**Agenda**

**Part 2: Describe Azure Core Services**

* + Overview of Azure Compute Services
  + Explore Azure Storage Services
  + Explore Azure Database Services
  + Explore Azure Network Services

**Part 2: Overview of Azure Compute Services**

**Azure Compute Services**

|  |  |  |
| --- | --- | --- |
| **Compute Service Name** | **Symbol** | **Description** |
| Azure Virtual Machines |  | * Software emulations of physical computers * VMs host an operating system, you can install and run software * Infrastructure as a service (IaaS) * VMs are best choice whenever you need over an operating system and environment |
| Virtual Machine Scale Sets |  | * Deploy and manage a set of identical VMs * With all VMs configured the same, virtual machine scale sets are designed to support true auto scaling |
| Containers and Kubernetes |  | * Deploy and manage containers * Containers are lightweight, virtualized application environments |
| App Service |  | * Quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform * PaaS Service |
| Functions |  | * Serverless computing * Concerned only about the code and not the underlying platform or infrastructure * Trigger it in response to an event timer, or message from another Azure service |

**When to use Azure Virtual Machines**

**VMs are an ideal choice for:**

* Total control over the operating system (OS)
* The ability to run custom software
* To use custom hosting configurations

**Scale VMs in Azure:**

* + Virtual machine scale sets
  + Azure Batch

**Virtual machine scale sets:**

* + Create and manage a group of identical, load-balanced VMs
  + Centrally manage, configure, and update a large number of VMs in minutes to provide highly available applications
  + The number of VM instances can automatically increase or decrease in response to demand

**Azure Batch:**

* + Enables large-scale parallel and high-performance computing (HPC) batch jobs with the ability to scale to tens, hundreds, or thousands of VMs

When you're ready to run a job, Batch does the following:

* + Starts a pool of compute VMs for you
  + Installs applications and staging data
  + Runs jobs with as many tasks as you have
  + Identifies failures
  + Requeues work
  + Scales down the pool as work completes

Use when you need raw computing power or supercomputer-level compute power

**When to use Azure App Service**

* You can build and host web apps, background jobs, mobile back-ends, and RESTful APIs in the programming language of your choice without managing infrastructure
* Offers automatic scaling and high availability
* Platform as a service (PaaS) environment
* App Service, you can host most common app service styles like:
  + Web apps
  + API apps : To build web-based REST API to handle HTTP and HTTPS request
  + WebJobs: To run a program (.exe, Java, PHP, Python, or Node.js) or script (.cmd, .bat, PowerShell, or Bash). Mainly used to run background tasks
  + Mobile apps

**When to use Azure Container Instance and Kubernetes Service**

**What is Docker?**

* Docker is an open platform for developing, shipping and running applications
* Docker packages software into standardized units called containers
* Container contains everything the software needs to run including libraries, system tools, code, and runtime
* Docker provides standard way to run your code
* Docker is an operating system for containers
* VM virtualizes the server hardware whereas containers virtualizes the operating system of a server
* Docker is installed in each server and provides simple commands to build, start and stop containers
* Docker containers are lightweight because containers share the machine’s OS system kernel and therefore do not require an OS per application

Graphical user interface, diagram, application

Description automatically generated

**When to use Azure Container Instance and Kubernetes Service**

* Running one or two containers in a single host is simple
* What happens when you have
  + Tens of host and hundreds of containers
  + Hundreds of host and thousands of containers
* Need to achieve availability, resilience and performance
* Need to know the state of the system
  + How do you know when the container dies?
  + How do you hook into the load balancer?
  + How to know which instances has available memory and port?
  + How to manage containers at scale?

This is where container management platform comes into picture

Two ways to manage both Docker and Microsoft-based containers in Azure:

* + Azure Container Instances
  + Azure Kubernetes Service (AKS)

**Azure Container Instances:**

* Fastest and simplest way to run a container without managing infrastructure
* Platform as a service (PaaS) offering

**Azure Kubernetes Service:**

* The task of automating, managing, and interacting with a large number of containers is known as orchestration
* Complete orchestration service for containers with distributed architectures and large volumes of containers

**Use Case:**

* To create a *microservice architecture*
* This architecture is where you break solutions into smaller, independent pieces

**Use Case:**

* To create a *microservice architecture*
* This architecture is where you break solutions into smaller, independent pieces

**Azure Container Instances:**

* It is a great solution for any scenario that can operate in **isolated containers**, including simple applications, task automation, and build jobs

**Azure Kubernetes Service:**

* For scenarios where you need full container orchestration, including service discovery across multiple containers, automatic scaling, and coordinated application upgrades

Diagram

Description automatically generated

When to use Azure Functions

**Serverless Computing:**

* *Serverless* computing is the abstraction of servers, infrastructure, and operating systems.
* Azure takes care of managing the server infrastructure and the allocation and deallocation of resources based on demand
* Scaling and performance are handled automatically
* There's no need to even reserve capacity

**Azure Functions:**

* Functions can execute code in almost any modern language

**Azure Logic Apps:**

* Logic apps are designed in a web-based designer and can execute logic triggered by Azure services without writing any code
* Executes workflows

