Azure Virtual Machine Scale sets

- Azure Virtual Machine Scale Sets let you create and manage a group of load balanced VMs.
- The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide the following key benefits:
- Easy to create and manage multiple VMs.
- Provides high availability and application resiliency by distributing VMs across availability zones or fault domains.
- Allows your application to automatically scale as resource demand changes
- **Benefits:**
  - **Works at large-scale:** To provide redundancy and improved performance, applications are typically distributed across multiple instances. Scale sets support up to 1,000 VM instances
  - **Easy to create and manage multiple VMs:** When you have many VMs that run your application, it's important to maintain a consistent configuration across your environment.
  - With scale sets, all VM instances are created from the same base OS image and configuration.
  - Provides high availability and application resiliency
  - Scale sets are used to run multiple instances of your application.
  - Allows your application to automatically scale as resource demand changes. Customer demand for your application may change throughout the day or week.
  - To match customer demand, scale sets can automatically increase the number of VM instances as application demand increases, then reduce the number of VM instances as demand decreases.

Azure Storage Services

- The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios.
- Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud.
- Azure Storage data objects are accessible from anywhere in the world over HTTP or HTTPS via a REST API.
- The Azure Storage platform includes the following data services:
  - Azure Blobs: A massively scalable object store for text and binary data. Also includes support for big data analytics through Data Lake Storage Gen2.
  - Azure Files: Managed file shares for cloud or on-premises deployments.
  - Azure Elastic SAN (preview): A fully integrated solution that simplifies deploying, scaling, managing, and configuring a SAN in Azure.
  - Azure Queues: A messaging store for reliable messaging between application components.
  - Azure Tables: A NoSQL store for schemaless storage of structured data.
  - Azure Disks: Block-level storage volumes for Azure VMs.

Storage accounts

- An Azure storage account contains all of your Azure Storage data objects, including blobs, file shares, queues, tables, and disks.
- The storage account provides a unique namespace for your Azure Storage data that's accessible from anywhere in the world over HTTP or HTTPS.
- Data in your storage account is durable and highly available, secure, and massively scalable.

- Different Types of Storage Account:

Type of storage account	Supported storage services
Standard general-purpose v2	Blob Storage, Queue Storage, Table Storage, and Azure Files
Premium block blobs3	Blob Storage
Premium file shares3	Azure Files
Premium page blobs3	Page blobs only

Benefits of Azure Storage

Azure Storage services offer the following benefits for application developers and IT professionals:

- **Durable and highly available.** Redundancy ensures that your data is safe in the event of transient hardware failures. You can also opt to replicate data across data centers or geographical regions.
- **Secure.** All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- **Scalable.** Azure Storage is designed to be massively scalable to meet the data storage and performance needs of today's applications.
- **Managed.** Azure handles hardware maintenance, updates, and critical issues for you.
- **Accessible.** Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS.

Azure Blob Storage

- Azure blob storage is fundamental for the entire Microsoft Azure because many other Azure services will store the data within a storage account, inside the blob storage, and act upon that data.
- Azure Blob Storage is Microsoft's object storage solution for the cloud.
- Blob Storage is optimized for storing massive amounts of unstructured data.
- Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data.
- Blob Storage is designed for:
    - Serving images or documents directly to a browser.
    - Storing files for distributed access.
    - Streaming video and audio.
    - Writing to log files.
    - Storing data for backup and restore, disaster recovery, and archiving.
    - Storing data for analysis by an on-premises or Azure-hosted service.
- Users or client applications can access objects in Blob Storage via HTTP/HTTPS, from anywhere in the world.

Azure DNS

- The Domain Name System, or DNS, is responsible for translating (or resolving) a service name to an IP address.
- Azure DNS is a hosting service for domains and provides naming resolution using the Microsoft Azure infrastructure.
- Azure DNS not only supports internet-facing DNS domains, but it also supports private DNS zones.

- You can't use Azure DNS to buy a domain name. However, you can buy a domain name by using any third-party domain name registrar. Your domains then can be hosted in Azure DNS for record management.
- DNS domains in Azure DNS are hosted on Azure's global network of DNS name servers.
- Each DNS query is answered by the closest available DNS server to provide fast performance and high availability for your domain.
- Azure DNS supports both public and private DNS domains.

