Microsoft Azure Well-Architected Framework)

The benefit of having reliability is that even if there is a catastrophic failure in one region, your infrastructure can be transferred to another unaffected location and continue working.

**Predictability** can be focused on performance predictability or cost predictability. Both are heavily influenced by the Microsoft Azure Well-Architected Framework.
**Performance Predictability** - predicts which resources are needed to deliver a positive experience for your customers.

- Autoscaling, load balancing, and high availability are some concepts that support performance predictability.
- autoscaling - increasing or decreasing resources to meet demands | load balancing - redirects traffic from an overloaded area to a lesser used area.
  **Cost Predictability** - predicts how much your cloud infrastructure is going to cost.
- you can track your resource use in real time, make sure you using your resources efficiently, and use data analytics to find patterns and trends that help better plan resource deployments

## Describe the benefits of security and governance in the cloud

**Governance in the cloud:** the cloud can help you meet corporate standards and government regulatory requirements.
Things like set templates can help ensure all the deployed resources meet those rules, and can update them as standards change.
Cloud-based auditing can help find any resource not following standards and can provide strategies to fix them.
Updates and patches can also be automatically applied which can help with governance and security.

**Security in the cloud:** if you want max control of security, **IaaS** is a good option, if you want to not have to worry about updates/patches **PaaS** or **SaaS** might be a better fit.
Because cloud is over internet, cloud providers have to be good at handling things like DDoS attacks, which means your network is more secure.

## Describe the benefits of manageability in the cloud

**Management of the cloud** - managing your cloud resources

- automatically scale resource deployment based on need
- deploy resources based on a preconfigured template, removing the need for manual configuration
- monitor the health of resources and automatically replace failing resources

- receive automatic alerts based on configured metrics, so you're aware of performance in real time
  **Management in the cloud** - how you're able to manage your cloud environment and resources
- through a web portal
- using a command line interface
- using APIs
- using PowerShell

# !Describe cloud service types

## Describe Infrastructure as a Service (IaaS)

**IaaS** gives you the maximum amount of control for your cloud resources.
You are essentially renting the hardware in a cloud datacenter (the cloud provider maintains it) - you are then responsible for everything else.

- installation, configuration, updates, and security is all something you are responsible while using **IaaS**

## Describe Platform as a Service (PaaS)

**PaaS** is the middle ground between **IaaS** (renting a datacenter) and **SaaS** (complete and deployed solution).
The cloud provider maintains the datacenter, as well as the operating systems, middleware, development tools, and business intelligence services that make up a cloud solution

- you don't have to worry about the licensing for OS or databases (IoT, SQL, etc.)
- you, or the cloud provider (depending on the config) **will be responsible** for networking settings, connectivity within your cloud environment, network and application security, and the directory infrastructure.

## Describe Software as a Service (SaaS)

**SaaS** is essentially renting/using a fully developed application. Email, financial software, messaging apps, and connectivity software are all common examples of SaaS implementations.

- least amount of customization but easiest to get up and running.
- only responsible for the data you put in, the devices that can connect, and the users.

### Identify appropriate use cases for each cloud service (IaaS, PaaS, and SaaS)

**IaaS Use Cases:**

- lift-and-shift migration - set up an environment similar to your on-prem datacenter, then transitioning from on-prem to IaaS infrastructure
- testing and development - using templates, you can rapidly replicate testing environments, then when they are no longer needed, shut them down.
  **PaaS Use Cases:**
- development framework - PaaS acts as a framework that developers can build upon to create cloud-based applications. It lets developers create applications using built-in software components.
  Features like: scalability, high-availability, and multi-tenant capacity are included - reducing amount of coding
- Analytics or business intelligence - tools provided as a service with PaaS allow organizations to analyze their data finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns, and other business decisions.
  **SaaS Use Cases:**
- Email and Messaging - teams, slack, discord
- Business and productivity applications - Office 365
- finance and expense tracking

# !Describe Azure Architecture and Services (35-40%)

## !Describe the core architectural components of Azure

### !Describe Azure regions, region pairs, and sovereign regions

**Regions** - a geographical area on the planet that has at least one, potentially more datacenters nearby and networked together with a low-latency network.

- Azure assigns and controls the resources within each region to ensure workloads are appropriately balanced.
  **Region Pairs** - Azure regions paired with another region within the same geography (example, US west and US east paired together) at least 300 miles away.
- this allows for the replication of resources across a geography that helps reduce the likelihood of interruptions from natural disasters, power outages, or physical network

outages that affect and entire region.
- if a region in a pair was to fail, services would failover to the other region in the pair.
  **Sovereign Regions** - instances of azure that are isolated from the main instance of azure - often used for compliance or legal purposes.
- US DoD central, US Gov Virginia, US Gov Iowa, etc. - All have their own azure instances.
- China east, China north, etc. - All have their own azure instances, Microsoft doesn't own any of the datacenters.

