

DEVOPS MID 01

In Microsoft Azure, several key components are used to build and deploy cloud applications and services. Here's an overview of each of the components you mentioned, along with their functionalities:

Resource Group:

A resource group is a **logical container for organizing and managing Azure resources**. Resources within a group share the same lifecycle, permissions, and policies.

Storage Account:

A storage account in Azure provides scalable, secure, and highly available storage services. It can be used to **store data objects like blobs, files, queues, tables, and disks**.

Function App:

A Function App is a serverless compute service that allows you to run event-driven code (functions) without the need to manage infrastructure. It is often used for building microservices and serverless applications.

Function:

A function in Azure Functions is a piece of code that is executed in response to an event or trigger. It can be written in various programming languages and is used for serverless computing, such as processing HTTP requests, messages from queues, or responding to events.

HTTP Trigger:

An HTTP trigger is a type of event that causes an Azure Function to execute when an HTTP request is made to a specific URL. It's commonly used for building APIs and handling web requests.

App Service:

Azure App Service is a fully managed platform for building, deploying, and scaling web apps and APIs. It supports multiple programming languages and frameworks, including .NET, Java, Node.js, and more.

Virtual Machine (VM):

A Virtual Machine is a scalable compute resource that allows you to run virtualized Windows or Linux servers in the cloud. VMs can be customized with various configurations and used for a wide range of applications, including hosting web services, databases, and more.

Blobs (Binary Large Objects):

Blobs are a type of object storage used for storing unstructured **binary or text data, such as images, videos, documents, and backups**. They are often used for content distribution and media storage. Blobs are organized into containers, and you can choose between Block Blobs (optimized for streaming and storage) or Page Blobs (optimized for random read/write operations).

App Service VS VM :

Azure App Service is a Platform-as-a-Service (PaaS) offering ideal for hosting web apps and APIs. It simplifies deployment, management, and scalability, making it great for developers. You don't manage the underlying infrastructure but have less control.

Azure Virtual Machine, an Infrastructure-as-a-Service (IaaS) offering, provides full control over VM instances, suitable for various workloads. It's more flexible but requires manual management, including OS updates and scaling. Costs vary based on resource usage, offering more customization at the expense of greater administrative overhead.

Azure Blob Storage is a robust choice for storing both original images and generated thumbnails when using an **HTTP Trigger** for Azure Functions. It provides scalability, cost-effectiveness, data redundancy, and seamless integration with Azure Functions, making it well-suited for this specific use case.

CI ensures that code changes are integrated and tested frequently, while **CD** automates the process of delivering code to various environments, including production. Together, they enable teams to maintain a high level of code quality, reduce deployment risk, and release software more rapidly and reliably. **The choice between Continuous Delivery and Continuous Deployment depends on the organization's risk tolerance and specific deployment requirements.**

Continuous Delivery:

After successful CI, code is automatically deployed to a staging environment where additional testing (e.g., user acceptance testing) can take place. The final decision to release to production is manual.

Continuous Deployment: After successful CI, code is automatically deployed to production without manual intervention if it passes all tests.

GitHub Actions is an automation framework that can build, test, and deploy your app whenever a new commit is made in your repository

Hot Access Tier:

Purpose: Hot access tier is designed for frequently accessed data that requires low-latency access times.

Cost: Storage costs are higher compared to the cool and archive tiers, but there are no retrieval fees.

Use Cases: Suitable for frequently accessed data, active workloads, and applications that require fast data access.

Cool Access Tier:

Purpose: Cool access tier is for data that is infrequently accessed and can tolerate slightly higher access latencies.

Cost: Storage costs are lower compared to the hot tier, but there may be retrieval fees for accessing data.

Use Cases: Ideal for backup, archival, and data that is accessed less frequently but still needs to be readily available when needed.

Archive Access Tier:

Purpose: Archive access tier is for data that is rarely accessed and can tolerate higher retrieval latencies.

Cost: The lowest storage costs, but there are higher retrieval fees and longer retrieval times compared to hot and cool tiers.

Use Cases: Best for long-term archival of data that is expected to be accessed infrequently, such as compliance data and historical records.