

***AZ-900 MICROSOFT AZURE FUNDAMENTALS***

The Azure fundamentals tests the foundational level knowledge of cloud services and how those services are provided with Microsoft Azure. The details in this document is based on the updated version of July 2023.

The three fundamentals required to acquire AZ-900 certification is :

* Cloud concepts
* Azure architecture and services.
* Azure management and governance.

***What is Cloud?***

Cloud is a virtual thing and it is not visible and cloud can also be referred as some one else’s computer. The cloud enables us the ability to rent the resources that are in demand.

The resources that include:

* Linux and windows servers
* Unlimited File storage
* Databases
* Message Queues
* Content Delivery Network
* Batch Processing Jobs.

***Cloud Concepts(25% - 30%)***

**1) Describe cloud computing?**

Cloud computing means delivering the services(storage,networking,servers,vms etc..) and compute power(windows os, linux,web etc.) together without any user’s active management is known as cloud computing.

Cloud is a virtual thing and it is not visible and cloud can also be referred as some one else’s computer. The cloud enables us the ability to rent the resources that are in demand.

The resources that include:

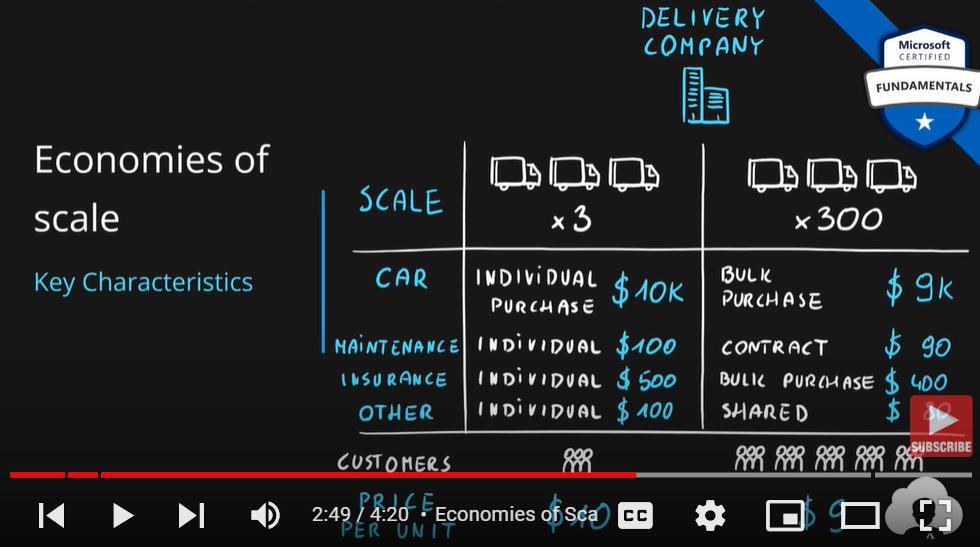
* Linux and windows servers
* Unlimited File storage
* Databases
* Message Queues
* Content Delivery Network
* Batch Processing Jobs.

***Characteristics of Cloud:***

1.**Scalability:** The ability of resource to scale. Scaling is a process of allocating and deallocating the resources. There are two types of Scaling

* ***Vertical Scaling:*** allocating or deallocating more space,diskstorage, CPU etc. to a resource. Allocation means ***scaling up*** and deallocating refers to ***scaling down.***
* ***Horizontal Scaling :*** Horizontal Scaling means adding more resources or removing the existing resources. ***Scale in*** means removing the existing resources and ***scale out*** means adding the resources to the group.

1. **Elasticity:** The ability of the cloud to scale dynamically is known as elasticity. There will be scenarios where there will be more users for the application and the number of users falls and we also know when there are more users using the application we need more resources and when the users count decreases eventually we also need to decrease the number of resources. The ability to automatically scale itself is known as elasticity.
2. **Agility:** The ability to act or react quickly is also known as ***“agility”***. Whenever we are required to provision the resources to an application there are two ways of doing that one is manually through interface and another is using scripting. Whatever it may be but allocating resources for the on-prem servers takes long response time(like days,months) when compared to requesting for resources in the cloud(in seconds or minutes).
3. **Fault Tolerance:** Fault tolerance is the ability to remain up and running during component or server failures.
4. **Disaster Recovery Management:** Incase of disasters we ensure there is a copy of data is replicated and stored in another azure location which can be recovered and ensuring no data loss happens.
5. **High availability:** availability is the ratio to the total uptime to the uptime + downtime.



The price per unit gets decreased to $9 from $10.

CapEx vs OpEx

Differences between Capital Expenditure and Operational Expenditure. In capital Expenditure we own the servers so we have to make huge amount of investment on infrastructure in order to setup all the hardware and as the time passes we all aware about the fact that hardware value get lowered every year which is also known as depreciation rate and the maintainence cost of the servers requires power supply, support staff, and have to handle the hardware replacements and failures. Whereas in the cloud all of these will be managed by the cloud service provider.

|  | Capital Expenditure | Operational Expenditure |
| --- | --- | --- |
| Up front cost | Significant | None |
| Ongoing cost | Low | Based on usage |
| Tax Deduction | Over time | Same year |
| Early Termination | No | Anytime |
| Maintenance | Significant | Low |
| Value over time | Lowers | No change |

What is a consumption-based model?

The consumption-based model is a **pricing model** used in the cloud so that customers are only charged **based on their resource usage**.

This model is characterized by

* **No associated upfront cost**
* **No wasted resources** as such *no charges are incurred for unused resources*\*. Unused in this case is different per service. For instance, blob storage that stores any data is considered to be used, as it consumes the storage space. Virtual Machines that are running consume CPU, memory and other resources even if there isn’t any traffic. Hence they are considered to be used and will incur charges.
* **Pay for what you need**
* **Stop paying when you don’t**