## !Describe availability zones

An **Availability Zone** are physically separate datacenters within an Azure region.

- Each availability zone is made up of one or more datacenters, each equipped with independent power, cooling, and networking.
- An availability zone is set up to be an isolation boundary. if one zone goes down, the other keeps working.
  Availability zones can be used to run mission-critical apps and build high-availability into them by co-locating compute, storage, networking, and data resources in one availability zone and replicating them to another.

## Describe Azure datacenters - ?

## !Describe Azure resources and Resource groups

**Resource** - a resource is the basic building block of Azure. Anything you create, provision, deploy, etc. is a resource.

- VMs, virtual networks, databases, cognitive services, etc. are all considered resources within Azure.
  **Resource Groups** - groupings of resources.
- every resource needs to be in a resource group (only one resource group per resource)
- doing something to a resource groups applies to all the resources in that group (ex. deleting the group deletes all the resources)

## !Describe Subscriptions

Azure Subscriptions are a unit of management, billing, and scale - allow you to logically organize your resource groups and facilitate billing.
A subscription provides you with authenticated ad authorized access to Azure products and services.

- A subscription links to an Azure account, which is an identity in Microsoft Entra ID, or in a directory that Microsoft Entra ID trusts.
  An account is required to have one subscription, but it can have more subscriptions and create boundaries around products, services, and resources.
- Billing Boundary - determines how an Azure account is billed for using Azure. Azure generates separate billing reports and invoices for each subscription so that you can organize and manage costs.
- Access control boundary - applies access-management policies at the subscription level. An example would be like giving different departments different azure subscription policies to control what they would be allowed to access.

### !Describe management groups

Management groups are what subscriptions can be organized with, and applied governance conditions.

- Subscriptions inherit settings from management groups
- management groups can be nested up to 6 levels deep (not including root level or subscription level)
- 10,000 management groups can be supported in a single directory
- each management group and subscription can support only one parent

### !Describe the hierarchy of resource groups, subscriptions, and management groups

Resources > Resource Groups > Subscriptions > Management Groups

- rules follow through (propagate top down)

## !Describe the Azure Compute and Networking services

### !Compare compute types, including containers, virtual machines, and functions

**Virtual Machines** are machines that need to use an operating system, and allow for the most customization in general.
**Containers** are a virtualization environment that don't need an OS to run on. They are lightweight and designed to be created, scaled out, and stopped dynamically.

- Containers are more light weight, and agile than VMs.
- You can quickly restart containers if there is a crash or hardware interruption. - only needs to relaunch the app, not the OS & app

- if you need to run multiple instances of the same app - containers are your guy.
  **Azure Container Instances** - fastest and simplest way to run a container in Azure - without having to manage any VMs or adopt any additional services.
- Azure Container Instances allow you to upload your containers, then the service will run the containers for you
  **Azure Container Apps** - remove the container management piece of Azure Container Instances and you have Azure Container Apps.
- Azure Container Apps can also incorporate load balancing and scaling.
  **Azure Kubernetes Service** - a container orchestration service. manages the lifecycle of containers. Can make deployment of a fleet of containers simpler and more efficient.
  **Functions** - event-driven, serverless compute option that doesn't require maintaining VMs or Containers.
- if you build an app on a VM or container, those resources have to be "running" for your app to work.
- With functions, an even wakes the function, which means you don't need to keep resources provisioned when there are no events.
  Functions are ideal when you're only concerned about the code, and not the platform its running on - ex. when a task needs to be performed based on a request
  functions are auto scaling
  stateless (default) - behave as if they were just restarted
  stateful (durable functions) - context is passed through the function to track prior activity.
  Overall:
- **VMs** are best for needing customization but are slow and costly
- **Containers** are better for running specific applications, are very quick, and easy to deploy
- **Functions** are used for running code with no need for an underlying system

## !Describe virtual machine options, including Azure virtual machines, Azure Virtual Machine Scale Sets, availability sets, and Azure Virtual Desktop

**VMs** provide **IaaS** in the form of a virtualized server. VMs are an ideal choice when you need:

- total control over the OS
- the ability to run custom software
- to use custom hosting configurations
  VMs have the ability to create and or use images that can cut down provision time.
  **Scale VMs** - are VMs that are grouped together for high availability, scalability, and redundancy
  **VM Scale Sets** - let you create and manage a group of identical, load-balanced VMs.

- azure automates most of the work allowing you to centrally manage, configure, and update a large number of VMs in minutes.
- number of VMs can also be increased of decreased in response to demand, or you can set it to scale based on a schedule.
- deploy a load balancer to make sure your resources are being used efficiently.
  **VM Availability Sets** - designed to ensure that VMs stagger updates and have varied power and network connectivity - this prevents you from losing all your VMs with a single network or power failure.
- this is done by grouping VMs by: update domain, and fault domain.
  - **Update Domain** - groups VMs that can be rebooted at the same time, this lets you update half of the VMs, then update the other half later (this is done for no downtime)
  - **Fault Domain** - groups VMs by common power source and network switch. by default, an availability set will split your VMs across up to 3 fault domains. This helps protect against a physical power or networking failure by having VMs in a different fault domain.
  no extra cost for configuring an availability set - only pay for the VM instances you create.
  **Azure Virtual Desktop** - data and apps are not on local hardware, they are access remotely meaning there is a much lower chance for confidential info being left on a personal device. Actual services are running in the cloud.