**When to use Azure Virtual Desktop**

* Azure Virtual Desktop is a desktop and application virtualization service that runs on the cloud
* It enables users to use a cloud-hosted version of Windows from any location
* Azure Virtual Desktop works across devices like Windows, Mac, iOS, Android, and Linux

**Part 2: Explore Azure Storage Services**

**Azure Storage Account Fundamentals**

**Azure Storage:**

* You can use Azure Storage to store files, messages, tables, and other types of information
* To begin, you need to create an Azure Storage account to store your data objects
* Storage account will contain all Azure Storage data objects, such as blobs, files, and disks

Note: Azure VMs use Azure Disk Storage to store virtual disks. However, you can't use Azure Disk Storage to store a disk outside of a virtual machine

Disk Storage Fundamentals

**Disk Storage:**

* Provides disks for Azure virtual machines
* Block-level storage volumes
* Allows data to be persistently stored and accessed from an attached virtual hard disk
* Standard SSD and HDD disks - Less critical workloads
* Premium SSD disks - Mission-critical production applications
* Ultra disks - Data-intensive workloads such as SAP HANA, top tier databases, and transaction-heavy workloads

**OS disk**

* Every virtual machine has one attached operating system disk
* OS disk has a pre-installed OS, which was selected when the VM was created
* This disk contains the boot volume

**Data disk**

* Managed disk that's attached to a virtual machine to store application data, or other data you need to keep

Azure Blob Storage Fundamentals

**Blob Storage:**

* It is an Object storage
* To store massive amounts of data, such as text or binary data
* Azure Blob Storage is unstructured, meaning that there are no restrictions on the kinds of data it can hold

You store blobs in containers, which helps you organize your blobs depending on your business needs

**Azure File Fundamentals**

**File Storage:** Offers fully managed file shares accessible via the industry standard Server Message Block and Network File System (preview) protocols

* Azure file shares can be mounted concurrently by cloud or on-premises deployments of Windows, Linux, and macOS
* Applications running in Azure virtual machines or cloud services can mount a file storage share to access file data

**Blob Access Tiers**

**Access Tiers:**

**Hot access tier**: Optimized for storing data that is accessed frequently (for example, images for your website)

**Cool access tier**: Optimized for data that is infrequently accessed and stored for at least 30 days (for example, invoices for your customers)

**Archive access tier**: Appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (for example, long-term backups)

**Part 2: Explore Azure database and analytics service**

**Azure Cosmos DB**

* Fully managed NoSQL database for modern app development
* Single-digit millisecond response times, and automatic and instant scalability, guarantee speed at any scale
* Globally distributed, multi-model database service
* Supports schema-less data, which lets you build highly responsive and "Always On" applications to support constantly changing data. You can use this feature to store data that's updated and maintained by users around the world

**Azure Relational Databases**

|  |  |
| --- | --- |
| **IF YOU WANT TO** | **USE THIS** |
| Managed, intelligent SQL in the cloud | Azure SQL Database |
| Managed, always up-to-date SQL instance in the cloud | Azure SQL Managed Instance |
| Build scalable, secure and fully managed enterprise-ready apps on open-source PostgreSQL, scale out single-node PostgreSQL with high performance or migrate PostgreSQL and Oracle workloads to the cloud | Azure Database for PostgreSQL |
| Deliver high availability and elastic scaling to open-source mobile and web apps with a managed community MySQL database service or migrate MySQL workloads to the cloud | Azure Database for MySQL |
| Deliver high availability and elastic scaling to open-source mobile and web apps with a managed community MariaDB database service | Azure Database for MariaDB |
| Accelerate your transition to the cloud using a simple, self-guided migration process | Azure Database Migration Service |

**Azure Big Data and Analytics**

|  |  |
| --- | --- |
| **IF YOU WANT...** | **USE THIS** |
| Limitless analytics service with unmatched time to insight | Azure Synapse Analytics |
| A fully managed, fast, easy and collaborative Apache Spark based analytics platform optimized for Azure | Azure Databricks |
| A fully managed cloud Hadoop and Spark service backed by 99.9% SLA for your enterprise | HDInsight |
| A fully managed on-demand pay-per-job analytics service with enterprise-grade security, auditing, and support | Data Lake Analytics |

**Part 2: Explore Azure Network Services**

* Private network in Azure
* Azure virtual network enables Azure resources to securely communicate with each other, the internet, and on-premises networks

**Communicate between Azure resources**

* **Virtual networks**: Virtual networks can connect not only VMs but other Azure resources, such as the App Service Environment for Power Apps, Azure Kubernetes Service, and Azure virtual machine scale sets
* **Service endpoints**: You can use service endpoints to connect to other Azure resource types, such as Azure SQL databases and storage accounts. This approach enables you to link multiple Azure resources to virtual networks to improve security and provide optimal routing between resources.

**Azure Network Services**

|  |  |
| --- | --- |
| If you want to | Use this |
| Add private network connectivity from your corporate network to cloud | Azure ExpressRoute |
| Securely use the internet to access Azure Virtual Networks | Azure VPN Gateway |
| Choose how your traffic routes between Azure and the Internet | Routing preference |

---------------------------------------End---------------------------------