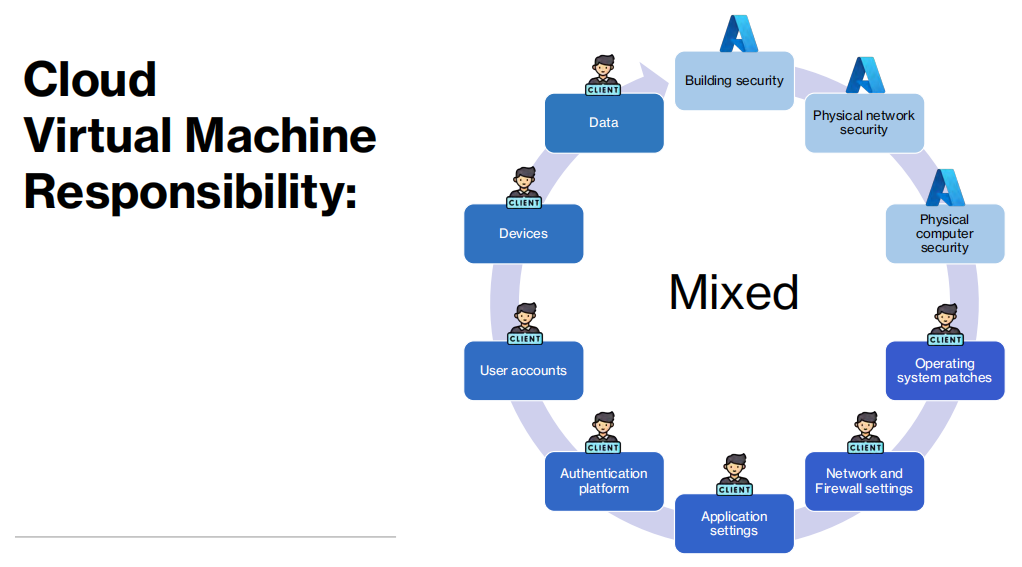
**Consumption** is the virtual metric used to calculate how much each resource (service) in Azure was used. Each service has many smaller metrics that track its consumption to offer best possible pricing model. Those metrics are tracked on very granular level.

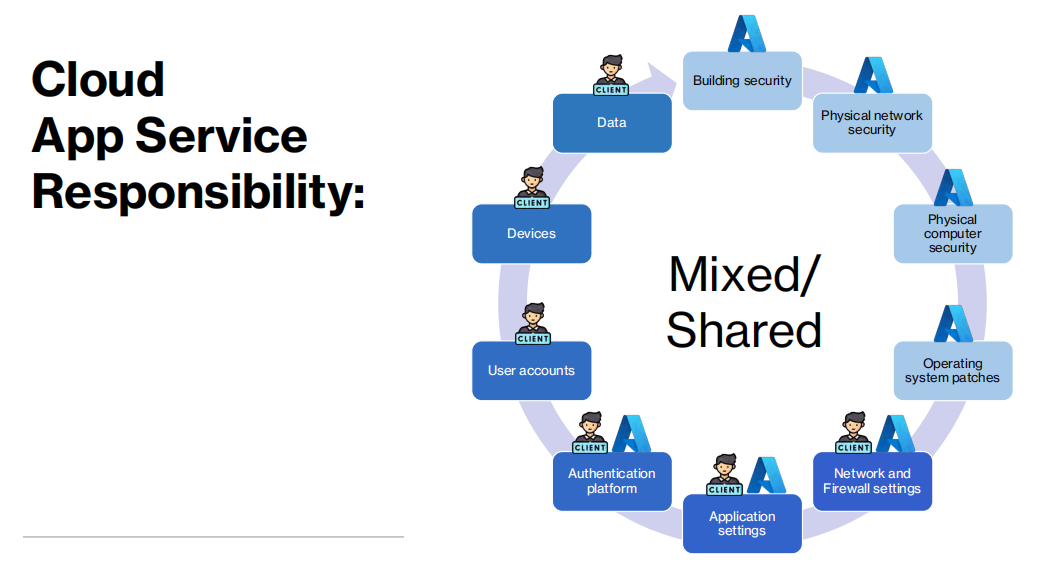
Some of the metrics on which the cloud bills us are

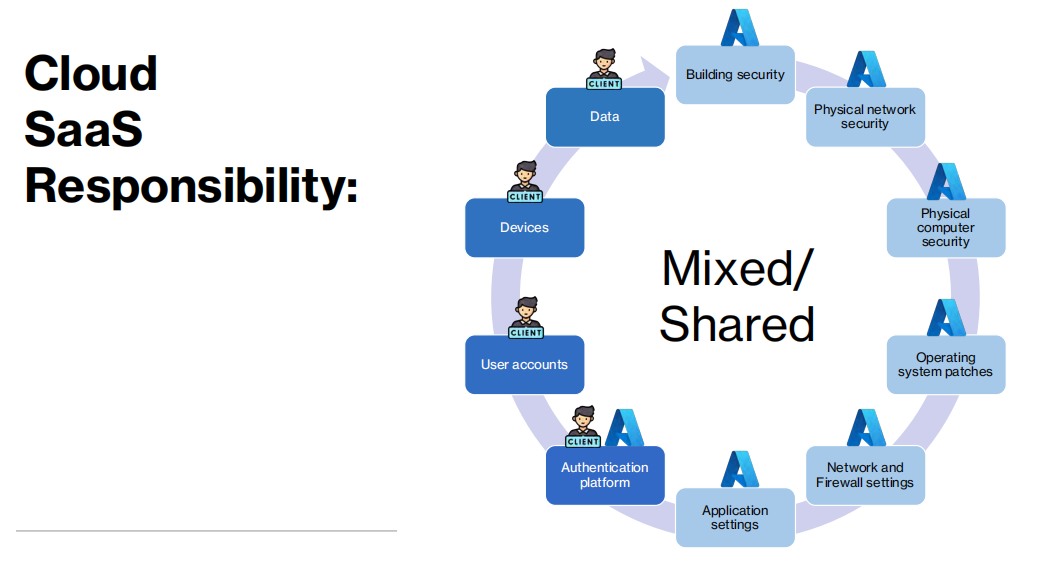
1. Compute 🡪 the power /size of the VM.
2. Storage attaced to the VM.
3. Networking.
4. The up-time or usage of the VM.

**2)Shared Responsibility Model**











Service Models responsibilities

**As a service** means which party will manage the particular layer and all the layers below.

* **Software** layer consists the application (application code and set) & the application data
* **Platform** layer means all the supporting software and the operating system required to host the application
* **Infrastructure** layer consists hardware the infrastructure and virtualization required to host the platform

| Layer | Layer |
| --- | --- |
| Application | Software |
| Data | Software |
| Runtime | Platform |
| Middleware | Platform |
| Operating System | Platform |
| Virtualization | Infrastructure |
| Servers | Infrastructure |
| Networking | Infrastructure |
| Storage | Infrastructure |

Responsibility Matrix

As such following table represents responsibilities

| Layer | On-Premises | IaaS | PaaS | SaaS |
| --- | --- | --- | --- | --- |
| Application | You | You | You | Cloud provider |
| Data | You | You | You | Cloud provider |
| Runtime | You | You | Cloud provider | Cloud provider |
| Middleware | You | You | Cloud provider | Cloud provider |
| Operating System | You | You | Cloud provider | Cloud provider |
| Virtualization | You | Cloud provider | Cloud provider | Cloud provider |
| Servers | You | Cloud provider | Cloud provider | Cloud provider |
| Networking | You | Cloud provider | Cloud provider | Cloud provider |
| Storage | You | Cloud provider | Cloud provider | Cloud provider |

## Cloud Deployment Model

**Cloud Deployment Model** is simple a separation which describes where are the company resources deployed. Whenever this is in public cloud provider environment or private datacenter.