## !Describe the resources required for virtual machines

**Size** - the purpose, number of processor cores, and amount of RAM
**Storage Disks** - hard disk drives, solid state drives, etc.
**Networking** - virtual network, public IP address, and port configuration

## !Describe application hosting options, including web apps, containers, and virtual machines

**Virtual Machines** can be used for giving maximum control of the hosting environment and allows you to configure it exactly how you want.
**Containers** give you the ability to isolate and individually mange different aspects of the hosting solution.
**Azure App Service** allows you to build and host: web apps, background jobs, mobile back-ends, and RESTful APIs in the language of your choice **without managing infrastructure.**
- it offers auto scaling and HA
- types of app service styles: Web apps, API apps, WebJobs, Mobile apps.
**Web Apps** - app service incudes full support for hosting web apps by using ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python.
**API Apps** - you can build REST-based web APIs - ability to package and publish your API in Azure Marketplace. Produced apps can be consumed from any HTTP- or HTTPS-based client
**WebJobs** - can be used to run a program or script in the same context as a web app, API app,

or mobile app. - often used to run background tasks as part of your application logic.
**Mobile Apps** - build a back end for iOS and Android apps - store mobile app data, authenticate customers, send push notifications, execute custom back-end logic in C# or node.js

## !Describe virtual networking, including the purpose of Azure virtual networks, Azure virtual subnets, peering, Azure DNS, Azure VPN Gateway, and ExpressRoute

**Azure Virtual Networks** allow Azure resources to communicate with each other, with users on the internet, and with your on-prem client computers.
**Azure Isolation and Segmentation (virtual subnets)** - Azure virtual network allows you to make multiple subnets with public or private IP schemes, but they aren't internet routable. They are only available in the virtual network.

- you can also use the Azure DNS, or use an internal or external DNS server.
  **Peering** allows two virtual networks to connect directly to each other, travelling over the Microsoft backbone network (not public internet).
- peered networks can be in separate regions and still connect privately and securely.
  **Azure DNS** - allows you to use Azure for DNS hosting (puts all your eggs in one basket if you will).
- Allows for reliability and high availability - using Azure's global network of DNS servers.
- Security - based on Azure Resource Manager (ARM)
- Easy to use
- Allows for Alias Records
  **Azure VPN Gateway** - you can have one gateway per location, but each gateway can connect to multiple different sites. Encrypted traffic, travels over internet.
- site-to-site (on-prem to virtual network)
- point-to-site (device to virtual network)
- network-to-network (virtual network to virtual network)
  Policy-based (based on packet contents) or Route-based (based on IP routing - preferred for on-prem).
  **ExpressRoute** - allows you to connect your on-prem networks to the Microsoft cloud with a private connection. Each of these connections is called an **ExpressRoute circuit**.
- ExpressRoute connections don't go over the public internet, meaning they are more reliable, faster, lower latency, and more secure.
- basically physical peering - a private connection that goes through Microsoft's backbone.

## !Define Public and Private endpoints

**Public Endpoints** - have a public IP address and can be accessed from anywhere in the world.
**Private Endpoints** - exist within a virtual network and have a private IP address from within that address space of that virtual network

# !Describe Azure Storage Services

## !Compare Azure Storage Services

**Azure Blobs** - can store massive amounts of data, unstructured (no restrictions) - manage thousands of uploads, massive amounts of video data - accessible anywhere with internet connection via URL.

- ideal for:
  - serving images or documents directly to a browser
  - storing files for distributed access
  - streaming video and audio
  - storing data for backup and restore, disaster recovery, and archiving
  - storing data for analysis by an on-premises or Azure-hosted service

  **Azure Files** - for file shares - uses SMB or NFS protocols, can be mounted concurrently by cloud or on-prem deployments. SMB for all, NFS for Linux/macOS
- benefits:
  - shared access - seamlessly replace on-prem file shares with this (using SMB/NFS)
  - fully managed - don't need to manage hardware or an OS
  - Scripting and tooling - PowerShell cmdlets and Azure CLI can be used to manage azure file shares - can also use Azure portal and Azure Storage Explorer
  - Resiliency - in the cloud, don't have to worry about power outages
  - Familiar programmability - azure apps can use I/O APIs - also can use Azure Storage Client Libraries or Azure Storage REST API

  **Azure Queues** - used for storing large numbers of messages - each message can be up to 64KB - can be access via HTTP/HTTPS
- often used to create a backlog of work to process asynchronously
- can be combined with Azure Functions - allows you to do an action after a message is received

  **Azure Disks** - block-level storage volumes for use with Azure VMs
- essentially virtualized disks - all you have to do is provision the disk

  **Azure Tables** - stores large amounts of structured data - NoSQL datastore.
- allows you to use Azure Tables to build hybrid / multi-cloud solution and have data always available
- tables are ideal for storing structured, non-relational data.