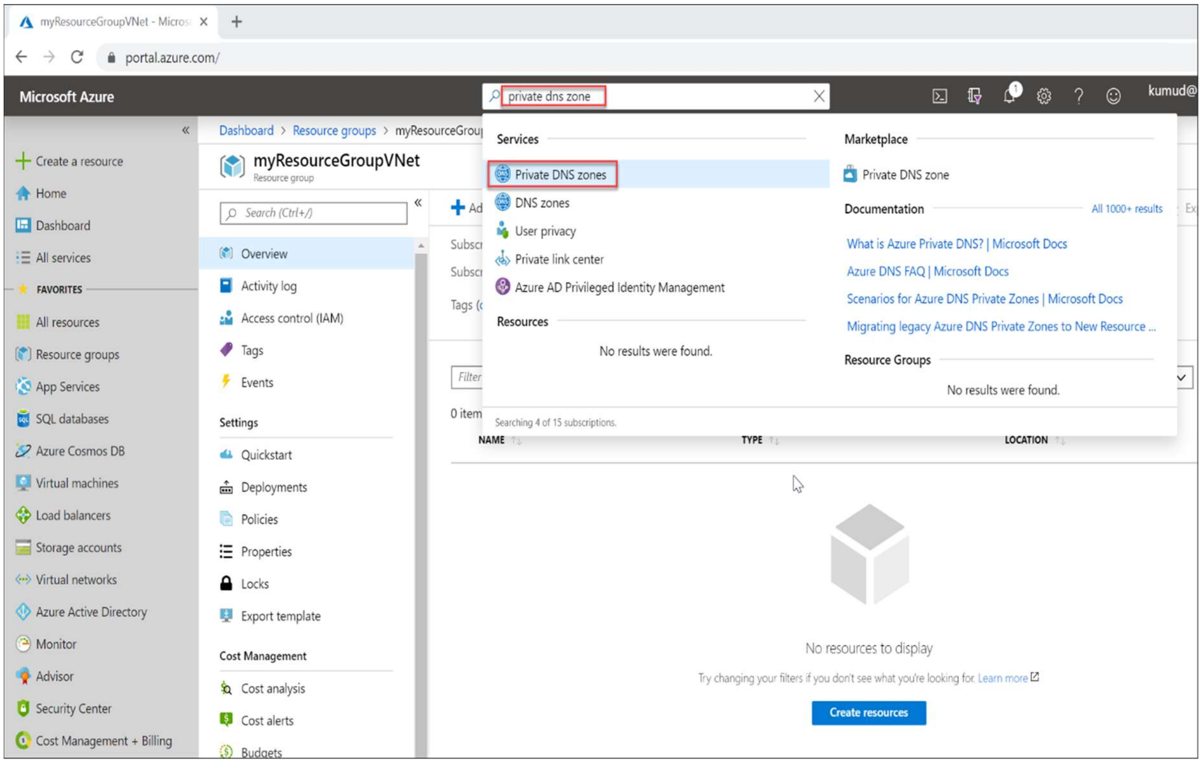
Azure Private DNS

- Azure Private DNS provides a reliable and secure DNS service for your virtual network.
- Azure Private DNS manages and resolves domain names in the virtual network without the need to configure a custom DNS solution.
- By using private DNS zones, you can use your own custom domain name instead of the Azure-provided names during deployment.
- Azure Private DNS provides the following benefits:
  - Removes the need for custom DNS solutions.
  - Use all common DNS records types. Azure DNS supports A, AAAA, CNAME, MX, PTR, SOA, SRV, and TXT records.
  - Automatic hostname record management.
  - Along with hosting your custom DNS records, Azure automatically maintains hostname records for the VMs in the specified virtual networks.
  - Familiar tools and user experience. To reduce the learning curve, this service uses well-established Azure DNS tools
  - Available in all Azure regions.

Create a private DNS zone

The following example creates a DNS zone called **private.contoso.com** in a resource group called **MyAzureResourceGroup**. A DNS zone contains the DNS entries for a domain. To start hosting your domain in Azure DNS, you create a DNS zone for that domain name.