Below table presents high level deployment model separation

| Layer | Cloud Provider | Own Datacenter |
| --- | --- | --- |
| **Public** | ✅ | ✖ |
| **Hybrid** | ✅ | ✅ |
| **Private** | ✖ | ✅ |

## Public Cloud

| Cloud Provider | Own Datacenter |
| --- | --- |
| ✅ | ✖ |

**Key Characteristics**

* Everything runs on cloud provider hardware
* No local hardware
* Some services share hardware with other customers

**Advantages**

* No CapEx (No initial investment)
* High Availability
* Agility
* Pay as you Go (PAYG) pricing
* No hardware maintenance
* No deep technical skills required

**Disadvantages**

* Not all security and compliance policies can be met
* No ownership over the physical infrastructure
* Rare specific scenarios can’t be done

## Private Cloud

| Cloud Provider | Own Datacenter |
| --- | --- |
| ✖ | ✅ |

**Key Characteristics**

* Everything runs on your own datacenter
* Self-service should be provided
* You maintain the hardware

**Advantages**

* Can support any scenario
* Total control over security and infrastructure
* Can meet any security and compliance policy

**Disadvantages**

* Initial investment is required (CapEx)
* Limited agility constrained by server capacity and team skills
* Very dependent on IT skills & expertise

## Hybrid Cloud

| Cloud Provider | Own Datacenter |
| --- | --- |
| ✅ | ✅ |

**Key Characteristics**

* Combines both Public & Private cloud

**Advantages**

* Great flexibility
* You can run any legacy apps in private cloud
* Can utilize existing infrastructure
* Meet any security& compliance requirements
* Can take advantage of all public cloud benefits

**Disadvantages**

* Can be more expensive
* Complicated to manage due to larger landscape
* Most dependent on IT skills & expertise from all three models

## Data Center

* **Physical facility**
* **Hosting for** group of networked **servers**
* Own **power**, **cooling** & **networking** infrastructure

## Region

* **Geographical area** on the planet
* **One but usually more datacenters** connected with **low-latency network** (<2 milliseconds)
* **Location** for your services
* Some services are **available only in certain regions**
* Some services are **global services**, as such are not assigned/deployed in specific region
* Globally available with **50+ regions**
* Special **government regions** (US DoD Central, US Gov Virginia, etc.)
* Special **partnered regions** (China East, China North)

## Availability Zone

* **Regional feature**
* Grouping of **physically separate** facilities
* Designed to **protect from data center failures**
* If zone goes down **others continue working**
* Two service **categories**
  + **Zonal** services (Virtual Machines, Disks, etc.)
  + **Zone-redundant** services (SQL, Storage, etc.)
* **Not** **all** regions are **supported**
* **Supported** region has **three or more zones**
* A **zone** is **one or more data centers**

## Region Pair

* **Each region** is **paired** with another region making it a region pair
* Region **pairs are static** and cannot be chosen
* Each pair resides within the **same geography**\*
  + Exception is Brazil South
* **Physical isolation** with at least 300 miles distance (when possible)
* Some services have **platform-provided replication**
* **Planned updates** across the pairs
* **Data residency** maintained for disaster recovery

| Region Pair A | Region Pair B |
| --- | --- |
| East US | West US |
| UK West | UK South |
| North Europe (Ireland) | West Europe (Netherlands) |
| East Asia (Hong Kong) | Southeast Asia (Singapore) |

## Geographies

* **Discrete market**
* Typically **contains two or more regions**
* Ensures **data residency**, **sovereignty**, **resiliency**, and **compliance** requirements are met
* **Fault tolerant** to protect from region wide failures
* Broken up into areas
  + **Americas**,
  + **Europe**,
  + **Asia Pacific**,
  + **Middle** **East** and **Africa**
* Each **region belongs only to one Geography**

## Azure Resource

* Object **used to manage services** in Azure
* Represents **service lifecycle**
* Saved as **JSON definition**

## Resource Groups

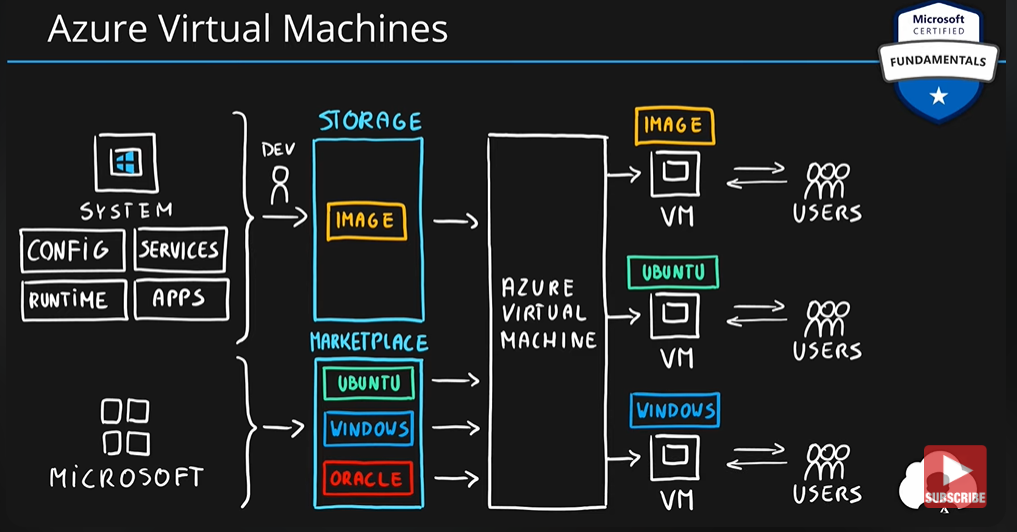
* **Grouping** of resources
* Holds **logically related** resources
* Typically organizing by
  + **Type**
  + **Lifecycle** (app, environment)
  + **Department**
  + **Billing**,
  + **Location** or
  + **combination of those**