## !Describe Storage Tiers

**Hot Access Tier** - optimized for data that is accessed frequently (like images for a website)
**Cool Access Tier** - optimized for data that is infrequently accessed and stored for at least 30 days (like customer invoices)
**Cold Access Tier** - optimized for data that is infrequently accessed and stored for at least 90 days
**Archive Access Tier** - appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (like long-term backups)

- hot and cool access tiers can be set at the account level
- hot, cool, cold, and archive can be set at the blob level, during or after upload.
- cool and cold data - lower availability SLA, and higher access costs - lower storage costs
- archive storage stores data offline - lowest storage costs, highest cost to rehydrate and access data.

## !Describe Redundancy Options

**Primary region redundancy**
Data is always replicated three times in the primary region.

- These are two options for how.
  **Locally redundant storage** - replicates your data 3 times within a single data center in the primary region.
  - provides 11 nines of durability (99.99999999999%)
  - lowest-cost, least durable. if the entire datacenter has a disaster, all storage may be lost.
    **Zone-Redundant storage** - replicates your storage across 3 availability zones (3 separate datacenters) within the primary region.
  - provides 12 nines of durability (99.999999999999%)
  - data still available if a zone goes down, follows country/region governance requirements (all in same region)

**Secondary region redundancy**
Used for applications that need high durability - copies your data to a secondary region hundreds of miles away for catastrophic failure protection.

- based on region pairs, can't be changed
- secondary region info can be failed over to be read, or enabled (this then turns into Read access - GRS, or read access GZRS)
  **Geo-Redundant storage** - replicates 3 times in a single datacenter, then 3 times in a second datacenter.
  - (LRS but with an added region)

- provides 16 nines of durability (99.9999999999999999%)
  **Geo-Zone-Redundant storage** - replicates your storage across 3 availability zones in the primary region, then copies it 3 times in a single location in the secondary region.
- (ZRS but with an added region LRS)
- provides 16 nines of durability (99.9999999999999999%)

## !Describe storage account options and storage types

**Storage Account** - provides a unique namespace for your storage that's accessible from anywhere over the web (HTTP/HTTPS)

- data is secure, highly available, durable, and massively scalable.
  **Storage Account Types**
- Standard general-purpose V2 - most recommended, used with: blobs, file shares, queues, and tables
- Premium Block Blobs - Premium storage for block blobs and append blobs - used for high transaction rates, smaller objects, or a consistently low latency.
- Premium File Shares - premium storage for file shares only. recommended for enterprise or high-performance scale applications. Used for support with **SMB** and **NFS** file shares
- Premium Page Blobs - premium storage for page blobs only.
  **Storage Account Endpoints** - using the unique name it gives you an address to get there - `<storage-account-name>.<INSERT-HERE>.core.windows.net`
- `INSERT-HERE` = `blob`, `dfs`, `file`, `queue`, `table`

## !Identify options for moving files, including AzCopy, Azure Storage Explorer, and Azure File Sync

**AzCopy** - a command line utility that can be used to copy blobs or files to or from your storage account.

- can upload, download, copy, or even sync (one way, basically copy something somewhere) files.
  **Azure Storage Explorer** - a standalone app that provides a GUI to manage files and blobs in your Azure Storage Account.
- uses AzCopy on the backend
  **Azure File Sync** - a tool that lets you centralize your file shares in Azure files - bi-directionally syncing on-prem to azure files.
- lets you use SMB, NFS, FTPS
- have as many caches as you need globally

- replace a failed local server by installing Azure File Sync on a new server in the same datacenter
- configure cloud tiering so that the most frequently accessed files are replicated locally, while infrequently accessed files are kept in the cloud until requested.

## !Describe migration options, including Azure Migrate and Azure Data Box

**Azure Migrate** - service that helps migrate from on-prem to the cloud.

- provides:
- unified migration platform - a single portal to start, run and track your migration to azure
- range of tools - tools used for assessment and migration (Azure Migration: discovery and assessment, Azure Migration: server migration)
- assessment and migration - you can assess and migrate your on-prem infrastructure to azure.
  - discovery and assessment - find on-prem servers running Hyper-V, VMware and physical servers in prep for migrating
  - server migration - migrate Hyper-V, VMware and physical server VMs, as well as public cloud VMs to Azure.
  - data migration assistant - stand-alone tool for assessing SQL servers - helps find issues for migration of SQL servers.
  - database migration service - migrate on-prem databases to azure VMs running SQL server, SQL database, or SQL managed instances
  - app service migration assistant - standalone tool to assess on-prem websites for migration to azure app service
  - data box - use azure data box products to move large amounts of offline data to azure.

