Azure

Components:

* **Cloud characteristics:**
  + Cost-effective;
  + Scalable;
  + Elastic;
  + Current;
  + Reliable;
  + **Secure:**
    - **Compliance with:**
      * CJIS;
      * CSA STAR Certification;
      * GDPR;
      * EU Model Clauses;
      * HIPAA;
      * ISO and IEC;
      * MTCS Singapore;
      * SOC 1,2 and 3;
      * NIST and CSF;
      * UK Gov G-Cloud;
* **Economies of Scale:**
  + **Capital Expenditure (CapEx):**
    - Spending of money up front, and then deducting that expense from your bill over time, has a value that reduces over time;
    - **Computing costs:**
      * **Server costs:**
        + Hardware and the cost of supporting them;
      * **Storage costs:**
        + Storage hardware, the cost of supporting it;
        + Can be centralized or organized by tiers;
      * **Network costs:**
        + On-premise hardware components;
        + Cabling, WAN, Internet connections.
      * **Backup and archive costs:**
        + Cost to back up, copy and archive data;
      * **Organization continuity and disaster recovery costs:**
        + Server fault tolerance and redundancy, and how to recover from disaster and keep operating.
      * **Datacenter infrastructure costs:**
        + Electricity, floor space, cooling, maintenance;
      * **Technical personnel**:
        + Specific to on-premise datacenters;
  + **Operational Expenditure (OpEx):**
    - Spending money now and being billed for them now, can deduct this in the same year, you pay as you use it;
    - **Cloud computing costs:**
      * **Leasing software and customized features:**
        + Uses pay-per-use model requires active management;
        + Your responsible to minimize costs;
      * **Scaling charges bases on usage/demand instead of fixed hardware or capacity:**
        + Pay per, for ex, number of CPU usage time or users, RAM/s, IOPS/s and storage space;
      * **Billing at the user/organization level:**
        + User/org is billed for the services used, on a recurring basis;
        + Can scale, customize and provision pc resources.
  + **Benefits of CapEx:**
    - Plan expenses at start.
    - Example: when you need to predict the expenses before a project starts due to a limited budget.
  + **Benefits of OpEx:**
    - Demand and growth can be unpredictable.
    - Agile;
* **Cloud deployment models:**
  + **Public cloud:**
    - No local hardware (Ex: MRCSTF Azure);
    - **Advantages:**
      * High scalability/agility;
      * Pay-as-you-go;
      * Not responsible;
      * Less needed technical knowledge;
    - **Disadvantages:**
      * Some security/legal/business requirements;
      * Not owning the hardware and being able to manage;
  + **Private cloud:**
    - You create a cloud environment in your own datacenter and provide self-service access.
    - **Advantages:**
      * Ensure your configuration;
      * Control and responsibility over security;
      * Can meet strict security/legal/business requirements;
    - **Disadvantages:**
      * Initial CapEx costs for hardware;
      * Owning the equipment limits agility;
      * Requires IT skills;
  + **Hybrid cloud:**
    - Combines both public and private clouds.
    - **Advantages;**
      * Easier transitions between public and private clouds;
      * Flexibility in what runs where (local/cloud);
      * Take advantage on economies of scale;
      * Can meet your legal/security/business requirements if needed;
    - **Disadvantages:**
      * Can be more expensive than that a single model since it involves some CapEx;
      * Can be more complicated to manage and maintain;
* **Types of cloud services:**
  + You can combine these to suit your needs;
  + **IaaS (Infrastructure as a Service):**
    - Most flexible;
    - Complete control (shared responsibility model);
    - Rent based hardware pricing;
    - Provided and managed over the internet;
    - **Commonly used in:**
      * Migrating workloads;
      * Test and development;
      * Storage, backup and recovery.
  + **PaaS (Platform as a Service):**
    - To help create an app fast;
    - No need to install an OS, web server or system updates;
    - Pay-as-you-go basis
    - Accessed through secure internet;
    - **Commonly used in:**
      * **Development frameworks:**
        + Scalability, high-availability, and multi-tenant capability included;
      * Analytics or business intelligence.
  + **SasS (Software as a Service):**
    - Centrally hosted and managed for the end customer;
    - Based on subscription pay model;
    - Used on Office365, Skype, Dynamics CRM Online;
  + **Summary (IaaS > PaaS > SaaS):**
    - **Upfront Costs:**
      * Pay what’s consumed > Pay what’s consumed > Monthly/Annual Subscription;
    - **User Ownership:**
      * Responsible for purchase, installation, configuration, management of software, OS, middleware and apps > Responsible for the dev of their apps > Not responsible for anything basically;
    - **Cloud provider ownership:**
      * Responsible for the cloud infrastructure > Responsible for everything apart from the app > Responsible for provision, management, maintenance of the app;
* **Azure (AZ) Basics:** 
  + **Characteristics:**
    - Durable;
    - Secure;