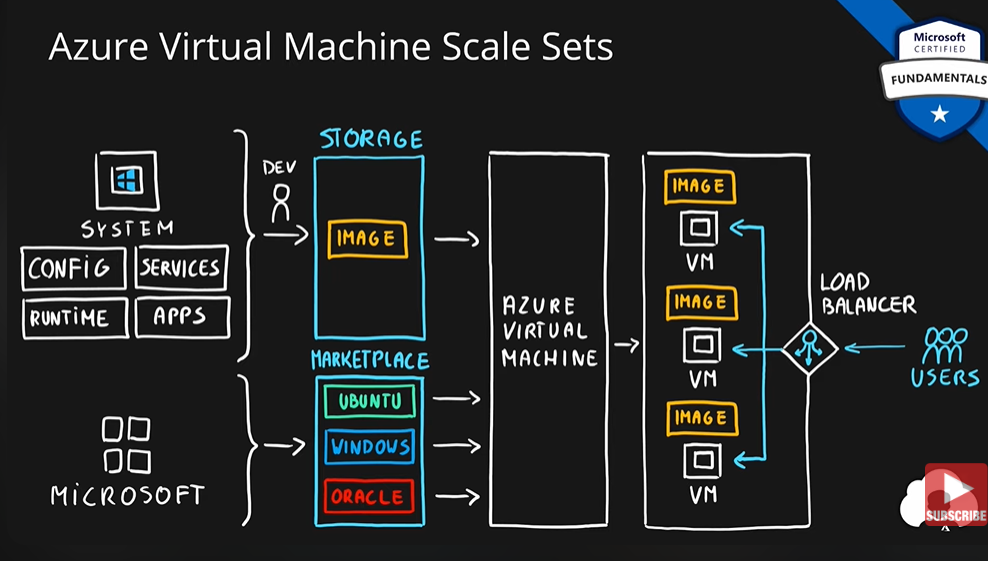
## Resource Manager

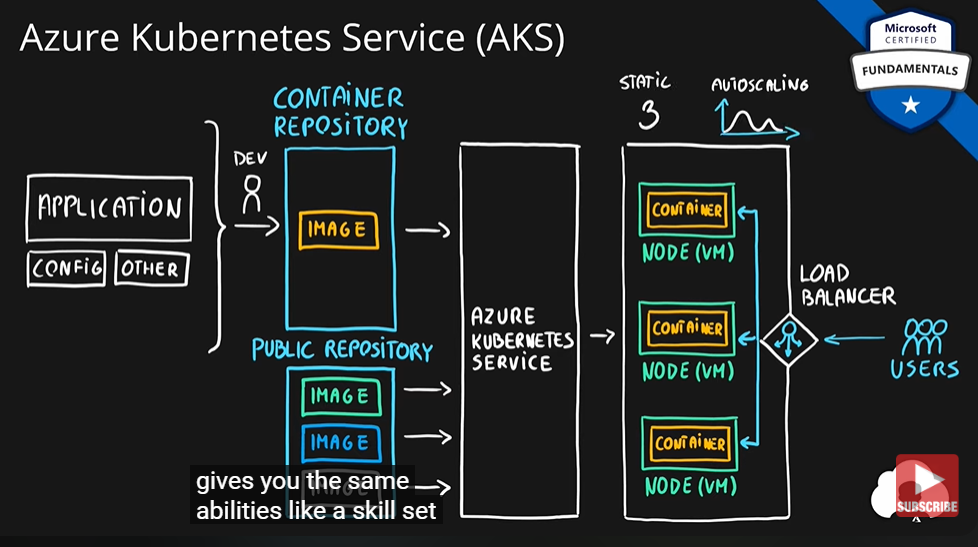
* **Management Layer** for all resources and resource groups
* **Unified** language
* **Controls** **access** and **resources**

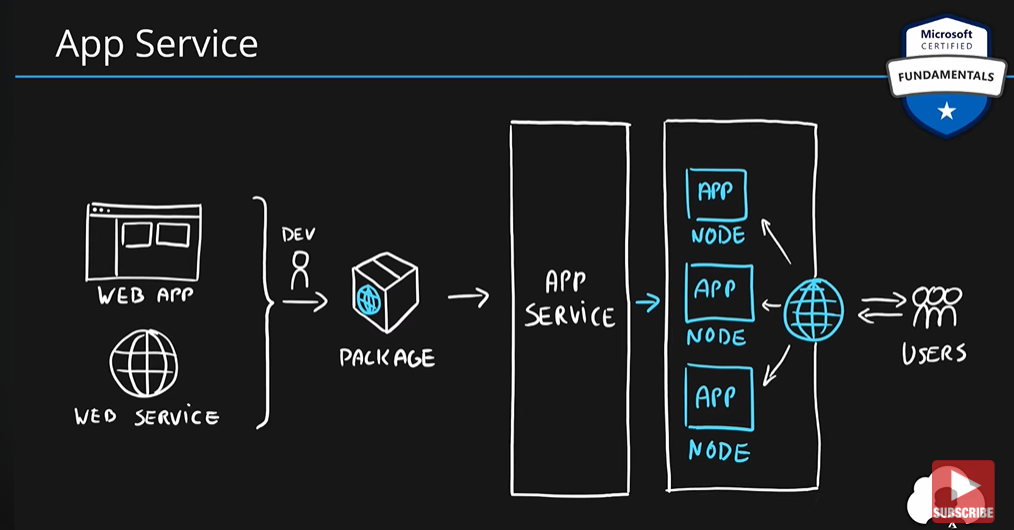
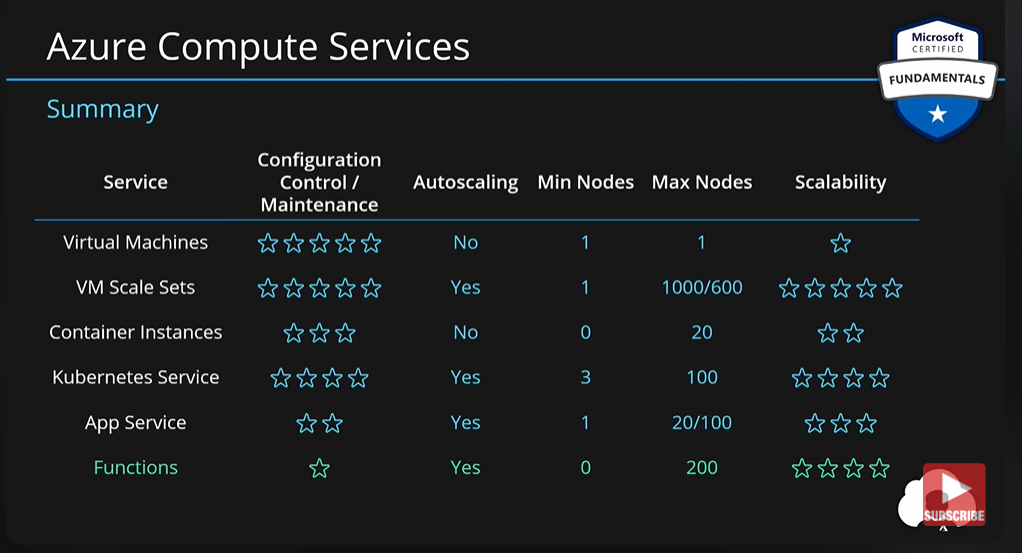
### **Additional Info**