  **Azure Data Box** - a physical migration service to transfer large amounts of data.
- Microsoft ships you a SAN with 80TB of space - you can either put data on or take data off of it.
- It is ideal for transferring data larger than 40TB in scenarios with little to no network connectivity.
- Can be one-time, periodic, or an initial bulk transfer followed by periodic transfers.
  - importing data:
  - onetime migration
  - migrating a media library from offline storage
  - moving historical data to azure for in-depth analysis
  - exporting data:
  - disaster recovery

- migrate back to on-premises or to another cloud service provider

# !Describe Azure Identity, Access, and Security

## !Describe directory services in Azure, including Microsoft Entra ID and Microsoft Entra Domain Services

**Microsoft Entra ID (AD in cloud)** - directory service that allows access to Microsoft Cloud applications - essentially the cloud version of Active Directory.

- can be linked with Active Directory to help monitor suspicious sign-in (free)
  **Who uses Entra ID?**
  - IT Admins - Entra ID can be used to control access to applications and resources based on their business requirements (active directory style)
  - App Developers - developers can use Entra ID to add standards-based functionality that allows for SSO - or allowing an app to work with a user's existing credentials
  - Users - users can manage their identities and do things like self-service password resets.
  - Online service subscribers - M365, Azure, etc. subscribers are already using Microsoft Entra ID to authenticate.
  **What does Entra ID do**
  - authentication - verifies identity to access applications/resources
  - SSO - allows users to use just one sign on for everything
  - application management - manage cloud and on-prem apps based on user permissions
  - device management - Entra ID supports registration of devices, which allows for device-based **Conditional Access** policies to restrict access attempts to only those coming from known devices, regardless of account.
  **Microsoft Entra Domain Services (DS in cloud)** - a service that allows for domain services such as domain join, group policy, LDAP, and Kerberos/NTLM authentication.
- lets you get the benefit of domain services without needing to deploy/manage DCs in the cloud.
- also lets you use legacy applications in the cloud
  pick a domain name for your cloud domain, it auto deploys 2 DCs in a **replica set**. - you never have to touch the DCs
  - the managed domain pulls from Entra ID, but does not push back to it.
  - in a hybrid environment Entra Connect syncs identity information with Entra ID to the managed domain (AD DS > Cloud, but not the other way around.)

## !Describe authentication methods in Azure, including Single Sign-On (SSO), Multi-Factor Authentication (MFA), and Passwordless

**Single Sign-On (SSO)** - allows a user to sign in a single time and use that credential to access multiple resources and applications from different providers.

- ties all access for a user to a single ID/Password combo - this makes it easier to change permissions, and to log in for the user.
- minimizes the number of passwords a user has to remember, which minimizes risk of credential-related security incidents.
  **Multi-Factor Authentication (MFA)** - requires two or more elements to fully authenticate.
- something the user knows - like a challenge question
- something the user has - like a code sent to a phone or email
- something the user is - like a biometric (face scan, or fingerprint)
  - no longer just username and password, now you need an extra thing - this adds a bunch of security.
  Microsoft Entra MFA - an app notification or phone call for MFA.
  **Passwordless Authentication** - skip needing the password in favor for using a registered / enrolled device.
- still uses MFA:
- windows hello for business - setup on a designated Windows PC - biometric / pin credential allows access to only the owner - allows access to corporate resources on-prem and in the cloud
- Microsoft Authenticator App - setup on an employee's phone - sends a notification to the app, and requires a biometric on the phone to continue
- FIDO2 security keys - (Fast IDentity Online) incorporates web authentication standard. - basically a physical device such as USB stick, or Bluetooth/NFC.

## !Describe external identities in Azure, including business-to-business (B2B) and business-to-customer (B2C)

**External Identities** - person, device, service, etc. that is outside of your organization.

- can be through a different organization's sign on - then your access to other organizations is managed (think of signing in using google to allow access to another app.)
  **Business to business (B2B) collaboration** - let users use their preferred identity to sign-in to your applications (SaaS apps, custom apps, Microsoft applications, etc.)
- collaboration users are represented in your directory typically as guest users.
  **B2B direct connect** - a mutual trust with another Microsoft Entra org for seamless collaboration.
- currently works with Teams shared channel.
  **Microsoft Entra Business to customer (B2C)** - publish modern SaaS apps or custom-

developed apps to consumers while using Azure AD B2C for identity and access management.