    - Scalable;
    - Managed;
    - Accessible;
  + **Compute:**
    - **AZ VM:** OS Virtual Machine hosting;
    - **AZ VM Scale Sets:** scaling for OS VMs hosted in AZ;
    - **AZ Kubernetes Service:** enables management of a cluster of VMs that run containerized services;
    - **AZ Service Fabric:** distributed systems platform; runs in AZ or on -premises;
    - **AZ Batch:** managed service for parallel and high-performance computing apps;
    - **AZ Container Instances:** run containerized apps on AZ without providing servers or VMs;
    - **AZ Functions:** event driven, serverless compute service;
  + **Networking:**
    - **AZ Virtual Network:** connects VMs to VPNs;
    - **AZ Load Balancer:** balances inbound and outbound connections to apps/services;
    - **AZ App Gateway:** optimizes app server farm delivery, while increasing app security;
    - **AZ VPN Gateway:** accesses AZ Virtual Networks through VPNs;
    - **AZ DNS:** fast DNS responses and high domain availability;
    - **AZ Content Delivery Network:** high-bandwidth content to customers;
    - **AZ DDoS Protection:** protects from DDoS attacks;
    - **AZ Traffic Manager:** networks traffic across AZ regions;
    - **AZ ExpressRoute:** connects to AZ over dedicated secure connections;
    - **AZ Network Watcher:** monitors and diagnoses the network;
    - **AZ Firewall:** high-security, high-availability firewall, with unlimited scalability;
    - **AZ Virtual WAN:** creates a unified wide area network, connecting to local and remote files;
  + **Storage:**
    - **AZ Blob Storage:** for large objects, such as video or bitmaps;
    - **AZ File Storage:** file share that you can access and manage like a file server;
    - **AZ Queue Storage:** for queuing and delivering messages between apps;
    - **AZ Table Storage:** a NoSQL data storage;
  + **Mobile:**
    - Simple inclusion of SAP, Oracle, SharePoint, etc.;
    - Offline data synch;
    - Connectivity to on-premise data;
    - Broadcast push notifications;
    - Autoscaling;
  + **Databases:**
    - **AZ Cosmo DB:** global DB that support NoSQL options;
    - **AZ SQL DB:** RDB with full management, auto-scale, integral intelligence, and secure;
    - **AZ DB for MySQL:** RDB with full management, auto-scale, integral intelligence, and secure for MySQL;
    - **AZ DB for PostgreSQL:** RDB with full management, auto-scale, integral intelligence, and secure for PostgreSQL;
    - **SQL Server on VMs:** host enterprise SQL server apps in the cloud;
    - **AZ SQL Data Warehouse:** full management, secure at every scale with no extra cost;
    - **AZ Data Migration Service:** migrate your DBs to the cloud;
    - **AZ Cache for Redis:** caches frequent used and static data for app latency;
    - **AZ for MariaDB:** full management and scalable MariaDB;
  + **Web:**
    - **AZ App Service:** create cloud-based web apps;
    - **AZ Notification Hubs:** send push notifications to any platform from any backend;
    - **AZ API Management:** publish APIs to devs, partners and employees;
    - **AZ Search:** search service;
    - **Web Apps feature of AZ App Service:** create and deploy mission-critical web apps at scale;
    - **AZ SignalR Service:** real-time web functionalities;
  + **IoT (Internet of Things):**
    - **IoT Central:** managed IoT as a SaaS, easy to connect, monitor and manage IoT assets at scale;
    - **AZ IoT Hub:** secure messaging and monitoring between millions of IoT devices;
    - **IoT Edge:** push data analysis models onto IoT devices;
  + **Big Data:**
    - **AZ SQL Data Warehouse:** run analytics using EDW that leverages MPP to run complex queries over petabytes of data;
    - **AZ HDInsight:** process data with managed clusters of Hadoop cluster in cloud;
    - **AZ Databricks:** Apache Spark-based analytic service, can be integrated with other Big Data services;
  + **AI (Artificial Intelligence):**
    - **AZ Machine Learning Service:** cloud-based environment to make ML models, train them and scale them to the cloud;
    - **AZ Learning Studio:** simple UX/UI to use pre-built MI algorithms and data-handling modules;
  + **APIs:**
    - **Vision:** image-processing for images and videos;
    - **Speech:** convert audio to text, voice verification, speech recon;
    - **Knowledge mapping:** map info to solve tasks such as recommendations and semantic search;
    - **Bing search:** Bing search APIs;
    - **Natural Language Processing:** allows apps to process natural language;
  + **DevOps:**
    - **AZ DevOps:** dev collab tools, pipelines, GIT repositories, Kanban boards, automated and cloud-based load testing;
    - **AZ DevTest Labs:** create OS environments to test or demo apps directly from deployment pipelines;
* **Datacenters and regions:**
  + **Region:**
    - Area containing one or more datacenter;
    - Important for scalability, redundancy, and preserve data residency;
    - There are *special* regions;
  + **Geographies:**
    - Discrete market containing two or more regions;
    - Preserves data residency and compliance regulations;
    - Fault-tolerant to withstand complete region failure;
  + **Availability zones:**
    - One or more datacenters equipped with independent power, isolated boundaries;