* Each **resource must** be in one, and **only one resource group**
* Resource **groups have their own location** assigned
* Resources in the resource groups **can reside in a different locations**
* Resources **can be moved** between the resource groups
* Resource **groups can’t be nested**
* Organize based on your organization needs but consider
  + Billing
  + Security and access management
  + Application Lifecycle







### **Virtualization**

* Emulation of physical machines
* Different virtual hardware configuration per machine/app
* Different operating systems per machine/app
* Total separation of environments
  + file systems,
  + services,
  + ports,
  + middleware,
  + configuration

## Virtual Machines

* Infrastructure as a Service (IaaS)
* Total control over the operating system and the software
* Supports marketplace and custom images
* Best suited for
  + Custom software requiring custom system configuration
  + Lift-and-shift scenarios
* Can run any application/scenario
  + web apps & web services,
  + databases,
  + desktop applications,
  + jumpboxes,
  + gateways, etc.

## Virtual Machine Scale Sets

* Infrastructure as a Service (IaaS)
* Set of identical virtual machines
* Built-in auto scaling features
* Designed for manual and auto-scaled workloads like web services,\* batch processing, etc.

### **Containers**

* Use host’s operating system
* Emulate operating system (VMs emulate hardware)
* Lightweight (no O/S)
  + Development Effort
  + Maintenance
  + Compute & storage requirements
* Respond quicker to demand changes
* Designed for almost any scenario

## Azure Container Instances

* Simplest and fastest way to run a container in Azure
* Platform as a Service
* Serverless Containers
* Designed for
  + Small and simple web apps/services
  + Background jobs
  + Scheduled scripts

## Azure Kubernetes Service (AKS)

* Open-source container orchestration platform
* Platform as a Service
* Highly scalable and customizable
* Designed for high scale container deployments (anything really!)

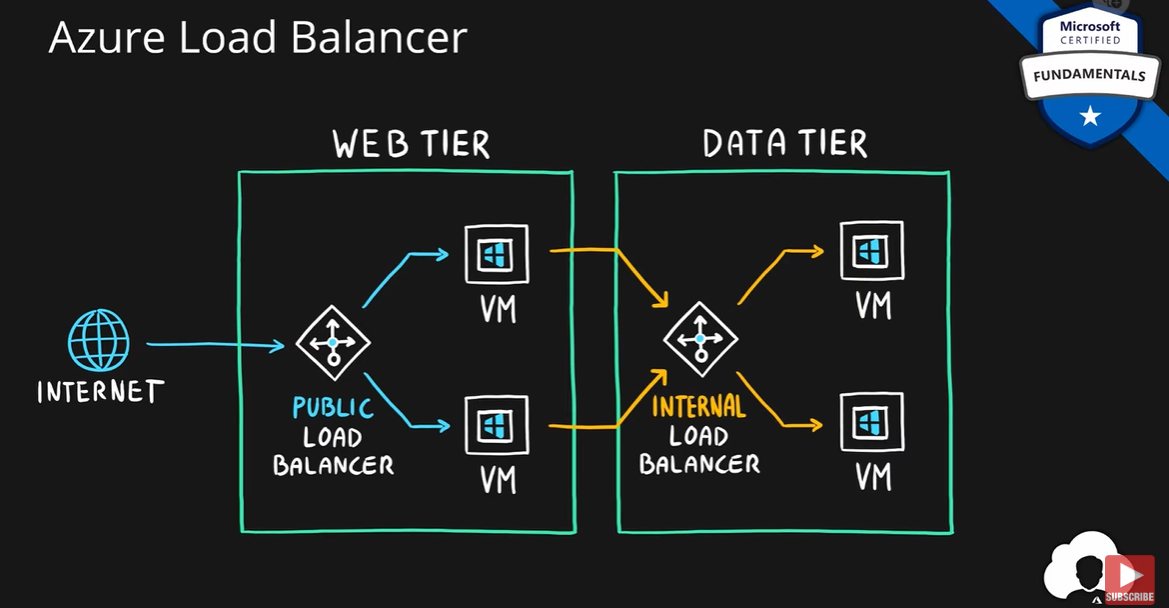
## App Service

* Designed as enterprise grade web application service
* Platform as a Service
* Supports multiple programming languages and containers

## Azure Functions (Function Apps)

* Platform as a Service
* Serverless
* Two hosting/pricing models
  + Consumption-based plan
  + Dedicated plan
* Designed for micro/nano-services

# **Summary**

* Virtual Machines (IaaS) - Custom software, custom requirements, very specialized, high degree of control
* VM Scale Sets (IaaS) - Auto-scaled workloads for VMs
* Container Instances (PaaS) - Simple container hosting, easy to start
* Kubernetes Service (PaaS) - Highly scalable and customizable \* container hosting platform
* App Services (PaaS) - Web applications, a lot of enterprise web \* hosting features, easy to start
* Functions (PaaS) (Function as a Service) (Serverless) - micro/nano-services, excellent consumption-based pricing, easy to start. 

## Azure Networking

* Connect cloud and on-premises
* On-premise networking functionality

### **Azure Virtual Network**

* Logically isolated networking components
* Segmented into one or more subnets
* Subnets are discrete sections
* Enable communication of resources with each-other, internet and on-premises
* Scoped to a single region
* VNet peering allow cross region communication
* Isolation, Segmentation, Communication, Filtering, Routing

### **Azure Load Balancer**

* Even traffic distribution
* Supports both inbound and outbound scenarios
* High-availability scenarios
* Both TCP (transmission control protocol) and UDP (user datagram protocol) applications
* Internal and External traffic
* Port Forwarding
* High scale with up to millions of flows

### **VPN Gateway**

* Specific type of virtual network gateway for on-premises to azure traffic over the public internet

### **Application Gateway**

* Web traffic load balancer
* Web application firewall
* Redirection
* Session affinity
* URL Routing
* SSL termination

### **Content Delivery Network**

* Define content
* Minimize latency
* POP (points of presence) with many locations

Data Types

* Structured - Data that can be represented using tables with very strict schema. Each row must follow defined schema. Some tables have defined relationships between them. Typically used in relational databases.
* Semi-structured - Data that can be represented using tables but without strict defined schema. Rows must only have unique key identifier.
* Unstructured - Any files in any format. Like binary files, application files, images, movies, etc.