- On the portal search bar, type **private dns zones** in the search text box and press **Enter**.
- Select **Private DNS zone**.
- Select **Create private dns zone**.
- On the **Create Private DNS zone** page, type or select the following values:
  - **Resource group**: Select **Create new**, enter *MyAzureResourceGroup*, and select **OK**. The resource group name must be unique within the Azure subscription.
  - **Name**: Type *private.contoso.com* for this example.
  - For **Resource group location**, select **West Central US**.
- Select **Review + Create**.
- Select **Create**

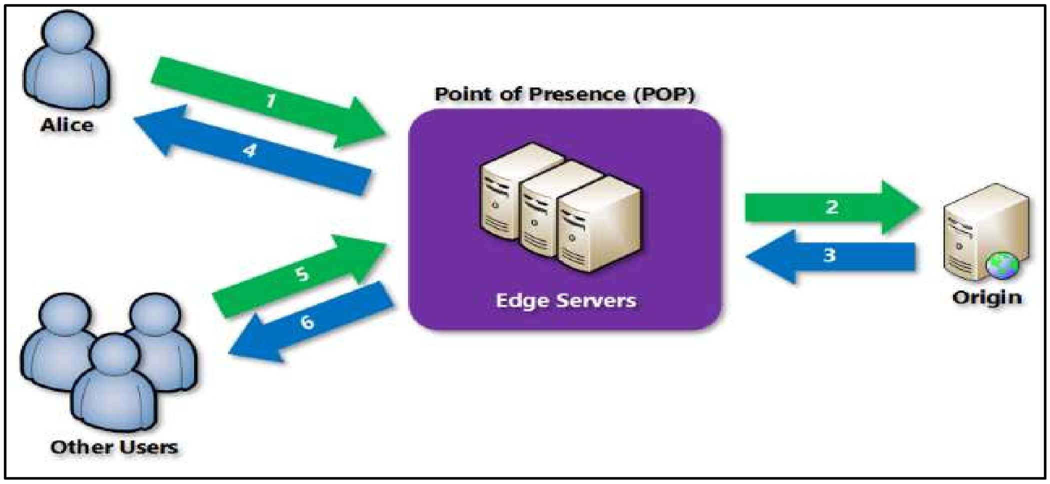


Azure CDN

Overview of Azure CDN

- A content delivery network (CDN) is a distributed network of servers that can speed up the deliver of web content to users.
- CDNs store cached content on edge servers in point-of-presence (POP) locations that are close to end users, to minimize latency.
- Azure CDN provides developers with a global solution for delivering high-bandwidth content to users by caching it at CDN POPs (Edge Locations or Edge Servers) throughout the world.
- The benefits of using Azure CDN:
  - Distribution of user requests and serving of content directly from edge servers so that less traffic is sent to the origin server.
  - High performance in delivering web content
  - No need to spend high amounts for setting up infrastructure
  - Easy configuration and maintenance
  - Decrease load on the main server and offload it to the edge servers

How Azure CDN works??



1. A user (Alice) requests a file (also called an asset) by using a URL. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.
2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server.
3. The origin server returns the file to an edge server in the POP.
4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time-to-live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.
5. Additional users can then request the same file by using the same URL that Alice used, and can also be directed to the same POP.
6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

How to use Azure CDN?

- When a user requests data from a website, the DNS sends this request the appropriate POP, which is usually the one that is geographically closest to the user.
- If the requested data is not present in a cache server on the POP.
- The Cache server requests the data from the origin server.
- The data is sent to the cache server and is cached there and returned.
- The user will then get this cached data.
- The advantage of caching this data is that any number of users who request the same kind of data from then will get a cached copy of the data until its life cycle in the POP server is over.
- This makes for the efficient delivery of content to the users.
- Caching is done on static content only but Azure CDN can also deliver dynamic content at high speeds through dynamic site acceleration.

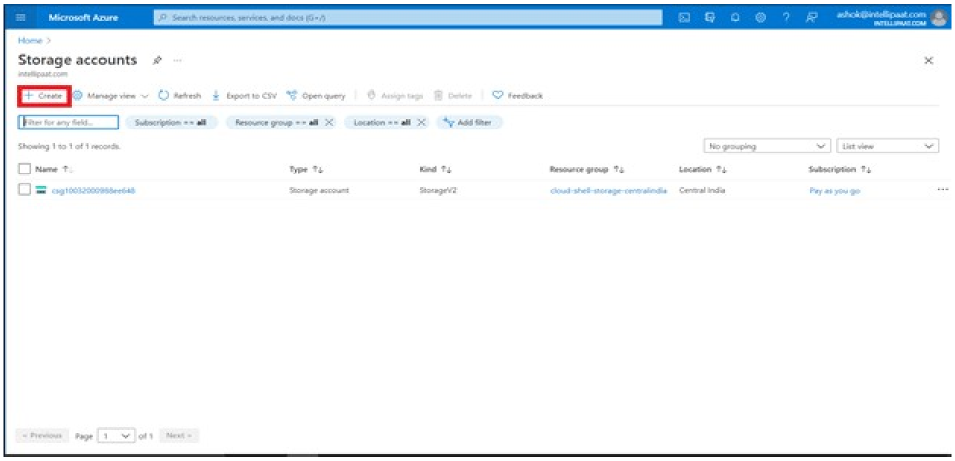
Create Azure CDN

- Azure App Service, Azure storage account, Azure cloud services, etc., can be used to configure the CDN.