    - High-speed networks;
    - Used for zonal services or zone-redundant services
  + **Region Pair:**
    - Regions are always combined with another region;
  + **Service-Level Agreements (SLAs):**
    - MCRSTF’s commitment to provide specific performance standards;
    - **Characteristics:**
      * Performance targets;
      * Uptime and connectivity guarantees;
      * Service credits;
    - You can have composite SLAs that provide higher or lower uptime values depending on your app architecture;
    - Able to add a queue to for higher cost and management requirements but less probability of failure;
    - You’re able to calc the probability of failure from an SLA;
    - **Must understand your app requirements to use SLAs efficiently:**
      * **Resiliency:**
        + Ability to recover from failures and continue to run;
        + Do Failure Mode Analisys (FMAs) on your apps to test resiliency;
    - **Cost and complexity vs high availability:**
      * Availability refers to the time that a system is functional and working;
    - **Considerations for defining app SLAs:**
      * If 99.99% requirements for performance, our AZ solution must be self-diagnosing and self-healing;
      * It’s difficult to respond to failures quick enough to meet 99.99% requirements;
      * Carefully define the time window for your SLAs;
* **AZ Accounts and Subscriptions:**
  + **AZ Account:**
    - Entity in an AZ Directory (AZ AD) or in AZ AD trusted directory;
    - Contains personal info (Name, email, contact preferences, card);
  + **AZ Subscription:**
    - Logical container to provide resources to AZ;
    - Holds details of your resources;
    - Associated with an AZ AD;
    - **Subscription types:**
      * **Free;**
        + $200 credit on first 30 days;
        + 25 products are always free;
      * **Pay-as-you-go;**
        + Monthly charges;
        + For a wide range of users;
      * **Enterprise Agreement;**
        + Enterprise-scale organization focused;
        + Provides flexible cloud service prices and licenses;
        + Some discounts;
      * **Student;**
        + $100 credit on first 12 months;
    - **When to use multiple Azure subscriptions in your account:**
      * Lets access control and billing occur at the sub level, not the account level;
      * Allows avoiding reaching sub limits, creating separate environments or isolating data;
    - **Billing:**
      * A bill is generated per sub on a monthly basis;
      * Can be analyzed in an AZ Portal;
      * You can transfer AZ subs between accounts;
  + **Authentication with AZ AD:**
    - AZ AD is partitioned into separate tenants;
    - Tenants can be owned by orgs, individuals, teams, companies;
    - AZ AD tenants and subs have a one-to-many trust relationship;
  + **AZ support options:**
    - Billing and subs support;
    - AZ products and services and documentation;
    - Online self-help documentation;
    - Whitepaper;
    - Community support forum;
  + **Paid support plans;**
    - **MCRSFT offers four paid AZ supports for customers who require technical and operational support:**
      * Scope \*;
      * Technical support \*;
      * Case severity / response times \*;
      * Architecture support \*;
      * Operations support;
      * Training;
      * Proactive guidance;
      * Launch support;
    - Some plans are only available depending of your circumstances;
    - **Other support options:**
      * AZ Knowledge Center;
      * MSDN Forums;
      * Stack Overflow;
      * Server Fault;
      * AZ Feedback Forums;
      * AZ Twitter;
* **AZ management options:**
  + **AZ tools:**
    - **AZ portal (using GUI);**
      * Public website;
      * For creating, managing and monitoring AZ services;
      * No automation, time consuming and error-prone for complex tasks;
    - **AZ PowerShell**
      * Enables connection to AZ subscription and to manage resources;
      * Able to create admin scripts and use automation tools;
    - **AZ CLI**
      * Cross platform command line program;
      * Executes admin commands on AZ resources;
    - **AZ Cloud shell (web-based command-line interface);**
      * Browser shell;
      * Can use Bash or PowerShell;
      * **Dev tools, editors, and other tools:**
        + .Net Core, Python, Java, Node.js, Go;
        + code, vim, nam, emac;
        + git, maven, make, npm; AZ SDKS, APIs, etc.
    - **AZ mobile app;**
      * Allows to quick access, management and monitoring dorm your phone/tablet;
* **AZ computation:**
  + **Common techniques to perform compute in AZ:**
    - Virtual Machines;
    - AZ App Service;
    - Serverless Computing;
    - Containers**;**
    - **Virtual Machines:**
      * AZ VMs let you create and use VMs in the cloud;
      * Provides IaaS;
      * **Are an ideal choice when you need:**
        + Total control over the OS;
        + The ability to run custom software, or;
        + To use custom hosting configs;
      * **When to use VMs:**
        + During testing and development;
        + Running apps in the cloud;
        + When extending your datacenter to the cloud;
        + During disaster recovery;
      * **Scaling VMs in AZ:**
        + **Using:**