Storage Account

* Group of services which include
  + blob storage,
  + queue storage,
  + table storage, and
  + file storage
* Used to store
  + files,
  + messages, and
  + semi-structured data
* Highly scalable (up to petabytes of data)
* Highly durable (99.999999999% - 11 nines, up to 16 nines)
* Cheapest per GB storage

Blob Storage

* BLOB – binary large object – file
* Designed for storage of files of any kind
* Three storage tiers
  + Hot – frequently accessed data
  + Cool – infrequently accessed data (lower availability, high durability)
  + Archive – rarely (if-ever) accessed data

Queue Storage

* Storage for small pieces of data (messages)
* Designed for scalable asynchronous processing

Table Storage

* Storage for semi-structured data (NoSQL)
  + No need for foreign joins, foreign keys, relationships or strict schema
  + Designed for fast access
* Many programming interfaces and SDKs

File Storage

* Storage for files accessed via shared drive protocols
* Designed to extend on-premise file shares or implement lift-and-shift scenarios

Disk Storage

* Disk emulation in the cloud
* Persistent storage for Virtual Machines
* Different
  + sizes,
  + types (SSD, HDD)
  + performance tiers
* Disk can be unmanaged or managed

## Data Types

* **Structured** - Data that can be represented using tables with very strict schema. Each row must follow defined schema. Some tables have defined relationships between them. Typically used in relational databases.
* **Semi-structured** - Data that can be represented using tables but without strict defined schema. Rows must only have unique key identifier.
* **Unstructured** - Any files in any format. Like binary files, application files, images, movies, etc.

## Cosmos DB

* Globally distributed NoSQL (semi-structured data) Database service
* Schema-less
* Multiple APIs (SQL, MongoDB, Cassandra, Gremlin, Table Storage)
* Designed for
  + Highly responsive (real time) applications with super low latency responses <10ms
  + Multi-regional applications

## SQL Database

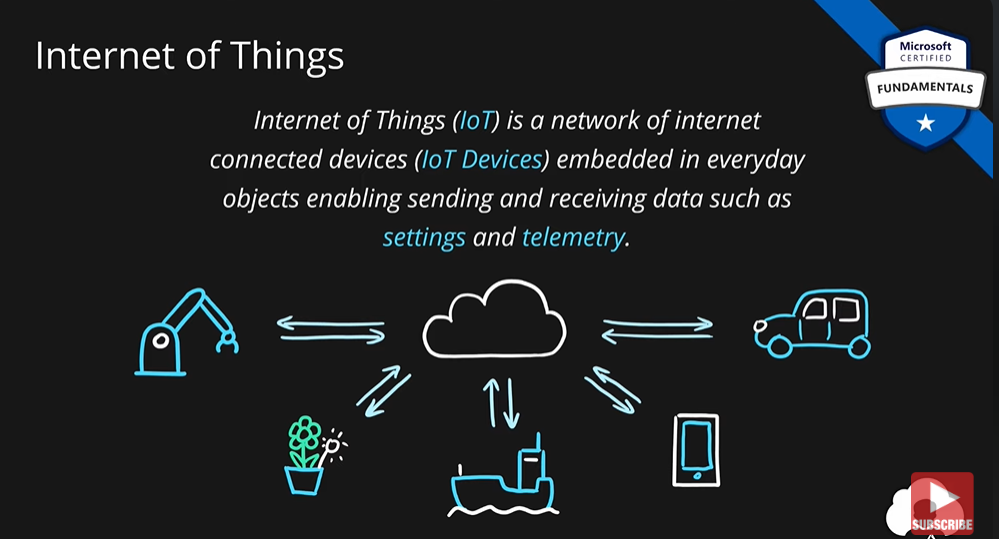
* **Relational database** service in the cloud (PaaS) (DBaaS - Database as a Service)
* **Structured data service** defined using schema and relationships
* **Rich Query Capabilitie**s (SQL)
* **High-performance**, reliable, fully managed and secure database for building - applications

## Azure SQL product family

* Azure **SQL Database** – Reliable relational database based on SQL Server
* Azure **Database for MySQL** – Azure SQL version for MySQL database engine
* Azure **Database for PostgreSQL** – Azure SQL version for PostgreSQL database engine
* Azure **SQL Managed Instance** – Fully fledged SQL Server managed by cloud provider
* Azure **SQL on VM** – Fully fledged SQL Server on IaaS
* Azure **SQL DW (Synapse)** – Massively Parallel Processing (MPP) version of SQL Server

Azure Marketplace

* Think of it like an “Azure Shop” where you purchase services and solutions for the Azure platform
* Each product is a template which contains one or multiple services
* Products are delivered by first and third-party vendors
* Solutions can leverage all service categories like IaaS, PaaS and SaaS



## What is Internet of Things?

Internet of Things (**IoT**) is a network of internet connected devices (**IoT Devices**) embedded in everyday objects enabling sending and receiving data such as **settings** and **telemetry**.

## Azure Iot Hub

* Managed service for bi-directional communication
* Platform as a Service (PaaS)
* Highly secure, scalable and reliable
* Integrates with a lot of Azure Services
* Programmable SDKs for popular languages (C, C#, Java, Python, Node.js)
* Multiple protocols (HTTPS, AMQP, MQTT)

## Azure IoT Central

* IoT App Platform - Software as a Service (SaaS)
* Industry specific app templates
* No deep technical knowledge required
* Service for connecting, management and monitoring IoT devices
* Highly secure, scalable and reliable
* Built on top of the IoT Hub service and 30+ other services

## Azure IoT Sphere

* Secure end-2-end IoT Solutions
  + Azure Sphere certified chips (microcontroller units - MCUs)
  + Azure Sphere OS based on Linux
  + Azure Security Service trusted device-to-cloud communication

## What is Big Data?

**Big Data** is a field of technology that helps with the **extraction**, **processing** and **analysis** of information that is **too large or complex** to be dealt with by traditional software.