Creating a Storage Account

- The first requirement is an Azure Account to which you should sign in. Go to portal.azure.com and log in with your credentials.

- Search for Storage Accounts and click on Create

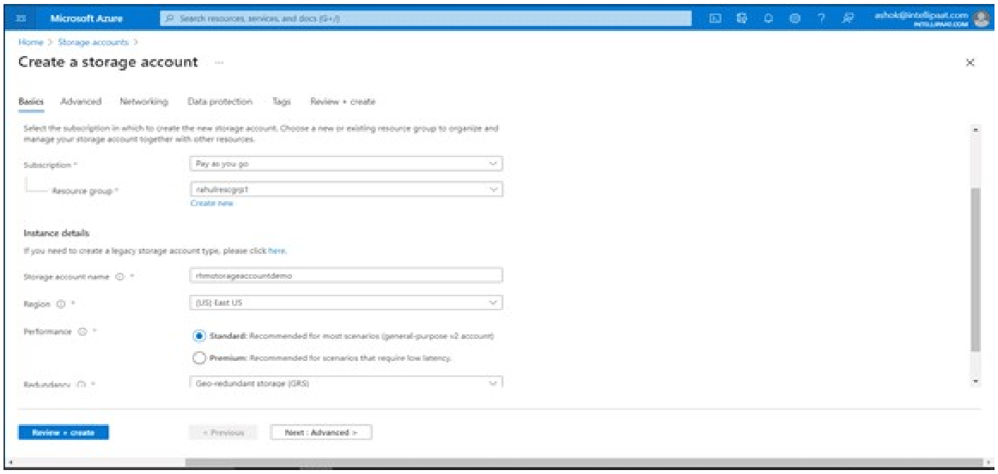


Storage accounts
intellipaat.com

Showing 1 to 1 of 1 records.

Name	Type	Kind	Resource group	Location	Subscription
csg10032000988ee648	Storage account	StorageV2	cloud-shell-storage-centralindia	Central India	Pay as you go

- Enter the details in the Create Storage Account panel.



Create a storage account

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *: Pay as you go

Resource group *: rahulrescgrp1

Instance details

If you need to create a legacy storage account type, please click here.

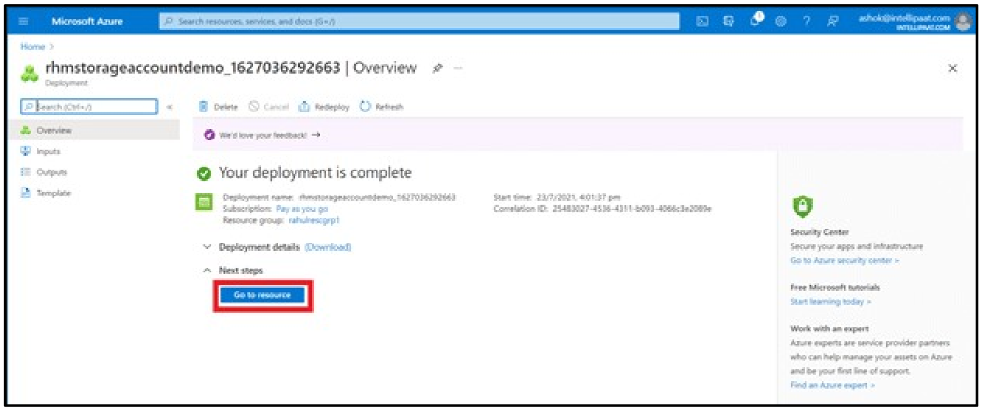
Storage account name *: rhmstorageaccountdemo

Region *: (US) East US

Performance *: ● Standard: Recommended for most scenarios (general-purpose v2 account) ○ Premium: Recommended for scenarios that require low latency.

Redundancy *: Geo-redundant storage (GRS)

- Add the subscription type you prefer. Give the name of the resource group if it already exists or create a new resource group here.
- The name of the storage account should be unique as it is used as an identifier for the storage account created.
- You can keep the rest of the settings in default and go for Review and Create.
- Creating the resource can be time-consuming. Once complete, click on Go to resource.