Availability sets;

Virtual Machine Scale Sets;

Azure Batch;

* + - * + **Availability set:**

Logical grouping of two or more VMs;

Used for planned maintenance, when the AZ fabric that hosts VMs is updated by MCRSTF;

Used for unplanned maintenance, when there’s hardware failure;

You get 1-3 fault domains with dedicated server racks and;

5-20 logical update domains;

* + - * + **Virtual Machine Scale Sets:**

Lets you create and manage a group of identical, load balanced VMs;

Allows for central management, configuration, and updates;

* + - * + **AZ Batch:**

Large-scale job scheduling and compute management;

**Batch can:**

Start a pool of compute VMs for you;

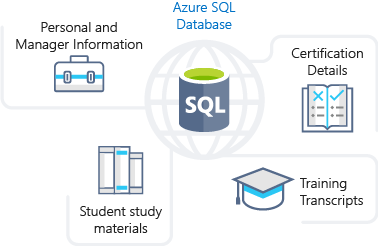
Install apps and staging data;

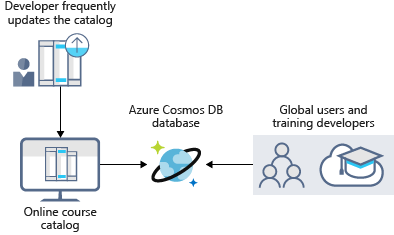
Runs jobs with as many tasks as you have;

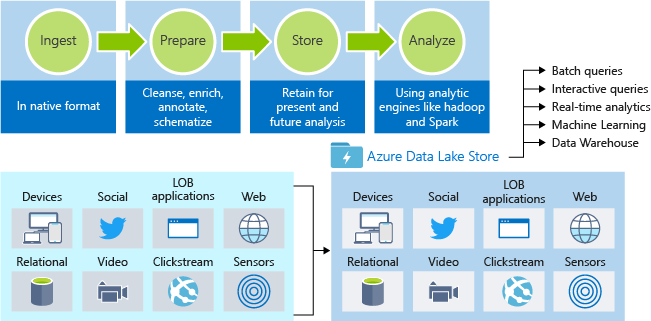
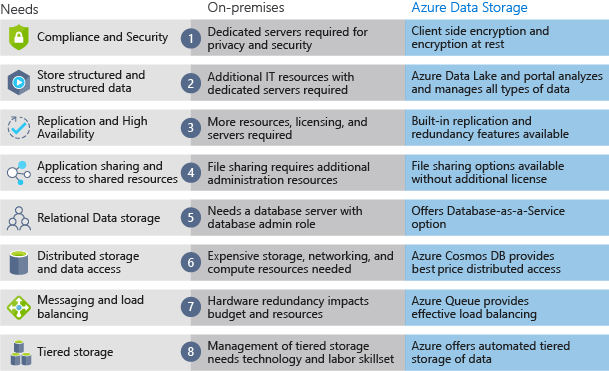
Identifies failures;