### **The three V’s rule**

Big data typically has one of the following characteristics

* **Velocity** - how fast the data is coming in or how fast we are processing it
  + Batch
  + Periodic
  + Near Real Time
  + Real Time
* **Volume** - how much data we are processing
  + Megabytes
  + Gigabyte
  + Terabytes
  + Petabytes
* **Variety** - how structured/complex the data is
  + Tables
  + Databases
  + Photo, Audio
  + Video, Social Media

## Azure Synapse Analytics

* Big data analytics platform (PaaS)
* Multiple components
  + Spark
  + Synapse SQL
    - SQL pools (dedicated – pay for provisioned performance)
    - SQL on-demand (ad-hoc – pay for TB processed)
  + Synapse Pipelines (Data Factory – ETL)
  + Studio (unified experience)

## Azure HDInsight

* Flexible multi-purpose big data platform (PaaS)
* Multiple technologies supported (Hadoop, Spark, Kafka, HBase, Hive, Storm, Machine Learning)

## Azure Databricks

* Big data collaboration platform (PaaS)
* Unified workspace for notebook, cluster, data, access management and collaboration
* Based on Apache Spark
* Integrates very well with common Azure data services

### **What is Artificial Intelligence?**

**Artificial Intelligence** (**AI**) is the simulation of human intelligence & capabilities by computer software.

### **What is Machine Learning?**

**Machine Learning** is a subcategory of AI where a computer software is “**taught**” to **draw conclusions** and **make predictions** **from data**.

## Azure Machine Learning

* Cloud-based platform for creating, managing and publishing machine learning models
* Platform as a Service (PaaS)
* Machine Learning Workspace – top level resource
* Machine Learning Studio – web portal for end-2-end development
* Features
  + Notebooks – using Python and R
  + Automated ML – run multiple algorithms/parameters combinations, choose the best model
  + Designer – graphical interface for no-code development
  + Data & Compute – management of storage and compute resources
  + Pipelines – orchestrate model training, deployment and management tasks

### **What is Serverless?**

**Serverless computing** is cloud-hosted execution environment that allows customers to **run their applications** in the cloud while **completely abstracting underlying infrastructure**.

## Azure Functions

* Serverless coding platform (Functions as a Service, FaaS)
* Designed for nano-service architectures and event-based applications
* Scales up and down very quickly
* Highly scalable
* Supports popular languages and frameworks (.NET & .NET Core, Java, Node.js, Python, PowerShell, etc.)

## Azure Logic Apps

* Serverless enterprise integration service (PaaS)
* 200+ connectors for popular services
* Designed for orchestration of
  + business processes,
  + integration workflows for applications, data, systems and services
* No-code solution

## Azure Event Grid

* Fully managed serverless event routing service
* Uses publish-subscribe model
* Designed for event-based and near-real time applications
* Supports dozen of built-in events from most common Azure services

## What is DevOps?

**DevOps** is a set of practices that combine both development (**Dev**) and operations (**Ops**).

DevOps aims to **shorten the development life cycle** by providing **continuous integration** and **delivery** (CI/CD) capabilities while **ensuring high quality** of deliverables.

## Azure DevOps

* **Collection of services** for building solutions using DevOps practices
* Services included
  + **Boards** – tracking work
  + **Pipelines** – building CI/CD workflows (build, test and deploy apps)
  + **Repos** – code collaboration and versioning with Git
  + **Test Plans** – manual and exploratory testing
  + **Artifacts** – manage project deliverables
* Extensible with **Marketplace** – over 1000 of available apps
* Evolved from **TFS** (Team Foundation Server), through **VSTS** (Visual Studio Team Services)

## Azure DevTest Labs

* Service for creation of **sandbox environments** for developers/testers (PaaS)
* Quick setup of **self-managed virtual machines**
* **Preconfigured templates** for VMs
* Plenty of additional **artifacts** (tools, apps, custom actions)
* Lab **policies** (quotas, sizes, auto-shutdowns)
* **Share** and **automate** labs via custom images
* Premade plugins/API/tools for **CI/CD pipeline automation**

## Azure Portal

* Public web-based interface for management of Azure platform
* Designed for self-service
* Customizable
* Simple tasks

## Azure PowerShell

* PowerShell and module
* Designed for automation
* Multi-platform with PowerShell Core
* Simple to use
  + Connect-AzAccount – log into Azure
  + Get-AzResourceGroup – list resource groups
  + New-AzResourceGroup – create new resource group
  + New-AzVm – create virtual machine

## Azure CLI

* Command Line Interface for Azure
* Designed for automation
* Multi-platform (Python)
* Simple to use
  + az login – log into Azure
  + az group list – list resource groups
  + az group create – create new resource group
  + az vm create – create virtual machine
* Native OS terminal scripting

## Azure Cloud Shell

* Cloud-based scripting environment
* Completely free
* Supports both Azure PowerShell and Azure CLI
* Dozen of additional tools
* Multiple client interfaces
  + Azure Portal integration (portal.azure.com)
  + Shell Portal (shell.azure.com)
  + Visual Studio Code Extension
  + Windows Terminal
  + Azure Mobile App
  + Microsoft Docs integration

## Azure Advisor

* **Personalized consultant** service
* Designed to provide **recommendations** and **best practices** for
  + **Cost** (SKU sizes, idle services, reserved instances, etc.)
  + **Security** (MFA settings, vulnerability settings, agent installations, etc.)
  + **Reliability** (redundancy settings, soft delete on blobs, etc.)
  + **Performance** (SKU sizes, SDK versions, IO throttling, etc.)
  + **Operational** Excellence (service health, subscription limits, etc.)
* **Actionable** recommendations
* **Free**!

## Network Security Groups

* Designed to **filter traffic** to (inbound) and from (outbound) Azure resources located in - Azure Virtual Network
* Filtering controlled by **rules**
* Ability to have **multiple** inbound and outbound **rules**
* Rules are created by specifying
  + **Source**/**Destination** (IP addresses, service tags, application security groups)
  + **Protocol** (TCP, UDP, any)
  + **Port** (or Port Ranges, ex. 3389 – RDP, 22 – SSH, 80 HTTP, 443 HTTPS)
  + **Direction** (inbound or outbound)
  + **Priority** (order of evaluation)

## Application Security Groups

* Feature that allows **grouping of virtual machines** located in Azure virtual network
* Designed to **reduce** the **maintenance effort** (assign ASG instead of the explicit IP address)

## Routing

Process of finding/selecting a path for traffic in one or across multiple networks.

## User-defined Routes

* Custom (user-defined, static) routes (UDRs)
* Designed to override Azure’s default routing or add new routes
* Managed via Azure Route Table resource
* Associated with a zero or more Virtual Network subnets

Firewall

Firewall is a network security service that monitors and controls incoming and outgoing traffic.