Microsoft Azure
Search resources, services, and docs (G+/)
ashok@intellipaat.com
INTELLIPAAT.COM
Home >
Search (Ctrl+/)
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Refresh
We'd love your feedback! →
Deployment details (Download)
Next steps
Go to resource
Security Center
Secure your apps and infrastructure
Go to Azure security center >
Free Microsoft tutorials
Start learning today >
Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

rhmstorageaccountdemo_1627036292663 | Overview
Deployment

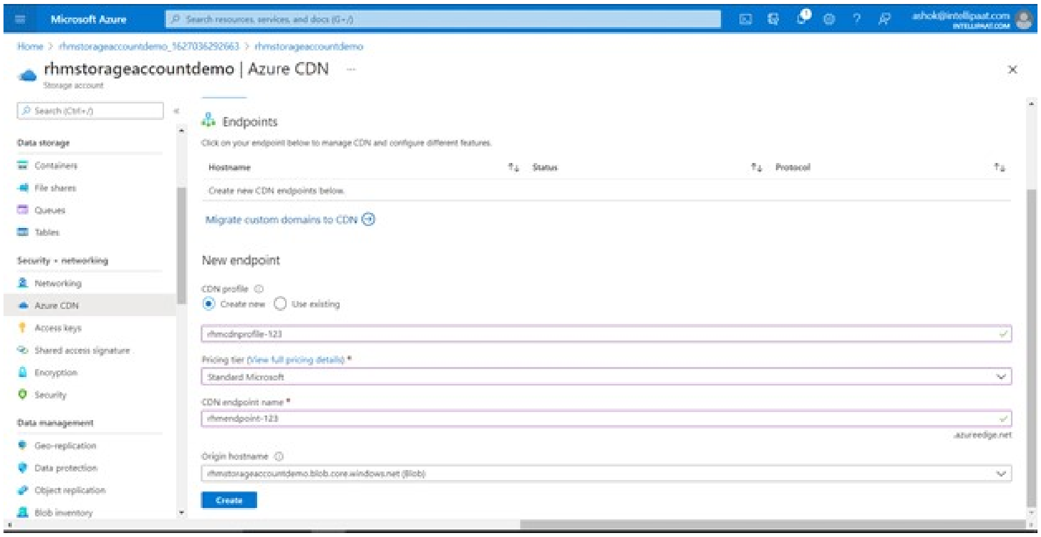
Your deployment is complete

Deployment name: rhmstorageaccountdemo_1627036292663
Subscription: Pay as you go
Resource group: rahulrescgrp1

Start time: 23/7/2021, 4:01:37 pm
Correlation ID: 25483027-4536-4311-b093-4066c3e2089e

Enabling Azure CDN for the Storage Account

- On the resource page, select Azure CDN from Security + networking in the left-hand menu and enter the details in the New endpoint section.



rhmstorageaccountdemo | Azure CDN

Storage account

Endpoints

Click on your endpoint below to manage CDN and configure different features.

Hostname	Status	Protocol
Create new CDN endpoints below.		

New endpoint

CDN profile: ● Create new ○ Use existing

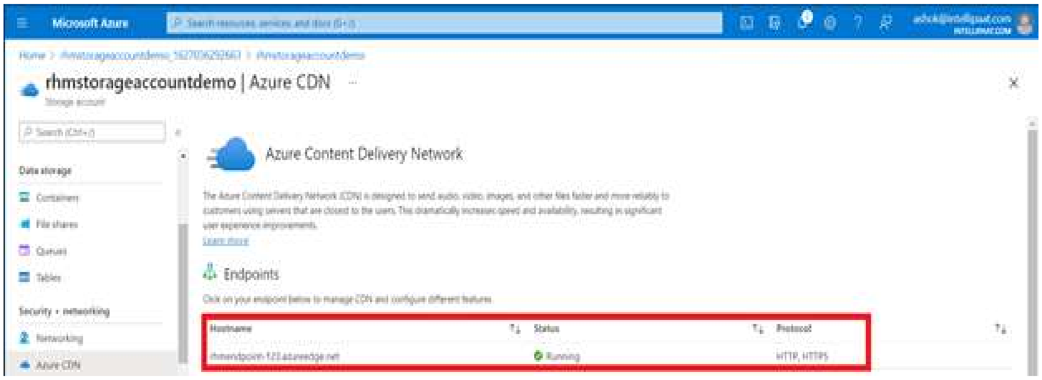
rhmcdnprofile-123

Pricing tier (View full pricing details) *: Standard Microsoft

CDN endpoint name *: rhmendpoint-123 .azureedge.net

Origin hostname: rhmstorageaccountdemo.blob.core.windows.net (Blob)