## !Describe Conditional access in Microsoft Entra ID

**Conditional Access** - a tool that can allow or deny access to resources based on identity signals

- the signals include
  - who the user is
  - where the user is
  - what device the user is using
  This can allow for a more granular approach to MFA - if at work, don't prompt for a second factor, if at a different location on a different device, do prompt.
  Can also block out sketchy locations entirely.
  Signal -> Decision (will it allow? will it block?) -> Enforcement (carries out the decision)
- when to use:
- when something requires MFA based on their role, location, or network (could enforce MFA for admins, but not for users)
- when you require users to use managed devices to sign in.
- blocking from untrusted locations

## !Describe Azure Role-Based access control (RBAC)

**Azure RBAC** - provides built-in (or custom) roles that describe common access rules for cloud resources

- you can assign individuals or groups to one or more roles - they will receive the associated permissions.
  RBAC is applied to a **scope** - which is a resource or set of resources that this access applies to.
  RBAC are passed down from parent scope to child scope
- RBAC is enforced on any action that's initiated against an Azure resource that passes through Azure Resource Manager (ARM.)
- RBAC uses an allow model which means that if one role grants you read permissions and another grants you write, you have both read and write.
  **Scope** includes:
- a management group
- a single subscription
- a resource group
- a single resource

## !Describe the concept of Zero Trust

**Zero Trust** - a security model that assumes worst case scenarios and protects resources.

- it acts as if every transaction is coming from an untrusted source
- uses these principles:
  - Verify explicitly - always authenticate and authorize based on all available data points
  - use least privilege access - limit user access with Just-In-Time and Just-Enough-Access, risk-based adaptive policies, and data protection.
  - assume breach - minimize blast radius and segment access. Verify end-to-end encryption. Use analytics to get visibility, drive threat detection, and improve defenses.
- basically - authenticate every time.

## !Describe the purpose of the defense-in-depth model

**Defense-in-depth** aims to protect information and prevent it from being stolen.

- uses a series of mechanisms to slow the advance of an attack that aims at acquiring unauthorized access to data.
  Layers:
- Physical Security - making sure the building is secure and random people can't plug things into computers/datacenters.
- Identity & Access - making sure access is only granted to what's needed, and that sign-in events are logged.
  - control access to infrastructure and change control, SSO and MFA, audit events and changes
- Perimeter - protects from network based attacks (like DDoS)
  - DDoS protection to filter large scale attacks out
  - Perimeter firewalls to identify and alert malicious attacks
- Network - limits communication to what's necessary, and limits resources through segmentation
  - deny by default
  - restrict inbound and limit outbound where appropriate
- Compute - makes sure there is no malware or unpatched exploits in the compute resources (VMs)
  - secure access to VMs, implement endpoint protection
- Application - makes sure applications are free of security vulnerabilities
  - store sensitive app secrets in a secure storage medium
- Data - this layer makes sure the data is properly secured - ensure confidentiality, integrity, and availability of the data.

### !Describe the purpose of Microsoft Defender for Cloud

**Microsoft Defender for Cloud** - a monitoring tool used for security posture management and threat protection.

- Can protect
  - cloud - natively integrated into azure
    - Azure PaaS services - can detect threats, perform anomaly detection
    - Azure Data services - can mark potential vulnerabilities across Azure SQL and storage, and recommend how to mitigate them
    - Networks - helps limit brute force attacks, reduces access to VMs by using Just-In-time access and disabling unused ports.
      also can protect AWS and GCP clouds
  - on-prem
    -
  - hybrid
    - uses Azure Arc and Defender for Cloud's enhanced security features
  - multi-cloud environments.
- Hardens resources, track your security posture, protect against cyber attacks, and streamlines security management.
- **Assess** - always watching and tracking vulnerabilities - through vulnerability scans
- **Secure** - hardens resources and services with Azure Security Benchmark - this can recommend security changes for your azure setup (does this by giving you a score, and telling you what to change)
- **Defend** - Detects and resolves threats to resources, workloads, and services
  - security alerts (give details of what is affected, how to fix, sometimes can give option to auto trigger logic) -
  - advanced threat protection - secures management ports of machines, and creates adaptive application controls for what apps are allowed to run on your machines

# !Describe Azure management and Governance (30-35%)

# !Describe Cost management in Azure

### !Describe Factors that can affect costs in Azure

**Resource Type** - the type, the settings and region will all have an impact on how much a resource costs.

**Consumption** - depending on how many resources you use and how much of them you use, you will pay more or less.
**Maintenance** - making sure that only the things that are supposed to be provisioned are, you can save money
**Geography** - Different locations have different prices, it's less expensive to move data within a region that to another region.

- network traffic - billing zones are a factor in determining the cost - some inbound bandwidth is free, outbound depends on where you are taking the information to.
  **Subscription Type** - a free subscription is cheaper than a different one - duh.
  **Azure Marketplace** - the more things you use from the Azure Marketplace the more you have to pay.

## !Compare the pricing calculator and the Total Cost of Ownership (TCO) Calculator

**Pricing Calculator** - Allows you to estimate the price of provisioning resources in Azure.

- think more VMs and SQL databases.
  **TCO Calculator** - Compares the costs of running an on-prem infrastructure compared to an Azure cloud infrastructure.
- think more migrating physical servers.