## Identity

* A user with a username and password.
* Also applications or other servers with secret keys or certificates.
* The fact of being something or someone.

### **Authentication**

The process of **verification/assertion of identity**

### **Authorization**

The process of **ensuring** that only **authenticated identities** get **access to the resources** for which they have been granted access.

### **Access Management**

The process of **controlling**, **verifying**, **tracking** and **managing access** to authorized users and applications.

Azure Firewall

* Managed, cloud-based **firewall service** (PaaS, Firewall as a Service)
* Built-in **high availability**
* Highly **Scalable**
* **Inbound** & **outbound** traffic filtering rules
* Support for **FQDN** (Fully Qualified Domain Name), ex. microsoft.com
* Fully integrated with Azure monitor for logging and analytics

### **DoS - Denial of Service**

Cyber-attack with intent to cause temporary or indefinite **disruption of service**

### **DDoS - Distributed Denial of Service**

**DoS** attack that is originating **from multiple servers**

## Azure DDoS Protection

* **DDoS protection service** in Azure
* Designed to
  + **Detect malicious traffic** **and block it** while allowing legitimate users to connect
  + **Prevent additional costs** for auto-scaling environments
* Two tiers
  + **Basic** – automatically enabled for Azure platform
  + **Standard** – additional mitigation & monitoring capabilities for Azure Virtual Network resources
* Standard tier uses machine learning to **analyze traffic patterns** for better accuracy

## Azure Active Directory

* Identity and Access Management service in Azure
* Identities management – users, groups, applications
* Access management – subscriptions, resource groups, roles, role assignments, authentication & authorization settings, etc.
* Used by multiple Microsoft cloud platforms
  + Azure
  + Microsoft 365
  + Office 365
  + Live.com services (Skype, OneDrive, etc.)

### **Multi-factor Authentication (MFA)**

* Process of authentication using more than one factor (evidence) to prove identity
* Factor types
  + Knowledge Factor – “Something you know”, ex. password, pin
  + Possession Factor – “Something you have”, ex. phone, token, card, key
  + Physical Characteristic Factor – “Something you are”, ex. fingerprint, voice, face, eye iris
  + Location Factor – “Somewhere you are”, ex. GPS location
* Supported by Azure AD by default (simple on-off switch)

## Identity

* **Centralized**/**unified** infrastructure and platform **security management service**
* **Natively embedded** in Azure services
* **Integrated** with **Azure Advisor**
* Two tiers
  + **Free** (Azure Defender OFF) – included in all Azure services, provides continuous assessments, security score, and actionable security recommendations
  + **Paid** (Azure Defender ON) – hybrid security, threat protection alerts, vulnerability scanning, just in time (JIT) VM access, etc.

## Azure Key Vault

* **Managed service** for **securing sensitive information** (application/platform) (PaaS)
* **Secure storage service** for
  + **Keys**,
  + **Secrets** and
  + **Certificates**
* **Highly integrated** with other Azure services (VMs, Logic Apps, Data Factory, Web Apps, etc.)
* **Centralization**
* Access **monitoring** and **logging**

## What is a Role?

**Role** (role definition) is a **collection of actions** that **the assigned identity** will be able to perform.

Role definition is an answer to a question “**What** can be done?”

## What is a Security Principal?

**Security Principal** is an Azure object (identity) that   
can be assigned to a role (ex. users, groups or applications).

**Security Principal assignment** is an answer to a question “**Who** can do it?”

## What is a Scope?

**Scope** is one or more Azure resources that the access applies to.

**Scope assignment** is an answer to a question “**Where** can it be done?”

### **What is a Role Assignment?**

**Role assignment** is a combination of the **role definition**, **security principal** and **scope**.

## Azure Role-based Access Control (RBAC)

* Authorization system built on Azure Resource Manager (ARM)
* Designed for fine-grained access management of Azure Resources
* Role assignment is combination of
  + Role definition – list of permissions like create VM, delete SQL, assign permissions, etc.
  + Security Principal – user, group, service principal and managed identity and
  + Scope – resource, resource groups, subscription, management group
* Hierarchical
  + Management Groups > Subscriptions > Resource Groups > Resources
* Built-in and Custom roles are supported

## What is an Azure Resource Lock?

* Designed to **prevent accidental deletion** and/or **modification**
* Used in conjunction with RBAC
* Two types of locks
  + **Read-only** (**ReadOnly**) – only read actions are allowed
  + **Delete** (**CanNotDelete**) – all actions except delete are allowed
* Scopes are **hierarchical** (**inherited**)
  + Subscriptions > Resource Groups > Resources
* **Management Groups** can’t be locked
* Only **Owner** and **User Access Administrator** roles can manage locks (**built-in** roles)

Azure Resource Tags

* Tags are simple **Name** (key) - **Value** **pairs**
* Designed to help with **organization of Azure resources**
* Used for resource **governance**, **security**, **operations management**, **cost management**, **automation**, etc.
* Typical **tagging strategies**
  + **Functional** – mark by **function** ( ex: environment = production )
  + **Classification** – mark by **policies used** ( ex: classification = restricted )
  + **Finance**/**Accounting** – mark for **billing purposes** ( ex: department = finance )
  + **Partnership** – mark by **association of users/groups** ( ex: owner = adam )
* Applicable for **resources**, **resource groups** and **subscriptions**
* **NOT inherited** by default

Azure Policy

* Designed to help with resource **governance**, **security**, **compliance**, **cost management**, etc.
* **Policies** focus on **resource properties** (**RBAC** focused on **user actions**)
* Policy **definition** – Defines what should happen
  + Define the **condition** (if/else) and the **effect** (deny, audit, append, modify, etc.)
  + Examples include allowed *resource types*, *allowed locations*, *allowed SKUs*, *inherit resource tags*
* **Built-in** and **custom** policies are supported
* Policy **initiative** – a **group** of policy definitions
* Policy **assignment** – assignment of a policy definition/initiative to a scope
  + Scopes can be assigned to
    - management groups,
    - subscriptions,
    - resource groups, and
    - resources
* Policies allow for **exclusions of scopes**
* Checked during **resource creation** or **updates** and **existing ones with remediation tasks**

Azure Blueprints

* **Package** of various Azure components (**artifacts**)
  + **Resource Groups**
  + **ARM Templates**
  + **Policy Assignments**
  + **Role Assignments**
* **Centralized storage** for organizationally **approved design patterns**
* Blueprint **definition** – describing what should happen (reusable package)
* Blueprint **assignment** – describing where it should happen (package deployment)