- Select Create New and enter a name for the CDN profile. The profile is a collection of endpoints. After that, give a unique name for the endpoint.
- Click on Create to deploy the CDN. The created endpoint can be seen in the list as running.



rhmstorageaccountdemo | Azure CDN

Storage account

Azure Content Delivery Network

The Azure Content Delivery Network (CDN) is designed to send audio, video, images, and other files faster and more reliably to customers using servers that are closest to the users. This dramatically increases speed and availability, resulting in significant user experience improvements.

Endpoints

Click on your endpoint below to manage CDN and configure different features.

Hostname	Status	Protocol
rhmendpoint-123.azureedge.net	Running	HTTP, HTTPS

Serverless Computing in Azure

- Serverless computing enables developers to build applications faster by eliminating the need for them to manage infrastructure.
- With serverless applications, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code.
- servers are still running the code. The serverless name comes from the fact that the tasks associated with infrastructure provisioning and management are invisible to the developer.
- This approach enables developers to increase their focus on the business logic and deliver more value to the core of the business.
- Serverless computing helps teams increase their productivity and bring products to market faster
- Benefits:

  - No infrastructure management: sing fully managed services enables developers to avoid administrative tasks and focus on core business logic.
  - Dynamic scalability: With serverless computing, the infrastructure dynamically scales up and down within seconds to match the demands of any workload.
  - Faster time to market
  - Serverless applications reduce the operations dependencies on each development cycle, increasing development teams' agility to deliver more functionality in less time.

Azure Serverless

- Go serverless—build apps faster without managing infrastructure.
- Why choose Azure serverless solutions?
  - Increase developer velocity
  - Reduce the time spent on tasks that are non-core to the business by freeing developers from infrastructure provisioning and management.
  - Boost team performance
  - Improve team agility and performance by using a fully managed platform to build, deploy and operate applications.
  - Improve organisational impact
- Accelerate time to market with Azure server-less solutions that help your organisation clear the path to innovation and new revenue opportunities.
- Reduce your infrastructure total cost of ownership and minimise risk with intelligent security management and advanced threat protection.

Azure serverless solutions

- Serverless Containerized Microservices
- Serverless Kubernetes
- Serverless functions
- Serverless application environments
- Azure Functions

Azure Functions

- Azure Functions is a serverless solution that allows you to write less code, maintain less infrastructure, and save on costs.
- Instead of worrying about deploying and maintaining servers, the cloud infrastructure provides all the up-to-date resources needed to keep your applications running.
- every application needs a way to run some code as these events occur.
- To meet this need, Azure Functions provides "compute on-demand" in two significant ways.
  - First, Azure Functions allows you to implement your system's logic into readily available blocks of code. These code blocks are called "functions". Different functions can run anytime you need to respond to critical events.
  - Second, as requests increase, Azure Functions meets the demand with as many resources and function instances as necessary - but only while needed. As requests fall, any extra resources and application instances drop off automatically.

- Use your preferred language: Write functions in C#, Java, JavaScript, PowerShell, or Python, or use a custom handler to use virtually any other language.

Lab – Creating and testing a Azure Function

Logic Apps

- Azure Logic Apps is a cloud platform where you can create and run automated workflows with little to no code.
- By using the visual designer and selecting from prebuilt operations, you can quickly build a workflow that integrates and manages your apps, data, services, and systems.
- Some of the examples that you can automate using Azure Logic Apps:
  - Schedule and send email notifications using Office 365 when a specific event happens, for example, a new file is uploaded.
  - Route and process customer orders across on-premises systems and cloud services.
  - Move uploaded files from an SFTP or FTP server to Azure Storage.
  - Monitor tweets, analyze the sentiment, and create alerts or tasks for items that need review.
- **Why use Azure Logic Apps?**
  - The Azure Logic Apps integration platform provides hundreds of prebuilt connectors so you can connect and integrate apps, data, services, and systems more easily and quickly.
  - You can focus more on designing and implementing your solution's business logic and functionality, not on figuring out how to access your resources.

Lab – Creating and testing a Azure LogicApp