## !Describe cost management capabilities in Azure

**Cost Management** - allows you to quickly view Azure resource costs and automate them.

- Cost Analysis is a subset, allows you to view costs from various things quickly
- Cost Alerts can tell you about:
  - budget alerts - when spending reaches or exceeds predefined budget.
  - credit alerts - for organizations with Enterprise Agreements - when you use 90% and 100% of your Azure credit balance
  - department spending quota alerts - lets you know when a department hits a percentage of their quota.
  **Budgets** - a spending limit for Azure
- can be set based on: subscription, resource group, service type, or other.

## !Describe the purpose of tags

**Tags** are another way to organize your resources - they add on metadata that can help sort/organize

- resource management - let you locate resources that are associated with specific workloads/environments/business units/owners
- cost management and optimization - lets you report on costs, track budgets, forecast estimated costs
- operations management - allows you to set tiers on how important a service is for your business - create SLAs from there
- security - let you classify data by tag
- governance and regulatory compliance - can be used to identify resources that need to follow regulatory standards
- workload optimization and automation - can help visualize all of the resources that participate in complex deployments - shows what happens in automation
  Tags can be added, modified, or removed through PowerShell, Azure CLI, ARM templates, REST API, or Azure Portal.
  You can assign multiple tags to a resource.

# !Describe features and tools in Azure for governance and compliance

## !Describe the purpose of Microsoft Purview in Azure

**Microsoft Purview** - a tool that allows you to get a single unified view into your data.

- Brings insights about on-prem, multi-cloud, and SaaS data
  - Automated data discovery
  - Sensitive data classification
  - End-to-end data lineage
  **Risk and Compliance** - purview can manage and monitor M365, OneDrive, Exchange data
- Helps: protect sensitive data across clouds, apps and devices
- Identify data risks and manage regulatory compliance requirements
- Get started with regulatory compliance
  **Unified Data Governance** - enables you to manage your data stored in Azure, SQL and Hive databases, locally, and in other clouds.
- Helps: create an up-to-date map of your entire data estate - including data classification and lineage
- identify where sensitive data is stored in your esate
- create a secure environment for data consumers to find valuable data
- generate insights about how your data is stored and used
- manage access to the data in your estate securely and at scale

## !Describe the purpose of Azure Policy

**Azure Policy** - service that helps control or audit your resources - enforce different rules across resource configurations to ensure they stay compliant to corporate standards

- allows you to define both individual policies and groups of related policies (initiatives)
- highlights any resources that aren't compliant, and can prevent noncompliant resources from being created.
  Policies can be set at each level - resource, resource group, subscription, management group, scope.
- policies are inherited
- policies can automatically apply (like tags)
  Azure Policy comes with built-in policy and initiatives for:
- Storage
- Networking
- Compute
- Security Center
- Monitoring
  Resources can be tagged as an exception
  **Azure Policy Initiatives** - a way of grouping related policies together
- think of like GPOs

## !Describe the purpose of resource locks

**Resource Locks** prevent resources from being deleted or updated (depending on lock type).

- Resource locks can be applied to individual resources, resource groups, or even an entire subscription.
- Resource locks are inherited.
  **Types of Resource Locks**
- Delete - authorized users can still read and modify a resource, but they can't delete the resource
- ReadOnly - authorized users can read a resource, but can't delete or update the resource. (Similar to making all authorized users have read only role)
  Resource locks can be managed from Azure Portal (locks tab), PowerShell, Azure CLI, or from ARM template.
  To delete a resource lock - you must delete the lock, then you can remove the resource. (even if you are the owner, must remove lock first.)

# !Describe features and tools for managing and deploying Azure resources

## !Describe the Azure Portal

**Azure Portal** is a web based GUI that allows you to manage your Azure subscription.

- Build, manage, and monitor everything from simple web apps to complex cloud deployments
- Create custom dashboards for an organized view of resources
- Configure accessibility options for an optimal experience

## !Describe Azure Cloud Shell, including Azure command-Line interface (CLI) and Azure PowerShell

**Azure Cloud Shell** - browser-based shell tool that lets you use PowerShell or Azure CLI (Bash) to configure Azure resources.