Requeues work;

Scales down the pool as work completes;

* + - **Containers:**
      * To run multiple instances of an app on a single host machine;
      * It’s a modified runtime environment built on top of a host OS that executes your app;
      * Doesn’t use virtualization;
      * **Dockers:**
        + AZ Container Instances (ACI);
        + AZ Kubernetes Services (AKS);
      * **AZ Container Instances:**
        + Fastest and simplest way to run a container in AZ;
        + PaaS that allows uploading of containers and execution with automatic elastic scale;
      * **AZ Kubernetes Services:**
        + Complete orchestration service with distributed architecture with multiple containers;
      * **Using containers in your solutions:**
        + Containers are often used to create solutions using microservices;
      * **Migrating apps to containers:**
        + Do this and run them with AKS;
        + Control access via AZ AD and SLAs via OSBA (Open Service Broker);
* **AZ App Service:**
  + Allows you to build and host web apps, background jobs, mobile backends, RESTful APIs without having to manage the infrastructure;
  + Automatic scaling and high availability;
  + PaaS;
  + **App Service Cost:**
    - You pay for the AZ compute resources your app uses based on your App Service Plan;
  + **Types of web apps:** 
    - **Web apps:**
      * Full-support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, Python; Use either Windows or Linux;
    - **API apps:**
      * Can build REST-based Web APIs, full Swagger support;
    - **Web Job apps:**
      * Allows running of a program or script;
      * Often used to run background tasks that are part of your app logic;
    - **Mobile apps:**
      * Quick builds of back-end of iOS and Android;
      * Store app data in cloud-based SQL DBs;
      * Provide authentication;
      * Send push notifications;
      * Execute custom back-end logic in C# or Node.js;
      * SDK support for native iOS and Android, Xamarin, and React Native apps;
* **Serverless computing:**
  + **Characteristics:**
    - **Abstraction of servers:**
      * No need to reserve servers;
      * Serverless architecture, simply deploy code with high availability;
      * No need for management;
    - **Event-driven scale:**
      * For workloads that respond to incoming events;
      * Timers, HTTP, queues and more;
      * Define a function which contains metadata about its triggers and bindings;
    - **Micro-billing:**
      * Pay only for the time your code runs;
  + **Implementations:**
    - **AZ Functions:** 
      * Can execute code in almost any modern language;
      * Code first approach;
      * Used when you need to perform work in response to an event;
      * Scale automatically,
      * Uses a function-based approach, which runs your code when it’s triggered and automatically deallocates resources when the function is finished;
      * Can be stateless or stateful;
      * Runs locally or in the cloud;
    - **AZ Logic Apps:**
      * Web-based designer, can execute logic triggered by AZ Services without writing of code, designer first approach;
      * Execute workflows designed to automate business scenarios and built from predefined logic;
      * Workflows are created with AZ Portal or VS Code;
      * Workflows are JSON;
      * Workflows provide over 200 connectors;
      * Stateful
      * Runs only in the cloud;
* **Benefits of using Azure to store data:**
  + Automated backup and recovery;
  + Replication across the globe;
  + Support for data analytics;
  + Encryption capabilities;
  + Multiple data types;
  + Data storage in virtual disks;
  + Storage tiers;
* **Types of data:**
  + **Structured data:**
    - Data that adheres to a schema;
    - Can be in a database with rows and columns;
    - Relies on keys to indicate how;
    - Relational data;
  + **Semi-structured data:**
    - Semi-structured data uses tags or keys that organize and provide hierarchy for the data;
    - Non-relational / NoSQL data;
  + **Unstructured data:**
    - No designated structure;
    - A blob can hold PDFs, a JPG, a JSON, etc.;
* **How Azure data storage can meet your business storage needs:**
  + **AZ SQL Database:**
    - It’s a relational database as a service (DaaS);
    - You can migrate existing SQL Server databases with minimal downtime using AZ DB Migration Service;
  + **AZ Cosmos DB:**
    - Supports schema-less data that lets you build highly responsive and Always On apps;