- uses your azure credentials so it knows what permissions you have
  **Azure PowerShell** - a shell that allows use of cmdlets that call the Azure REST API to perform tasks in Azure.
- can be used to setup, teardown, and maintain a single resource or multiple connected resources
- deploy an entire infrastructure, which might contain hundreds of resources, from imperative code.
  **Azure Command-Line Interface (CLI)** - has the same function as Azure PowerShell, but has different syntax
- Installable, or can be used through the browser.
- can do the same orchestration as PowerShell

## !Describe the purpose of Azure Arc

**Azure Arc** - a centralized governance and management platform for multi-cloud and on-prem deployments.

- manage your entire environment together by projecting your existing non-Azure resources into ARM
- manage multi-cloud and hybrid VMs, Kubernetes clusters, and databases as if they are running in Azure
- use familiar Azure services and management capabilities, regardless of where they live

- continue using traditional ITOps while introducing DevOps practices to support new cloud and native patterns in your environment
- configure custom locations as an abstraction layer on top of Azure Arc-enabled Kubernetes clusters and cluster extensions.
  **What can Azure Arc do outside of Azure** - allows you to manage the following:
- Servers
- Kubernetes Clusters
- Azure Data Services
- SQL Servers
- VMs (preview)

## !Describe infrastructure as code (IaC)

**Infrastructure as Code** - using lines of code to manage your infrastructure.

- ARM templates and Bicep are two examples of IaC with ARM.

## !Describe Azure Resource Manager (ARM) and ARM templates

**Azure Resource Manager (ARM)** - the deployment and management service for Azure - the middleman between any Azure tools, APIs, or SDKs.

- ARM authenticates and authorizes requests and sends the request to designated Azure Service.
  benefits of **ARM**:
  - manage infrastructure through templates rather than scripts - uses declarative JSON files.
  - deploy, manage, and monitor all resources as one group
  - re-deploy solutions in a consistent fashion
  - deploy resources in the correct order by defining dependencies
  - use RBAC for access control - natively integrated
  - apply tags to resources
  - allow you to view costs based on tags on resources
  **ARM Templates** - a declarative JSON formatted template that can deploy resources.
- The code is verified before it is run
- The orchestration of resources is parallel - meaning all instances of a resource are created at the same time
- It can even run PowerShell/Bash scripts before or after resource deployment.
  benefits of **ARM templates**:
  - declarative syntax - you declare what you want to deploy, but don't need to write the actual programming commands to deploy resources.
  - repeatable results - templates are exact, you can setup and tear down repeatedly with the

same results.
- orchestration - deploys resources in the correct order, when possible deploys in parallel so they finish faster than serial deployments. all deployment is through one command, not multiple in a row
- modular files - you can break your templates into smaller, reusable components - you can also nest templates without issues.
- extensibility - you can add PowerShell or Bash to the template using deployment scripts - give you the ability to complete end-to-end environment setup in a single ARM template
**Bicep** - language that uses declarative syntax to deploy azure resources.

- a simpler, more concise version of ARM template.
- same benefits as ARM templates + support for all resource types and API versions

# Describe Monitoring Tools in Azure

## !Describe the purpose of Azure Advisor

**Azure Advisor** - gives recommendation to improve reliability, security, performance, and reduce costs - suggestions give you actions you can do now, postpone, or dismiss.

- Recommendations available through Azure Portal and API - can also setup notifications to alert you about new recommendations.
Split into 5 categories:
  - Reliability - ensure and improve continuity of your business-critical apps
  - Security - used to detect threats and vulnerabilities that may lead to security breaches
  - Performance - used to improve speed of applications
  - Operational Excellence - used to help you achieve process and workflow efficiency, resource manageability, and deployment best practices
  - Cost - used to optimize / reduce spending

## !Describe Azure Service Health

**Azure Status** - shows you status of global Azure, lets you know of any outages or other incidents.
**Service Health** - shows you the status of the Azure Services and Region you are using/in - affected by outages, or planned maintenance
**Resource Health** - shows your actual Azure resources - health of your individual cloud resources (such as a VM)
The three of them give a completed view of your Azure environment - alerts are also stored and accessible for later review.

- this can help find trends.

## !Describe Azure Monitor, including Log Analytics, Azure Monitor alerts, and Application Insights

**Azure Monitor** - a platform for collecting data on your resources, analyzing that data, visualizing that data then possibly acting on it (assuming the automation is in place)

- can pull data from various sources - then puts that data into a storage repository - then that data can be used for visualization and or automation
  **Log Analytics** - a tool that allows you to write and run log queries on data gathered from Azure Monitor.
- queries can be simple or complex containing visualizations
- anything to do with querying logs is through this guy
  **Azure Monitor Alerts** - automated way of detecting when a threshold has been crossed. - you set the alert conditions, then receive a notification when that alert happens.
- Azure Monitor Alerts uses action groups - a collection of notification and action preferences that are associated with one or multiple alerts.
- Azure Monitor, Service Health, and Azure Advisor all use action groups to notify you.
  **Application Insights** - a feature that monitors web applications - can monitor web apps that are on Azure, on-prem, or in a different cloud environment.
- can use the application insights agent, or an SDK in your application
- one it is running you can use it to view:
  - request rates, response times, and failure rates
  - dependency rates, response times, and failure rates to show if external services are slowing performance
  - page views and load performance reported by users' browsers
  - AJAX calls from web pages, including rates, response times, and failure rates
  - user and session counts
  - performance counters from windows or Linux server machines, such as CPU, memory, and network usage
  - can also be used as a check to see if an application is alive.