* + **AZ Blob Storage:**
    - Unstructured, no data type restrictions;
    - Highly scalable and good connection to apps;
  + **AZ Data Lake Storage:**
    - Allows to perform analytics on your data usage and prepare reports;
    - Stores both unstructured and structured data;
    - Combines the scalability and cost benefits of object storage with the reliability and performance of the Big Data file system capabilities;
  + **AZ Files:**
    - Fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) protocol;
    - Can be mounted by cloud or on-premise deployment from any OS;
    - Apps running in AZ VMs or cloud services can mount a file storage share to access file data;
    - Used to share files, diagnose data or share data;
  + **AZ Queue:**
    - For storing large number of messages accessible anywhere;
    - To build flexible apps and separate functions for better durability across large workloads;
    - Provides asynchronous message queuing;
    - **Can use to:**
      * Create a backlog of work and to pass messages between different Azure web servers;
      * Distribute load among different infrastructure and manage outbursts of traffic;
      * Build resilience against component failure when multiple users;
  + **Disk Storage:**
    - Provides disks for VMs, apps, etc.;
    - Come in different sizes and performance levels, from solid-state drives (SSDs) to traditional spinning hard disk drives (HDDs);
  + **Storage Tiers:**
    - Hot storage – for storing frequently accessed data;
    - Cool storage – for data that are infrequently accessed and stored for at least 30 days;
    - Archive storage tier – for data rarely accessed and stored for at least 180 days with flexible latency requirements;
  + **Encryption and replication:** 
    - **AZ Storage Service Encryption (SSE):**
      * For data at rest;
      * Encrypts your data before storing it and decrypts it before retrieving it;
    - **Client-Side Encryption:**
      * Where the data is already encrypted by the client libraries.
    - **Replication for storage availability:**
      * Set up when you create an account;
      * Ensures your data is durable and always available;
* **Comparison between Azure data storage and on-premises storage:**
  + **Cost-effectiveness:**
    - Up-front for on-premise (CapEx) VS Pay-as-you-go for AZ data storage (OpEx);
  + **Reliability:**
    - More options for AZ Storage;
  + **Agility:**
    - Way more agile AZ Storage than on-premise storage;
* **Deploy your site to Azure:**
  + **Using a N-tier architecture:**
    - Divides an app into two or more logical tiers;
    - Higher tier can access lower tiers but not the other way around;
    - For example, with 3 tiers it could have:
      * The web tier;
      * The app tier;
      * The data tier;
  + **A Virtual Network:**
    - A logically isolated network on Azure;
    - Allows Azure resources to communicate with each other securely;
    - Scoped to a region;
    - Can be segmented into subnets;
  + **A Network security work:**
    - Allows or denies inbound traffic to your Azure resources;
    - Like a cloud-level firewall;
  + **Scale with AZ Load Balancer:**
    - **Availability:**
      * When five nines (99.999) means it’s the closest to perfect;
    - **Resiliency:**
      * Systems ability to stay operational;
    - **Load balancer:**
      * Distributes traffic evenly among each system in a pool;
    - **AZ Load Balancer:**
      * Help take care of maintenance;
      * Supports inbound and outbound scenarios;
      * Provides low latency;
      * High throughput;
      * Scales up to all TCP and UDP apps;
    - **AZ App Gateway:**
      * If you have HTTP traffic, AZ App Gateway may be better;
      * Load balancer designer for web apps;
      * Uses AZ Load Balancer at the TCP;
      * URL-based routing rules (OSI layer 7);
    - **AZ App Gateway Benefits:**
      * Cookie affinity;
      * SSL termination;
      * Web app firewall;
      * URL rule-based routes;
      * Rewrite HTTP headers;
  + **Content Delivery Network (CDN):**
    - Distributed network servers;
    - To get content to users in their region, minimizing latency;
  + **Domain Name System (DNS):**
    - Way to map user-friendly to their IPs;
    - You can use AZ DNS;
* **Manage Latency with AZ Traffic Manager:**